



Tomas Bata University in Zlín
Faculty of Management and Economics

DOCTORAL THESIS

Organizational Security Processes and Crisis Management in the Knowledge Society

Krizový management a bezpečnostní procesy organizace ve znalostní
společnosti

Author:	Ing. Libor Sarga
Degree Program:	P 6208 Economics and Management
Degree Course:	6208V038 Management and Economics
Supervisor:	Assoc. Prof. Mgr. Roman Jašek, Ph.D.
Defense date:	2014

© Ing. Libor Sarga, 2014

Published by Tomas Bata University in Zlín.

Full version of the Doctoral Thesis available at the TBU Library.

Klíčová slova

bezpečnost, data, ICT, informace, komunikační, krizový, malware, management, metoda, mobilní, politika, proces, profil, riziko, technologie, testování, útok, zranitelnost

Keywords

attack, communication, crisis, data, ICT, information, malware, management, method, mobile, policy, process, profile, risk, security, technology, testing, vulnerability

ISBN

Abstrakt

Informační a komunikační technologie (ICT) tvoří součást infrastruktury většiny moderních organizací ve všech sektorech ekonomiky a jejich popularita neustále vzrůstá s novými produkty a snadnou dostupností také u spotřebitelů. ICT organizacím pomáhají plnit operativní i strategické cíle, definované v podnikových dokumentech, a jsou dnes již natolik důležité, že některá odvětví jsou podmíněna jejich bezchybnou funkčností a nepřetržitou dostupností, namátkou algoritmické a vysokofrekvenční obchody, elektronická tržiště, energetika, vojenství a zdravotnictví. Finanční, hmotné a lidské ztráty, které byly v poslední době zaznamenány dokazují neplatnost výše zmíněných předpokladů.

Autor se domnívá a předpokladem této disertační práce je, že úzkým místem jsou především zaměstnanci a jejich chování. Zajištění ICT bezpečnosti je proto úkolem, na němž by se měli svým přístupem a chováním podílet všichni členové organizace. Dizertační práce je proto zaměřena na oblast bezpečnosti ICT z pohledu organizace i uživatele. Po rešerši sekundárních literárních zdrojů jsou formulovány předpoklady pro výzkumnou část, jež nejprve zhodnotí současný stav. Výstupy pak budou základem pro tvorbu modelu a doporučení, zajišťujících zvýšení bezpečnosti ICT a uživatelů, přicházejících s nimi do kontaktu a považovaných útočníky za cenný zdroj informací.

Uživatelé se z různých příčin nechovají v souladu s nejlepšími praktikami bezpečnosti, což mohou podniky korigovat direktivními a vzdělávacími metodami a také ukázkami možných dopadů nevhodného využívání ICT. Jedním z aktuálních, ale málo rozšířených způsobů je cílené oddělení osobního a pracovního prostoru využitím politiky elektronických, centrálně distribuovaných profilů pro chytrá mobilní zařízení, dovolující přesnou definici povolení a omezení pro každý přístroj po dobu pracovní doby i při vzdálených interakcích s informačními systémy. Ošetření na úrovni uživatelů však musí být doplněno opatřeními pro další součásti ICT infrastruktury: zabezpečení operačních a databázových systémů; audity webových rozhraní; revize krizových plánů a plánů obnovy pro případy neočekávaných výpadků nebo napadení; pravidelné testování úzkých míst a jejich odstraňování; včasná instalace aktualizací; a průběžný monitoring a aplikace myšlenky útočníka při zkoumání slabín systému i jeho prvků.

V první kapitole disertační práce je vymezena terminologie. Ve druhé kapitole jsou představeny základní principy informační bezpečnosti, tvořící CIA (Confidentiality, Integrity, Availability) triádu. Ve třetí kapitole je popsán trend BYOD (Bring Your Own Device), který se začíná prosazovat při práci s citlivými daty, budou také přiblížena mobilní zařízení a jejich vliv na bezpečnost. Ve čtvrté kapitole budou demonstrovány útoky, pomocí nichž je útočník schopen neoprávněně získat přístup k citlivým datům zneužitím softwarové infrastruktury nebo zaměstnanců, kteří mohou být manipulováni prostřednictvím cílených nebo plošných pokusů o kompromitaci cílového systému. V páté kapitole jsou formulovány cíle, vědecké otázky, hypotézy a metody, které budou v disertační práci využity pro jejich testování. V šesté a sedmé kapitole jsou provedeny kvalitativního i kvantitativního výzkumu a vhodných statistických testů vysloveny závěry o představených hypotézách. V osmé kapitole je představen model ICT bezpečnosti a ekonomické ukazatele, které mohou organizacím pomoci při vyhodnocování ekonomických přínosů implementace opatření ICT bezpečnosti. V deváté kapitole jsou shrnuty výsledky disertační práce. V deváté kapitole jsou zmíněny přínosy pro vědu, praxi a výuku spolu s možnostmi budoucího navazujícího výzkumu.

Dizertační práce představí procesní model ICT governance integrující poznatky z výzkumné části. Také formuluje řešení pro hlavní aspekty organizační bezpečnostní politiky, například BYOD management, vzdělávání zaměstnanců, a zabezpečení infrastruktury spolu se správou hesel a nejlepšími praktikami v těchto oblastech. Výsledkem implementace tohoto modelu do

praxe by měl být podnik s politikou reflektující existující i nastupující hrozby, a vzdělání, na bezpečnost orientovaní zaměstnanci.

Abstract

Information and Communication Technology (ICT) forms infrastructure basis of most modern organizations in all economic sectors and has become more popular with individuals via new products and its ubiquitousness. Companies use ICT to fulfill both operational objectives and strategic goals, outlined in their fundamental documents. Nowadays, whole industries including algorithmic and high-frequency trading, online retailing, energy industry, military, and health care all assume uninterrupted ICT functionality and continuous availability. The repeated financial, material, and human losses that have occurred recently demonstrate this status should not be taken for granted.

It is the author's belief and the focus of this dissertation that the primary cause for these losses is people, and their actions. Hence, each employee should strive to minimize threat exposure. The doctoral thesis deals with corporate- and user-centric ICT security. Based on evaluation of secondary sources, assumptions for the research part will be formulated by first assessing the current state. The research output will then help formulate recommendations to promote increased security of ICT and users coming into contact with sensitive electronic assets whom the attackers consider a valuable source of information.

Individuals tend not to behave in a variety of means for various reasons, requiring organizations to employ directive and educational methods along with real-life demonstrations of inappropriate use of ICT. For example, one of the current, albeit scarcely used means are centrally-managed profiles separating personal and work space on small form-factor devices (smartphones, tablets) which allow specifying permissions and restrictions during work time and when remotely accessing protected data. The user-level focus must be complemented with efforts pertaining to ICT infrastructure elements: operating and database systems security; web user interface audits; revisions of crisis and contingency plans for unexpected disruptions or targeted actions; bottleneck identification and elimination; patch management; continuous monitoring; and applying attacker's mindset when discovering weaknesses within the system and its parts.

The first chapter delimits terminology used throughout the work. The second chapter introduces elements of the CIA (Confidentiality, Integrity, Availability) triad. The third chapter deals with BYOD (Bring Your Own Device), a trend increasingly common in organizations; mobile devices and how they may affect security will be also described. The fourth chapter demonstrates vectors which enable the adversary unauthorized system access using malicious techniques directed at ICT infrastructure as well as users who could be manipulated by custom-tailored or large-scale campaigns aimed to penetrate defensive measures and establish persistence. The fifth chapter formulates goals, scientific questions, hypotheses, and methods used to test them in the doctoral thesis. The sixth chapter presents and analyzes results of a large-scale questionnaire research conducted on a representative sample of participants. The eighth chapter consists of two case studies which practically investigates weaknesses in selected areas of ICT security. The ninth chapter outlines an ICT security governance model and economic metrics based on findings from the questionnaire research and the case studies. The tenth chapter lists contributions of the thesis for theory and practice, possible future research directions along with concluding remarks.

The ICT security governance model in chapter nine articulates recommendations for major aspects of organizational policies such as BYOD management, employee training, infrastructure hardening, and password management which are discussed and best practices devised. The result of implementing the model should be an organization capable to face existing and novel threats, and educated, security-conscious employees.

Acknowledgments

Here is a list of acknowledgments. I apologize in advance to anyone I omitted, but mentioning everyone would have taken more space than would be appropriate. You all know who you are and how much you helped.

I would like to thank my supervisor, Roman Jašek, for supporting me throughout the four years I have spent laboring over what to write and how.

My sincerest thanks to Edwin Leigh Armistead not only for his brilliant proofreading skills, but also for the opportunity he has decided to give me when (or even though) he saw what I was capable of.

Charlie Miller and Moxie Marlinspike both answered my emails and gave me the much-needed nudge when I was unsure where and whether to continue. Getting a reply from one was big, but from both – that was immense. Good work, guys!

The colleagues at the Faculty of Management and Economics, Tomas Bata University in Zlín who were patiently sitting at offices with me and coped with my incoherent rambling, particularly related to the doctoral thesis deserve a mention. Some went, some stayed, but none will be forgotten.

Thanks to the Department of Statistics and Quantitative Methods, Tomas Bata University in Zlín which gave me just the freedom I needed to engage in such a gargantuan task.

My family has always supported me and I would not have been able to finish the work without them. The time away from them spent researching and dreaming of the day when I would write this acknowledgment was long, but we did it at the end. Originally, I did not want to single out anyone from my family, but in March 2014, my grandfather passed away unexpectedly. I hoped he would have been proud of me when I would have had told him the work is finished; unfortunately, it was not meant to be. Therefore, I dedicate the thesis to him.

And the very special someone I would like to thank is Kristina. You are the most wonderful person I have ever met in my life, and words cannot express how much you mean to me. It was you who held my hand through the most challenging moments, and despite your own hardships persevered in your enthusiasm. Without your patience and guidance, I would have given up a long time ago, but you made me realize that when I start something, I need to finish it, too. You are and will be my light in the dark.

“It is only when they go wrong that machines remind you how powerful they are.”

– Clive James

CONTENTS

LIST OF FIGURES AND TABLES	8
List of Figures	8
List of Tables	12
1 INTRODUCTION	13
2 CURRENT STATE OF THE PROBLEM	15
2.1 Terminology	15
2.1.1 <i>Cybernetics</i>	15
2.1.2 <i>Data, Information, Knowledge, Wisdom</i>	17
2.1.3 <i>Business Process</i>	18
2.1.4 <i>Risk</i>	21
2.1.5 <i>Security</i>	23
2.2 The CIA Triad	27
2.2.1 <i>Confidentiality</i>	28
2.2.2 <i>Integrity</i>	30
2.2.3 <i>Availability</i>	35
2.3 Bring Your Own Device	42
2.3.1 <i>Background</i>	43
2.3.2 <i>Hardware</i>	44
2.3.3 <i>Software</i>	46
2.3.4 <i>Summary</i>	47
2.4 Techniques for Unauthorized System Access	49
2.4.1 <i>Modeling the Adversary</i>	50
2.4.2 <i>Human Interaction Proofs</i>	52
2.4.3 <i>Passwords</i>	54
2.4.4 <i>Communication and Encryption Protocols</i>	57
2.4.5 <i>Social Engineering</i>	61
2.4.6 <i>Virtual Machines</i>	63
2.4.7 <i>Web</i>	65
2.4.8 <i>Penetration Testing</i>	68
3 GOALS, METHODS	73
3.1 Goals	75
3.2 Methods	78
3.3 Topic Selection Rationale	81

4	QUESTIONNAIRE RESEARCH	83
4.1	Background	84
4.2	Results	90
4.2.1	<i>Personal Information, General IT Overview</i>	90
4.2.2	<i>Mobile Phones, Additional Questions</i>	106
4.3	Conclusion	121
5	CASE STUDIES	123
5.1	Case Study 1: Reverse Password Engineering	126
5.1.1	<i>Phase 1: Background, Methodology</i>	127
5.1.2	<i>Phase 2: Analyzing the Data Set and the Tools</i>	130
5.1.3	<i>Phase 3: Brute-Force and Dictionary Attacks</i>	134
5.1.4	<i>Phase 4: Results, Conclusion</i>	137
5.2	Case Study 2: Penetration Testing	158
5.2.1	<i>Phase 1: Background, Methodology</i>	160
5.2.2	<i>Phase 2: Information Gathering, Reconnaissance</i>	164
5.2.3	<i>Phase 3: Vulnerability Assessment and Identification</i>	178
5.2.4	<i>Phase 4: Conclusion</i>	195
6	THE ICT SECURITY GOVERNANCE MODEL	199
6.1	User-Side Security	203
6.2	ICT-Side Security	210
6.2.1	<i>BYOD Management</i>	213
6.2.2	<i>Internal Network Segmentation</i>	215
6.2.3	<i>Incident Response, ICT Infrastructure Hardening</i>	219
6.2.4	<i>Password Composition Requirements</i>	226
6.3	Model Metrics	233
6.4	Conclusion	247
7	RESULTS SUMMARY	249
8	CONTRIBUTIONS OF THE THESIS	251
8.1	Science and Theory	251
8.2	Practice	252
9	FUTURE RESEARCH DIRECTIONS	253
10	CONCLUSION	254
	REFERENCES	256
	Monographs	256
	Articles, Chapters	258
	Online	270
	Reports, Standards, White Papers	274
	LIST OF PUBLICATIONS	278
	APPENDICES	279

LIST OF FIGURES AND TABLES

List of Figures

1	Expanded Ideal Feedback Loop Model	16
2	Internet Over-Provisioning Scheme	17
3	The DIKW Pyramid	19
4	Sample Business Process	20
5	Risk Management Process According to the ISO	22
6	Security Taxonomy	26
7	The CIA Triad	27
8	Types of Database Transactions	31
9	Avalanche Effect	33
10	General Communications Model	34
11	Availability Breakdown	36
12	Virtualization	38
13	Hazard Function	40
14	Smartphone Block Diagram	44
15	Schematic View of an Operating System	47
16	Windows of Opportunity	50
17	CAPTCHA	53
18	Man-in-the-Middle Attack	60
19	Hypervisor	64
20	Common Test Types	68
21	Penetration Testing Phases	69
22	Security Concepts and Relationships	70
23	Cost/Benefit for Information Security	71
24	Thesis Milestones	74
25	Elements Influencing Security	77
26	Questionnaire Answer Patterns	85
27	Age and Gender Frequency Tables	91
28	Gender Frequencies Bar Charts	91
29	Age Frequencies Bar Chart	92
30	Age And Gender Contingency Table	93
31	Age And Gender Clustered Bar Chart	94
32	Age And Gender Pearson's Chi-Squared Test	95
33	IT Proficiency Classification of Respondents	96

34	Browser Selection Frequency Table	97
35	Browser Selection Bar Chart	98
36	Browser Update Frequency Contingency Table	98
37	Operating System Selection Frequency Table	99
38	Operating System Selection Pie Chart	99
39	HTTPS Understanding Frequency Table	100
40	Password Length Frequency Table	100
41	Password Length Pie Chart	101
42	Password Composition: Lowercase Characters	101
43	Password Composition: Uppercase Characters	102
44	Password Composition: Special Characters	102
45	Password Composition: Spaces	103
46	Password Length and Frequency of Change Table	104
47	Password Length and Frequency of Change Pearson's Chi-Squared Test	104
48	Password Selection Rules Frequency Table	106
49	Password Selection Rules Bar Chart	107
50	Password Storing and Reuse Frequency Table	108
51	Password Storing and Reuse Bar Chart	109
52	Mobile Operating Systems Preference Frequency Table	109
53	Mobile Operating Systems Preference Pie Chart	110
54	Smartphone Use Frequency Table	111
55	Smartphone Use Bar Chart	112
56	Smartphones and PCs Perceived Functions Comparison Pie Chart	112
57	Smartphone Password Storing and Lock Screen Contingency Table	113
58	Smartphone Password Storing and Lock Screen Clustered Bar Chart	114
59	BYOD Profiles Acceptance Frequency Table	114
60	BYOD Profiles Acceptance Pie Chart	115
61	Likert Scale Kruskal-Wallis Test	116
62	Spam Resilience Self-Assessment Bar Chart	117
63	Spam and Phishing Delimitation Contingency Table	118
64	Spam and Phishing Delimitation Bar Chart	119
65	Willingness to Engage in Computer Crime Frequency Table	120
66	Willingness to Engage in Computer Crime Pie Chart	120
67	Third-Party Software Module Dependency Violation	126
68	Data Set Structure	132
69	Hashcat-GUI Test Setup	135

70	Hashcat Command Line Interface	135
71	Hashcat-GUI Word List Management	136
72	MD5 Brute-Force Attack Plaintext Passwords Sample	139
73	Type I and Type II Error Dependence	140
74	Password Composition Patterns From Brute-Force Attack	144
75	Straight Mode Dictionary Attack Plaintext Passwords Sample	145
76	Straight Mode Dictionary Attack Password Length Histogram	146
77	Rule-Based Mode Dictionary Attack Plaintext Passwords Sample	147
78	Rule-Based Mode Dictionary Attack Password Length Histogram	148
79	Toggle-Case Mode Dictionary Attack Plaintext Passwords Sample	150
80	Toggle-Case Dictionary Attack Password Length Histogram	151
81	Permutation Dictionary Attack Password Length Histogram	153
82	Case Study 1 Selected Metrics Graph	156
83	Case Study 1 Final Results Graph	157
84	Target Directory Listing	167
85	Administrative Panel Disclosure	168
86	PHP Error Log Disclosure	169
87	Partial Netcraft Fingerprinting Output	170
88	TheHarvester Email Address List	172
89	TheHarvester Subdomain List	173
90	Partial RIPE NCC WHOIS Query Output	173
91	Metagoofil Sample Output	174
92	SquirrelMail Online Login Screen	175
93	Roundcube Online Login Screen	176
94	Spam Message Partial Email Header	177
95	Internal Email Complete Header	178
96	Live Host Scan Results	179
97	Nmap Scan Progress	180
98	Nmap Scan Output	181
99	Nessus GUI	182
100	Nessus External Basic Network Scan Results	183
101	Nessus RDP Login Screen Screen Shot	184
102	Nessus Internal Basic Network Scan Results	184
103	Web Server phpinfo Disclosure	185
104	Nessus Advanced Scans Safe Checks On Results	186
105	Nessus Advanced Scans Safe Checks Off Results	186

106	Metasploit Payload Setup for Host 1	188
107	Metasploit Payload Sent to Host 1	188
108	Loopback Social Engineering Email	193
109	Loopback Social Engineering Email Source	193
110	Spoofed Email as Seen by the Recipient	194
111	Spoofed Email Header	195
112	Porter's Value Chain Model	199
113	The Proposed ICT Security Governance Model	202
114	Sensitive Data Encryption Start Points	207
115	LastPass Return on Investment Calculation Form	208
116	Defense in Depth Model	211
117	Wi-Fi Signal Leaks	218
118	Sample Network Stratification Topology	219
119	Incident Response Decision Tree	220
120	Plan-Do-Check-Act Model	221
121	Markov Chain for Mobile Password Input Process	227
122	KeePass Classic Edition Password Generator	228
123	Password Haystack Statistics	229
124	Password Reverse Engineering Exponential Wall	244
125	Password Generation Session	245

List of Tables

1	Edge Cases of Selected Encryption Properties	30
2	Definition of ACID Axioms	31
3	Wireless Positioning Systems	46
4	Overview of Mobile OS Landscape	48
5	Algorithmic Growth Functions	55
6	Password Mutations List	56
7	Summary of Thesis Goals	79
8	Types of Sensitive Data	123
9	Brute-Force Attack Summary	138
10	Contingency Table for Fisher’s Exact Test	142
11	Brute-Force Output Pattern Matching	143
12	Straight Mode Dictionary Attack Summary	145
13	Straight Mode with Rules Dictionary Attack Summary	147
14	Toggle-Case Mode Dictionary Attack Summary	149
15	Permutation Mode Dictionary Attack Summary	152
16	Brute-Force and Dictionary Attack Comparison Metrics	154
17	Case Study 1 Final Results	155
18	Birth Numbers Search Space Enumeration	190
19	Sample Employee Training Curriculum	204
20	LastPass Return on Investment Calculation	209
21	Sizes of Alphabet Sets Used in Passwords	231
22	Password Lengths for Fixed Zero-Order Entropy	232
23	User-Side Model Metrics 1	236
24	User-Side Model Metrics 2	237
25	ICT-Side Model Metrics 1	241
26	ICT-Side Model Metrics 2	242

1 INTRODUCTION

Our society has been increasingly accentuating knowledge-based skills and competencies over traditional production factors of labor, land, and capital (Drucker, 1996). As more economic sectors have transformed to include knowledge as their primary innovation engine and competitive advantage (Porter, 1985), “knowledge society” emerged to denote their importance. Knowledge society is a continuation of previous cycles of history where data and information played identical roles as driving forces of economies and societies. Even though “data society” and “information society” have not been widely used, the ideas about them resulted in adopting new ways of thinking that offered different perspectives, and brought about new challenges. One of them was to effectively manage the technology processing electronic data and information.

Complexity forces individuals to specialize while retaining broad pool of general knowledge. Work teams are assembled from different nationalities irrespective of geographic, cultural, and demographic boundaries. Their members are expected to communicate and collaborate in order to provide novel angles of addressing problems, necessitating informed, data-driven decisions in the process. Data and information, some stored in electronic systems, can be combined to enable knowledge creation (Ackoff, 1989). Even though data and information are also found in physical form, efforts have been made to transfer as much of them as possible to digital form, demonstrating how ICT helps in fostering knowledge creation (Seki, 2008). This process will not be a matter discussed in the doctoral thesis, though; an assumption will be made that knowledge already exists.

The history of ICT is shorter compared to mathematics. Its influence, however, has been growing rapidly in decades since the inception of a silicon-based integrated circuit on the onset of 1960s (Lécuyer & Brock, 2010). Prohibitive prices, low hardware performance and software selection as well as limited portability were initially preventing spread of ICT into commercial sector. Technological advances (microprocessors, storage, memory, networks) and economies of scale decreasing Total Cost of Ownership (TCO) have gradually made the technology viable for corporations in need of storing data and information for later use, providing analytic facilities, ensuring redundancy, allowing local and remote interactions with fast access times, and making data operations (adding, updating, deleting) more convenient. Later, small and medium enterprises (SMEs) have begun to greatly invest in ICT to create temporary competitive advantage window, but as more companies were adapting to the market change, these gains diminished. Firms have thus started looking for novel ways to get ahead of competitors, fully embracing technological advantages made possible by data and information economies. Governments, local administrations, health care, and educational institutions have also recognized ICT’s growing importance and have been moving toward storing, transmitting, and processing digital data. Broadband Internet connections, mobile devices, computer-aided design, cloud computing, social networking, and touch interfaces all demonstrate how new technologies shape industries and individuals.

In spite of benefits and positive effects, the accelerating rate of change results in an ever-increasing gap between general level of knowledge and ICT complexity, a trend called digital divide (Bindé, 2005). The majority of users lack understanding of lower-level hardware and software functioning with a result that the ICT infrastructures in organizations become morally or technically obsolete while still processing sensitive data (industrial simulations, product blueprints, financial transactions, personally-identifiable information). Moreover, contingency plans are neither updated nor tested, IT risks remain unmonitored and unmanaged, and employees untrained. This contributes to a business environment where ICT is expected to be error-prone and perform adequately all the time without financial support due to a belief the technology is

sophisticated enough. Such presumptions have been proven incorrect on numerous occasions, and this is where the author intends to address some of the challenges.

The objective of this doctoral thesis is to assemble a model conceptualizing recommendations and best practices from application, computer, data, network, and mobile security to better protect users and organizations from threats emerging due to pervasive use of technology. Through detailed overview of existing and emerging attack vectors, the author possesses a strong background to make informed decisions about taking precautions, setting security policies, as well as estimating and decreasing ICT-related risk. The topic is highly relevant: it was estimated the amount of newly created or replicated data in 2011 would surpass 1.8 trillion gigabytes¹ while “[l]ess than a third of the information... can be said to have at least minimal security or protection; only about half the information that should be protected is protected” (Gantz & Reinsel, 2011).

The author believes organizations can better face the dynamic developments in ICT by hardening their infrastructures and focusing on employee training. The doctoral thesis provides means for both based on research utilizing primary data. Its results are then formalized into an ICT security governance model addressing major security issues.

¹1 gigabyte (GB) = 10^9 bytes = 2^{30} bits.

2 CURRENT STATE OF THE PROBLEM

2.1 Terminology

Before information security principles are detailed in chapter 2.2, a set of recurring terms will be delimited to prevent ambiguity in meaning. Definition of cybernetics in particular has undergone revisions. Originally, it had described a system which alters its behavior through a feedback loop based on external and internal stimuli using mathematical equations; application to social sciences necessitated new approach due to presence of human element not adhering to any single quantifiable principle. The notion of risk varies (financial, political, technological, ecological) and sources prioritize different aspects. The International Organization for Standardization (ISO) introduced general definition of risk as a consensual agreement among subject-matter experts but different industries adopted their own. Business processes have been extensively researched and delimiting them should not pose a challenge. While security may seem intuitive, lack of accepted metrics has made it difficult to measure and compare. Because cybernetics, risk, and security are controversial with many opposing and contradictory views, it is the author's opinion that multiple definitions, summaries, and references to later chapters should provide sufficient background so that no misinterpretation can occur. Data, information, knowledge, and processes will also be analyzed to the extent necessary for use without broader discourse.

2.1.1 Cybernetics

Cybernetics is a transdisciplinary field dealing with transmission and processing of information in biological and non-biological systems. These contain mechanisms (feedback loops) which allow to modify inputs based on output characteristics, a form of self-control or self-regulation. Systems can operate without external interventions for extended time periods if they are required to integrate only changes from within; however, all organized structures are placed in environments which influence them and vice versa. A system must accept feedback signals coming from inside (internal) and outside (external) with the latter exhibiting wide fluctuations (e.g., automated financial trading). In cases where variations would threaten its stability, system operator tweaks settings and threshold parameters to guarantee appropriate response. Unfortunately, social systems are not manageable in such a way.

Foundations of modern cybernetics were laid at the Macy Conferences during 1946–1953 (American Society for Cybernetics, 2008). Notable scientists including Claude E. Shannon, Norbert Wiener, John von Neumann or Heinz von Foerster attended, each of whom contributed substantially to the field. Wiener is considered a founder of modern cybernetics (Yingxu Wang, Kinsner, & Zhang, 2009); he introduced it as the science of communication and autonomous control in both machines and living things (Wiener, 1965). The concept of self-operating machines had previously been explored by Turing (1950), Shannon (1956), and von Neumann (1981). Of special note is the work by Alan Turing who proposed a test to determine whether a machine is capable to act intelligently to the point where it is hardly or entirely unrecognizable from human as judged by human observer. Artificial intelligence (AI), a branch of computer science closely related to cybernetics was established to research such hardware and software agents. A reverse Turing test where humans prove themselves to machines, often encountered as Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA), also exists. In chapter 2.4.2, attacks will be demonstrated which employ machine learning techniques to break the test, creating a loop whereby a machine is used to help human pass a machine-imposed challenge.

Figure 1 illustrates an expanded ideal feedback loop model from a cybernetic viewpoint. System input is influenced by a sum of external stimuli taken into account when adjusting its behavior, and the feedback loop signal. Its immediate output, again modulated by some properties from the outside environment serves two functions: closing the feedback loop and producing a response to be sent out which alters all systems in the surroundings.

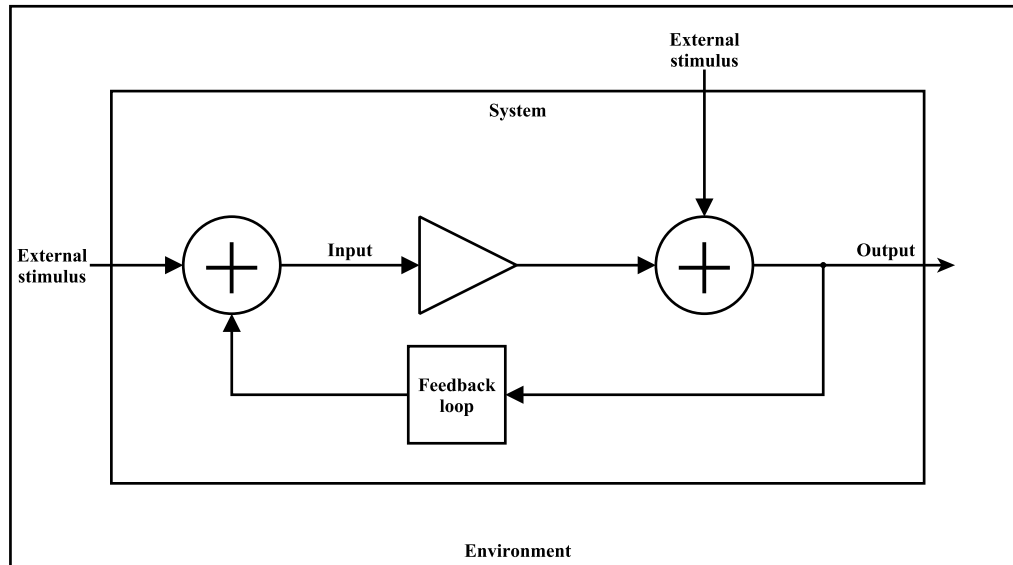


Fig. 1: *Expanded ideal feedback loop model. The system generates a response which reflects internal and external changes. Output influences other systems in the surroundings, altering their input characteristics.*

Source: own work.

Wiener’s limited definition needed expansion when findings from cybernetics were incorporated to anthropology, biology, control, information and systems theory, psychology, sociology, etc. In 1970s, a new (second-order) cybernetics emerged emphasizing how the participant observer affects the system under investigation due to them being part of either internal or external environment. A number of academics including Harries-Jones (1988), Pask (1996) , and von Foerster (2003) all argued self-organization without strict control to be an important property, providing complex systems (social, economic) with autonomy to flexibly react to changes. Interestingly enough, the Internet could be considered such structure: the Border Gateway Protocol (BGP) (Network Working Group, 2006) was designed to de-emphasize hardware components in favor of a result-based approach where each packet, the elementary unit of electronic communication, travels from one point to another through a series of “hops” using dynamically updated tables. The tables provide distinct pathways from source (input) to destination (output) with the lowest packet delivery (result) time as the decision criterion, making the Internet an algorithmically self-organizing system. The Internet is therefore designed as a highly redundant, over-provisioned network resistant to major disruptions.

Figure 2 schematically depicts the Internet routing structure. Nodes 0–15 represent hardware devices paired with several others, creating an interconnected mesh with redundant paths. When one becomes unavailable due to failure or packet overload, the remaining nodes will redistribute the traffic and provide surrogate routes for packets to travel through. If node 4 stopped responding and the objective was to deliver a packet from 0 to 10, four routes differing in the number of “hops” would be available:

- 5 “hops”: 0–3–7–8–9–10, 0–1–5–8–9–10,

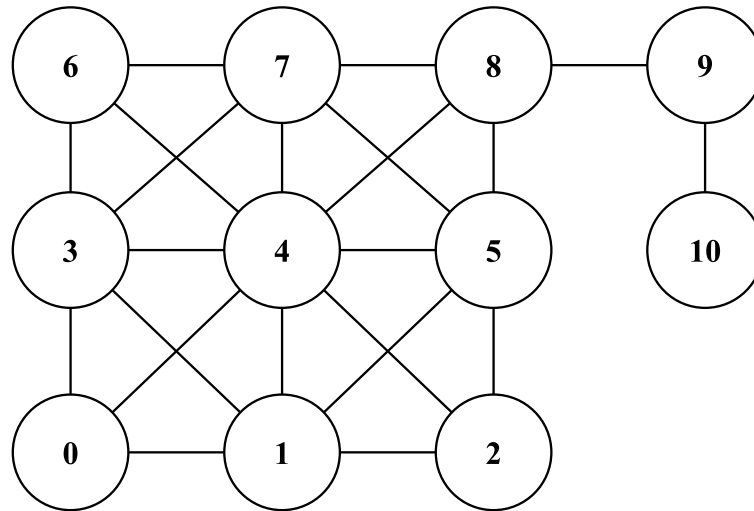


Fig. 2: Internet over-provisioning scheme. Each circle represents a “hop” (a hardware device) which forwards packets according to predefined, frequently updated tables.
Source: own work.

- 6 “hops”: 0–3–6–7–8–9–10, 0–1–2–5–8–9–10.

Nodes 8 and 9 are bottlenecks: was either of them to malfunction, destination (10) would become unreachable. An adversary can utilize this fact to identify system bottlenecks and flood them with denial-of-service attacks presented in chapter 2.4.4. They aim to disrupt ICT services by forcing components to repeatedly perform time- or resource-intensive operations on bogus incoming requests, saturating their resources. Such external stimulus results in instability if it goes undetected, distorting system’s input and generating skewed output.

2.1.2 Data, Information, Knowledge, Wisdom

Data is “. . . symbols that represent properties of objects, events and their environment. They are the products of *observation*. But are of no use until they are in a useable (i.e. relevant) form,” (Rowley, 2007, p. 5) or “. . . a set of discrete, objective facts about events” (Davenport & Prusak, 2000, p. 2). Data originates in physical signals, e.g., temperature, elevation, precipitation, speed, weight which are quantified and lack interpretative power supplied by an external agent (human, computer). Data is an unbiased set of values in numeric, textual, or other standardized form. Giarratano and Riley (2004) argue it is sampled from noise and hold a degree of subjectivity. In computing, data is “[a] subset of information in an electronic format that allows it to be retrieved or transmitted” (Committee on National Security Systems, 2010). Organized, structured, and stored in a single central repository or distributed to multiple locations, data may span passwords, credit card numbers, transactions, bank account balances, social security numbers, addresses, medical histories, files, and anything else considered sensitive due to regulatory obligations or policies.

Information, the second level of the DIKW (data, information, knowledge, wisdom) pyramid, “. . . is contained in descriptions, answers to questions that begin with such words as who, what, when and how many. . . [it] is inferred from data,” (Rowley, 2007, p. 5) or “. . . a *message*, usually in the form of a document or an audible or visible communication. . . [I]t has a sender and a receiver. . . [Information] is data that makes a difference” (Davenport & Prusak, 2000, p. 3). As there is a relation between data and information, any bias in the former will be projected

in the latter. Henry (1974, p. 1) makes the distinction less clear by not separating it from knowledge, and defining information as "... data that change us," a stance corroborated by Committee on National Security Systems (2010, p. 35): "Any communication or representation of knowledge such as facts, data, or opinions in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual." This top-down approach, i.e., with information representing knowledge, contrasts and complements the bottom-up reasoning that information gives meaning to data. In computing, information systems are used to store, retrieve, manipulate, and delete data while at the same time providing tools to extract, visualize, communicate, and simplify information. Multiple sets of information can be created from a data source each of which represents a valid output, but differs in the way it is interpreted; unlike data, information is pervaded with subjective connotations.

Knowledge is an epistemological construct, "... know-how, and is what makes possible the transformation of information into instructions. [It] can be obtained either by transmission from another who has it, by instruction, or by extracting it from experience," (Rowley, 2007, p. 5) or alternatively "... a fluid mix of framed experience, values, contextual information, and expert insights that provides a framework for evaluating and incorporating new experiences and information. . . it often becomes embedded not only in documents and repositories but also in organizational routines, processes, practices and norms" (Davenport & Prusak, 2000, p. 5). Despite the existence of knowledge management and knowledge economy, practitioners "... have failed to agree on [its] definition. . . Rather, their efforts have been directed toward describing different knowledge dichotomies. . . and ways in which to manipulate [it]" (Biggam, 2001, p. 7). Answers to questions such as if knowledge can be forgotten, effectively managed, formalized while retaining its properties, quantified, measured, unwillingly transferred, or stolen are to be determined. ICT security focuses on knowledge in databases and other sources to be protected, including people possessing and utilizing it.

Wisdom is a philosophical term, "the ability to increase effectiveness. [It] adds value, which requires the mental function that we call judgement. The ethical and aesthetic values that this implies are inherent to the actor and are unique and personal" (Rowley, 2007, p. 5). Defining wisdom is challenging, and Boiko (2004) even dismisses it from the model entirely. Wisdom will not be discussed further as its direct impact is limited in ICT security.

The DIKW pyramid depicted in Figure 3 shows how each layer corresponds to a particular type of information system. Transaction processing systems (TPS) are hardware and software which divide operations into units (transactions) and execute them sequentially, in batches, or simultaneously via time-sharing. Management information systems (MIS) include data, hardware, software (identically to TPS) together with procedures and people. Decision support systems (DSS) extend MIS with extensive predictive capabilities from existing data. Expert systems (ES) employ if-then rules to emulate an expert using natural language processing and fuzzy logic required for incomplete information.

2.1.3 Business Process

A business process is "... a set of linked activities that take an input and transform it to create an output. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream," (Johansson, McHugh, Pendlebury, & Wheeler, 1993, p. 16) or "... a structured, measured set of activities designed to produce a specific output for a particular customer or market" (Davenport, 1992, p. 5). Hammer and Champy (1993, p. 35) define it as "... a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer." Initially

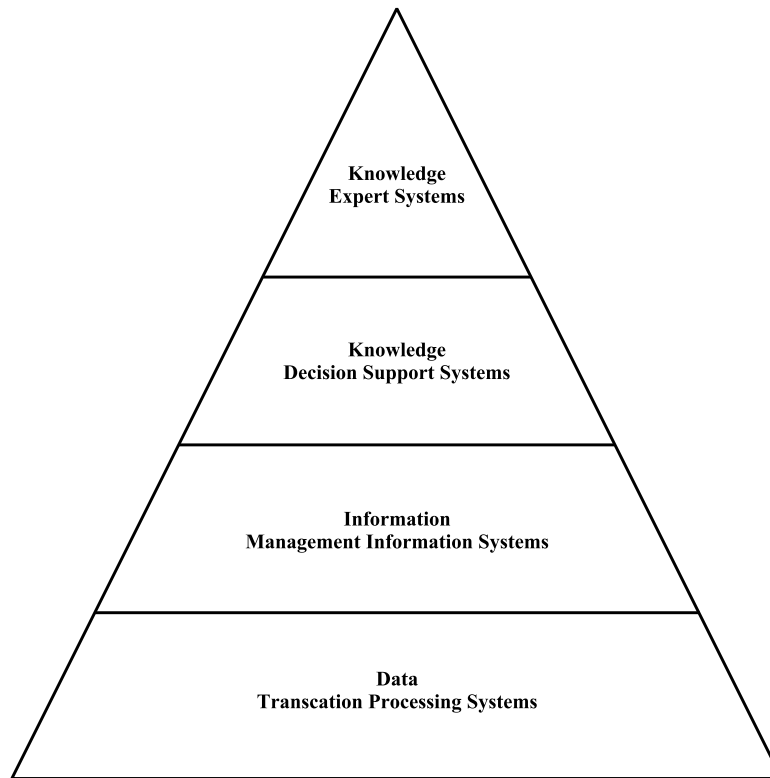


Fig. 3: *The DIKW Pyramid. Expert systems are based on knowledge formalization, simulating the subject-matter expert's decision-making process.*
 Source: Rowley (2007, p. 15), modified.

formalized in 1990s "... to identify all the activities that a company performs in order to deliver products or services to their customers," (Rotini, Borgianni, & Cascini, 2012) business processes have become a central point of strategies aiming to analyze, automate, iteratively or continuously improve, manage, map, partly outsource, and reengineer process hierarchies. Each process should have the following:

- input, output: transformation takes in tangible or non-tangible sources such as data, equipment, knowledge, methods, people, raw materials, specifications; the output, material (products) or immaterial (data, knowledge, methods, specifications), is intended for a customer,
- owner: an entity responsible for its successful fulfillment, constrained by available resources, duration, and customer's demands,
- duration: time frame during which the transformation takes place,
- transformation: a sequence which uses input is to produce output using procedures,
- procedure: a result-oriented activity ordered in time and place,
- customer: internal or external entity receiving the output, i.e., input to a consequent process,
- added value: the result of transformation beneficial or useful for the customer,
- embeddedness: placement in a process map visualizing processes' concurrence to produce outputs for customers.

A sample business process is depicted in Figure 4. Its inputs include data (credit score), information (customer's decision to apply for a loan), equipment (systems handling the transactions), people (customer, bank employee), methods (processing the application), and specifications (setting credit score and requested amount thresholds); output are data (modification of customer's

database entry, internal) and information (notification of acceptance or rejection, external). In this case, bank loan manager is the process owner.

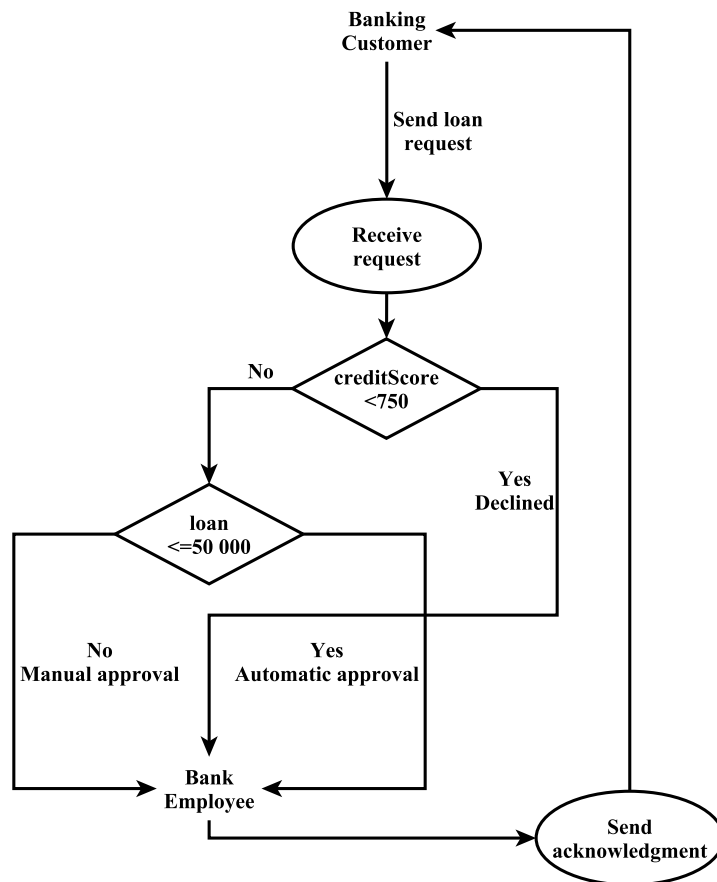


Fig. 4: Sample business process. Criteria to determine if a customer is eligible for a loan are their credit score (numeric value stored in a database) and the amount of money requested. The first decision is automatic, the second may involve bank employee’s permission to grant a loan over a set threshold. Source: IBM (2007), modified.

The role of owner and customer is not universally accented. The International Organization for Standardization defines process as “... an integrated set of activities that uses resources to transform inputs into outputs” (ISO, 2008b) without specifying value added throughout its execution nor defining responsible parties. The input→transformation→output sequence has also been criticized as too simplistic, and alternative views offered: business processes could be understood as deterministic machines, complex dynamic systems, interacting feedback loops, and social constructs (Melão & Pidd, 2000). Deterministic machines draw from computer science, especially theoretical work of Turing (1937, 1938, 1948) while the remaining views are associated with higher-order cybernetics. All may provide new perspectives on how organizational dynamics ties to formal descriptions of system functioning. This common ground was demonstrated earlier in Figure 1 which is strongly reminiscent of business process visualization.

Processes are sorted into three categories: management, operational, and supporting. The first group consists of meta-processes¹ which influence the remaining two and “... by which firms plan, organise and control resources” (Weske, van der Aalst, & Verbeek, 2004, p. 2). Operational (core) processes are “... those central to business functioning and relate directly to external customers. They are commonly the primary activities of the value chain” (Earl & Khan, 1994,

¹[ˈmɛtə], in n., adj., and v.: higher; beyond (Oxford University Press, 2011).

p. 2). Supporting processes use “. . . methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information,” (Weske et al., 2004, p. 2) “. . . have internal customers and are the back-up (or ‘back office’) of core processes. They will commonly be the more administrative secondary activities of the value chain” (Earl & Khan, 1994, p. 2). A fourth type, business network processes, is sometimes added and comprises “. . . those [processes] which extend beyond the boundaries of the organisation into suppliers, customers and allies” (Earl & Khan, 1994, p. 2) to accentuate growth of such tendencies in supply chains.

ICT belongs to the third category, i.e., supporting processes because it permeates not only core but management and business network processes. With communication across and within the organizational boundary conducted electronically, often in automated fashion, the space of attack vectors has been substantially expanded with few to no updates to security processes, employee training, and risk management.

2.1.4 Risk

Understanding of risk is tied to the development of probability theory in the 20th century. Knight first distinguished between subjective and objective probabilities by introducing the terms risk and uncertainty, respectively. “[T]he practical difference between the two categories. . . [is] that in the former the distribution of the outcome in a group of instances is known (either through calculation *a priori* or from statistics of past experience), while in the case of uncertainty this is not true, the reason being. . . it is impossible to form a group of instances, because the situation dealt with is in a high degree unique” (Knight, 1921, p. 233).

Risk is also related to objective and subjective interpretation of probability. Objective interpretation argues probabilities are real and discoverable by means of logical inference or statistical analyses, subjective equates them with beliefs humans specify to express their own uncertainty, extrinsically to nature (Holton, 2004). Therefore, given probability of previous occurrences, risk can be quantified which is favored by frequentist statistics (Venn, 1866) as opposed to Bayesian (Savage, 1972). In case such data are not available, it is instead correct to denote assumptions about future events as uncertainty. Nevertheless, as Mas-Colell, Whinston, and Green (1995, p. 207) point out: “[t]he theory of subjective probability nullifies the distinction by reducing all uncertainty to risk through the use of beliefs expressible as probabilities.” Henceforth, the term risk will denote chance of positive or negative outcome, although its perception predominantly leans toward the latter (Slovic & Weber, 1987).

Burt (2001) states: “Risk is the probability that an event will occur.” The definition is neutral, not specifying the event as either positive or negative. The International Organization for Standardization explicitly recognizes risk as “the effect of uncertainty on objectives,” effect as “deviation from the expected (positive or negative),” and uncertainty as “the state, even partial, of deficiency of information related to understanding or knowledge of an event, its consequence or likelihood” (ISO, 2009). The previous version of the standard (ISO, 2002) delimited risk as “chance or probability of loss.” Risk thus referred to advantageous and disadvantageous qualities associated with an action, going against the prevailing non-technical view associating it only with the latter.² The ISO definition is a consensus based on comments from thousands of subject-matter experts. The National Institute of Standards and Technology (NIST) adapted its definition from Committee on National Security Systems (2010, p. 61): “A measure of the extent

²[risk], *noun*: the possibility of something bad happening at some time in the future; a situation that could be dangerous or have a bad result (Oxford University Press, 2011).

to which an entity is threatened by a potential circumstance or event, and typically a function of 1) the adverse impacts that would arise if the circumstance or effect occurs; and 2) the likelihood of occurrence” (Computer Security Division, 2012, p. 59).

The International Organization for Standardization also focuses on risk management and risk management plans. The former is defined as “the co-ordinated activities to direct and control an organisation with regard to risk;” the latter as a “scheme, within the risk management framework, specifying the approach, the management components, and resources to be applied to the management of risk” (Dali & Lajtha, 2012). Risk management process is visualized in Figure 5.

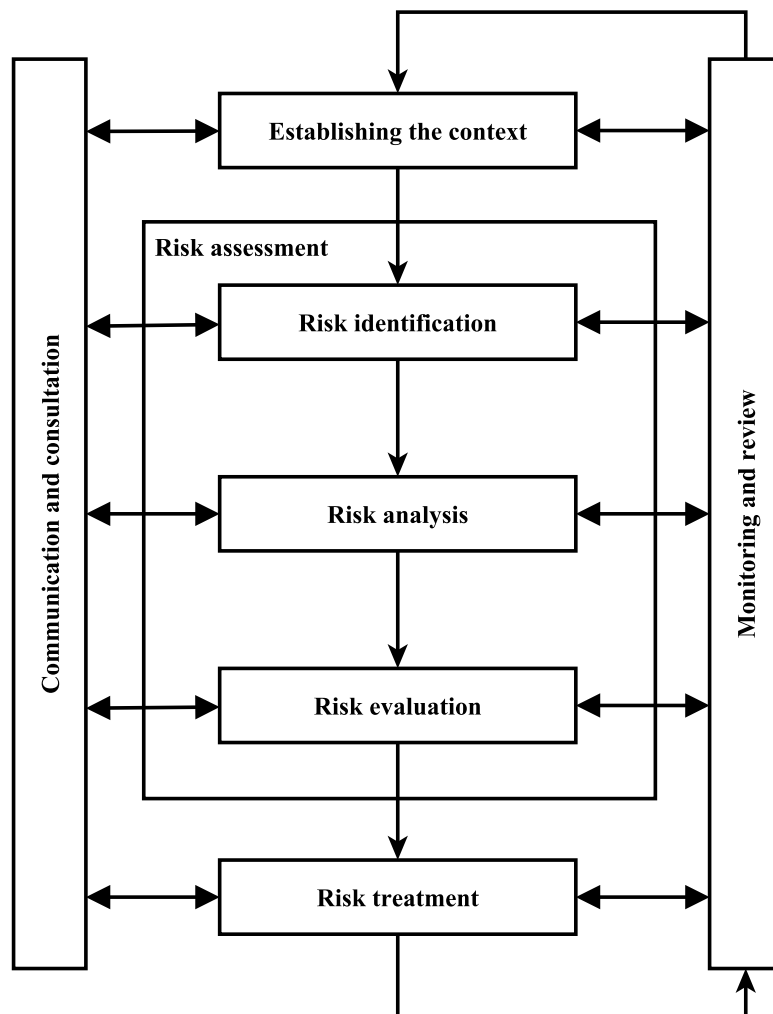


Fig. 5: Risk management process according to the ISO. The 73:2009 standard contains three interconnected modules: Principles for managing risk (1), Framework for managing risk (2), and Process for managing risk (3) with (1) → (2) ↔ (3) dependency chain. Module (3) is depicted here. Source: Dali and Lajtha (2012), modified.

Decision and game theory formalize the decision-making process under uncertainty, i.e., in absence of perfect information where risk is associated with each option. In economics, a proxy variable, utility, has been introduced to characterize the “most beneficial” out of all valid combinations. It is described as “[a] measure of the satisfaction, happiness, or benefit that results from the consumption of a good,” (Arnold, 2008, p. 402) for example associated with deploying a new security product. The alternative yielding the highest utility should be preferred. Economic utility is usually simplified to include two goods which limits its real-world use where

several options exist. Decision and game theory do not utilize economic utility model but were regardless criticized for assumptions such as agent rationality and known space of possibilities (Taleb, 2010).

From the business perspective, risk is “. . . the potential loss suffered by the business as a result of an undesirable event that translates either into a business loss or causes disruption of the business operations” (Open Information Systems Security Group, 2006, p. 81). Information technology-related risk is further defined as “. . . [a] business risk – specifically, the business risk associated with the use, ownership, operation, involvement, influence and adoption of IT within an enterprise” (ISACA, 2009, p. 7). This type of risk will be of interest in later chapters to encompass threats from external (attackers) and internal (insiders) sources as well as opportunities (deploying and testing patches to protect ICT systems and ensuring compatibility for customers). A language-neutral definition tying risk to security sees it as “[t]he level of impact on organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring” (Computer Security Division, 2006, p. 8).

2.1.5 Security

Security is “[a] form of protection where a separation is created between the assets and the threat. . . In order to be secure, the asset is removed from the threat or the threat is removed from the asset,” (Herzog, 2010, p. 22) or “[a] condition that results from the establishment and maintenance of protective measures that enable an enterprise to perform its mission or critical functions despite risks posed by threats to its use of information systems. Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the enterprise’s risk management approach” (Committee on National Security Systems, 2010, p. 64). It is understood as a state where each threat, i.e., “a potential cause of an incident, that may result in harm of systems and organization,” (ISO, 2008a) or “[a]ny circumstance or event with the potential to adversely impact an asset through unauthorized access, destruction, disclosure, modification of data, and/or denial of service,” (European Network and Information Security Agency, 2013) above certain threshold from within or outside the system has its associated risk partially mitigated by a suitable countermeasure. Not all threats are tracked due to risk tolerance, acceptance of some risk level as inherent. Resources spent on mitigating low-rated risks usually outweigh the benefits of increased security if suitable metric is introduced to quantify it.

Some discussion has been generated over the inclusion of the term “deterrence” in the definition of security mentioned above (Riofrio, 2013) as initiation of actions to thwart or retaliate against an attack may be in violation of law, e.g., Computer Fraud and Abuse Act. On the other hand, The Commission on the Theft of American Intellectual Property (2013, p. 6) suggest the following: “Without damaging the intruder’s own network, companies that experience cyber theft ought to be able to retrieve their electronic files or prevent the exploitation of their stolen information,” giving some legitimacy to attempts at breaching perpetrator’s network to incapacitate it or recover stolen digital property. However, The Office of Legal Education (2010, p. 180) states that “. . . the company should not take any offensive measures of its own. . . even if such measures could in theory be characterized as ‘defensive.’ Doing so may be illegal, regardless of the motive.” Active deterrence is a matter of ongoing academic discourse.

Measuring security is challenging because it involves trust, a degree of belief which is itself abstract. Denning (1987) described a real-time intrusion-detection (IDS) expert system

based on statistical evaluation of predefined anomalous activities in audit records, profiling legitimate users (insider threat) and external agents. It presupposes untampered electronic trail of evidence which does not hold in situations where separate backups are not created because the attacker must be assumed able to modify or delete logs after system compromise. Littlewood et al. (1993) provide an overview of existing approaches and suggest probabilistic requirements such as effort to instigate a security breach for a security metric. Stolfo, Bellovin, and Evans (2011, p. 2) admit that "...[t]he fundamental challenge for any computer security metric is that metrics inherently require assumptions and abstractions." They further analyze existing and hint at several new quantitative indices, namely automated diversity (relative), cost-based IDS (economic), polymorphic-engine strength (biological), and decoy properties and fuzzing complexity (empirical). Herzog (2010, p. 63) uses rav, "... a scale measurement of an attack surface, the amount of uncontrolled interaction with a target... [it] does not measure risk for an attack surface, rather it enables the measurement of it," for penetration testing presented in chapter 2.4.8.

Orlandi (1991, p. 2) reviews the concept of Information Economics applied by M. M. Parker, Benson, and Trainor (1988) which disrupts "... the common practice that mixes the business justification for [i]nformation [t]echnology, IT, and technological viability," and attempts to provide security cost-benefit analysis. It was pointed out that linking economics and security suffers due to microeconomic effects of network externalities, asymmetric information, moral hazard, adverse selection, liability dumping, and the tragedy of the commons (R. Anderson, 2001). Varian (2001) considered three scenarios where security was considered a different type of good and free riding was mathematically-proven result in each one. An indicator titled ROISI (Return on Information Security Investment) has been proposed (Mizzi, 2005) but is not mandatory in financial statements.

There are two prevailing views on how security should be treated: making decisions utilizing cost-benefit analysis (Gordon & Loeb, 2005) dependent on estimating probabilities of losses, and empirical or metrics-based (Jaquith, 2007) using quantitative tools and visualization to support the results. The first is preferred for its closeness to traditional economic paradigm and ease of understanding, the second for objectivity and effectiveness in expressing data without assuming known probability distributions of random variables in the model. The difference can be likened to frequentist and Bayesian statistics: the former estimates unknown priors while the latter favors building test cases to extract the values from real-life experiments. Hayden (2010, p. 141) makes a case for incorporating qualitative analytical techniques, "... the concept of coding, or assigning themes and categories to the data and increasingly specific levels of analysis," into security. He further advocates changing the top-down (applying a metric and subsequently assigning it interpretative meaning) approach to bottom-up (defining a goal, then finding a tool to measure it).

In cybernetics, cybersecurity (portmanteau of cybernetics and security) aims to protect signals exchanged between system parts from unauthorized access, interception, modification, or destruction which would negatively affect system stability had one or several of such actions occurred. Cybersecurity strives to prevent introduction and injection of counterfeit signals from within or outside the system with damaging properties in place of genuine ones. As computers have not been built with security as their primary requirement, some level of risk acceptance must be tolerated until initiatives such as trusted systems or Trusted Computing proliferate (Mitchell, 2005). Limited user-level modifications with Trusted Computing enabled were pointed out (R. Anderson, 2004; Stajano, 2003), accentuating the necessity to consider many concerns in next-generation computer architectures.

Security is asymmetric, and susceptibility of ICT to exploitation precludes labeling any system unconditionally (i.e., not relying on unproven assumptions) secure at any given time. Several

notions of what constitutes its “best” level in relation to encryption schemes, a necessary but insufficient condition for securing sensitive data, have been put forward: information-theoretic, semantic, reasonable, etc. Information-theoretic security assures the system is capable of withstanding attacks from an adversary with unlimited computational resources, making the data inaccessible even when outside of organization’s control (Y. Liang, Poor, & (Shitz), 2009). Semantic security relaxes the requirement and argues that even if some information about the data (but not their content) is revealed, the attacker is with high probability unable to use it to gain an advantage (Shannon, 1949). Reasonable security informally purports any method is suitable as long as it ensures the data remains encrypted until its relevance to the perpetrator is lost, or becomes obsolete. Knowledge-based authentication tokens (passwords) are, depending on the algorithm, at least reasonably secure if no data is recovered before the password-changing policy comes into effect. While the first two principles are theoretical, the third one is popular in real-world situations where practical considerations (convenience, ease of use, cost, maintenance) prevail over implementation of provably secure but resource-intensive measures. The Office of the Australian Information Commissioner (2013, p. 16) claims that “. . . ICT measures should also ensure that the hardware and the data stored on it remain accessible and useful to legitimate users.”

Neumann (2003, p. 3) posits that “. . . [i]n most conventional systems and networks, a single weak link may be sufficient to compromise the whole.” This was corroborated by a report (Mandiant, 2013, p. 1) stating that “[a]dvances in technology will always outpace our ability to effectively secure our networks from attackers. . . Security breaches are inevitable because determined attackers will always find a way through the gap.” A factor influencing system resilience is time: while it can be considered secure, a previously-unknown vulnerability (a weak link) creates a window of opportunity, and makes ICT considerably less secure if not fixed. Response should thus be prompt to minimize probability of exploiting the attack vector.

Two parties (defender and adversary) wishing to maximize their utility function at the expense of the other allows to model the situation using game theory, specifically rational two-player zero-sum adversarial games (X. Liang & Xiao, 2013). Nevertheless, some attackers may exhibit seemingly irrational behavior by not trying to increase their utility in a single turn, instead directing their resources toward high-value targets, e.g., users with administrative permissions on their devices. By focusing on this subset, they forgo maximization of present utility (accomplished by concentrating on users with low permissions) for a promise of possible future benefit outweighing the current one, an instance of economic marginal rate of time preference denoting “. . . a measure of . . . impatience. . . The higher the value. . . , the smaller is the utility. . . [derived] from consumption in the future” (Besanko & Braeutigam, 2010, p. 148).

Interrelation between various aspects of security is depicted in Figure 6. Within this framework, security is defined as “. . . the degree to which *malicious* harm to a valuable asset is prevented, reduced, and properly responded to. Security is thus the quality factor that signifies the degree to which valuable assets are protected from significant threats posed by malicious attackers” (Firesmith, 2004, p. 3).

The taxonomy builds up on existing infrastructure to first identify valuable sources for which risks are identified assuming theoretical (supposed) or practical (known) capabilities of an adversary. Subsequently, assessment of vulnerabilities constituting the attack surface, “[t]he lack of specific separations and functional controls that exist for that vector,” (Herzog, 2010, p. 21) is formulated. In context of information asset protection, security is considered a superset of confidentiality, integrity, and availability.

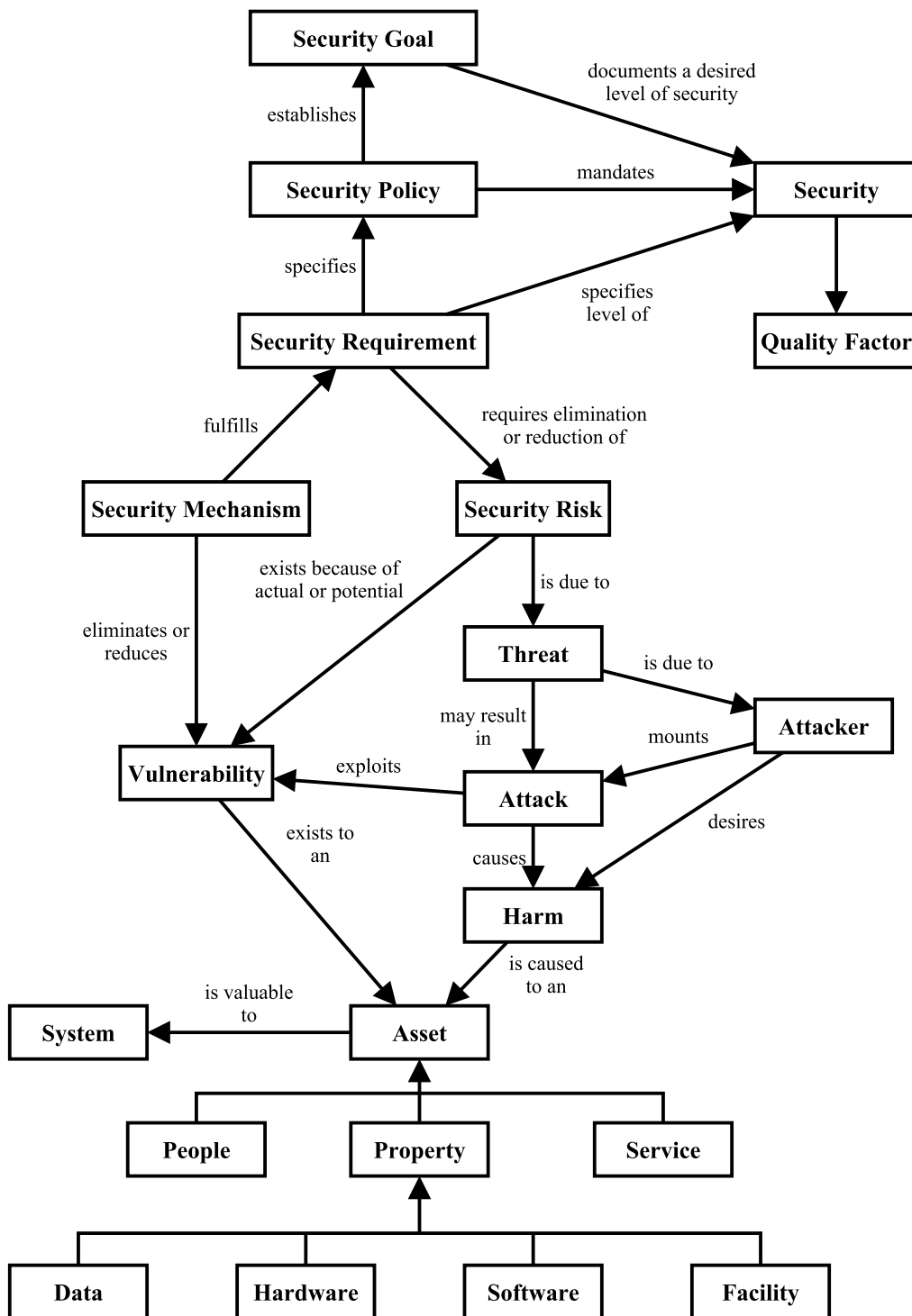


Fig. 6: Security taxonomy. Security is a direct product of a requirement, policy, and a goal with the requirement determining the other two while being influenced by factors further down the graph. The lower the position within the hierarchy, the more specific meaning. Source: Firesmith (2004), modified.

2.2 The CIA Triad

Sensitive electronic information may pertain to employees, customers, suppliers, business partners, and contain financial, personal, medical, or other data used to identify the subject. Most companies are connected to the Internet and accessible globally, attacks may thus originate in geographic proximity and across borders, making legislative actions and prosecution challenging due to atomized legal systems and lacking cooperation among sovereign countries. Some attempts to codify growing dependence and ICT risks have come to fruition, non-governmental organizations and commercial entities have proposed frameworks to mitigate risks associated with sensitive data retention, and corporations are bound to create and communicate risk prevention policies to comply with law.

The chapter will introduce the CIA triad as a set of information security principles. Further described will be how misappropriation of information could influence competitiveness of an organization targeted and breached. Specifics mobile technology whose integration with ICT infrastructures is a trend closely tied to the CIA triad will be analyzed in the next chapter.

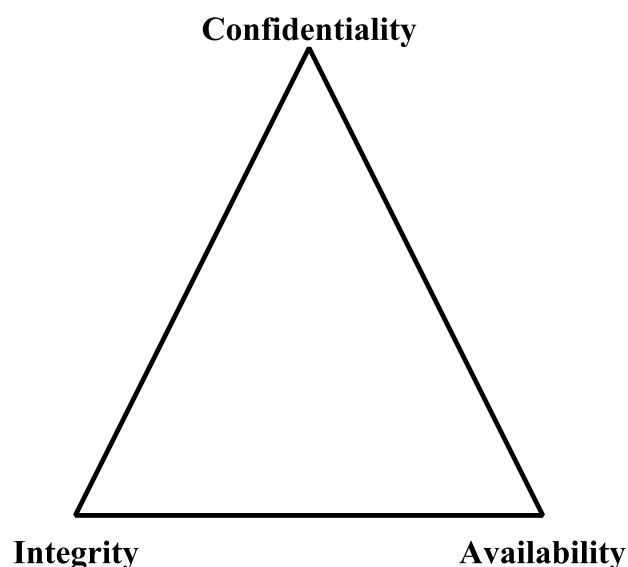


Fig. 7: The CIA triad. All three constituents should be balanced to ensure optimum level of security without incurring overheads when manipulating with sensitive assets.
Source: Walsh (2012), modified.

Information security protects systems from unauthorized access, modification, disruption, or other types of activities not endorsed by their owners. At the same time, legitimate users must be allowed access as information enters the transformation phase of organizational processes and despite being confidential, must be available with assurance of its integrity. The crux of the CIA triad is to assure mediation among the Confidentiality, Integrity, and Availability factors to support efficient, secure sharing and use of information. The triad is "... the concept of securing data... [T]o guarantee these principles are met... administrative, physical, and technical controls [are used] to provide a secure environment" (Stone & Merrion, 2004, p. 3). It is the most well-known framework cited in information security (Dhillon & Backhouse, 2001; Oscarson, 2007; Gollmann, 2011; S. Harris, 2012) and is frequently employed to assess steps for securing information systems. However, the scheme has been deemed obsolete and insufficient for complex ICT infrastructures associated with electronic business, medical records, and government (Wu, 2007) due to omitting critical properties such as utility and possession,

shortcomings addressed by Parkerian Hexad (D. B. Parker, 1998). Even Parkerian Hexad omits non-repudiation, though, an important property for financial transactions and digital signature schemes ubiquitous on the Internet.

Implementing all of these principles should respect ease of use while not compromising confidentiality. While it would be tempting to protect assets by several layers of security, concessions must be made to accommodate growing demand for remote connections from untrusted devices via unencrypted channels or shared computers. On the other hand, granting full availability for every employee can be exploited to surreptitiously gain entry by targeting human element using phishing, as described in chapter 2.4.5. Prioritizing integrity which is largely based on cryptographic hash functions creates processing overheads which may result in slowdowns, unexpected behavior, and crashes. Adversaries could also trivially saturate system resources by repeatedly forcing verification operations, a form of denial-of-service attack described in chapter 2.4.4. Therefore, setting parameters of the three factors so that infrastructure stability is ensured requires testing and setting priorities. The triad is schematically depicted in Figure 7.

2.2.1 Confidentiality

Confidentiality presupposes there exists an asset which “(i) . . . must be secret, i.e., not generally known or readily accessible to persons that normally deal with that kind of information; (ii) it must have commercial value because it is secret; (iii) the owner must have taken reasonable steps to keep it secret” (Irish, 2003). It then signifies “[p]reserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information” (McCallister, Grance, & Scarfone, 2010, p. 53). When an organization compartmentalizes assets into categories, each should be assigned a security level and protected accordingly. The notion that all information should be protected at all times (i.e., uniform confidentiality) is flawed; such scheme would require excessive resources while decreasing availability and adding complexity. It is reasonable to designate at least one category for publicly available or no-control sources which need not be monitored; one to include highly-sensitive and top-secret sources access to which must be logged in real time to detect intrusion attempts; and one or more categories with data for production environment access to which should be verified using per-user tokens, optimally in two- or multi-factor fashion. Perrin (2008) states that “[o]ne of the most commonly used means of managing confidentiality on individual systems include traditional Unix³ file permissions, Access Control Lists (ACLs), and both file and volume encryption.”

File permissions are connected to users and groups. Operating systems contain sets of rules to designate individuals or their sets as eligible to access specified files or folders, managed by an administrative entity with complete control. Users are not allowed to modifications themselves as their system permissions are set lower than those of the administrative entity, making for a separation titled *the principle of least privilege*. It states that “. . . [e]very program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job. . . [so] that unintentional, unwanted, or improper uses of privilege do not occur” (Saltzer, 1974, p. 2). Static assignment sometimes preclude services or programs from working correctly, and privilege separation which “[m]any applications are designed to take advantage of. . . to ensure they do not pose a significant security threat to the rest of the system even if they are compromised. . .” (Perrin, 2009) by partitioning the code with different level of privilege granted, is used instead. Also, due to strict separation of roles, security is centralized and

³[ˈjuːnɪks], *noun*: an operating system that can be used by many people at the same time (Oxford University Press, 2011)

controlled by one or at most several operators, ensuring accountability and redundancy in case of several operators. Permissions are part of a broader confidentiality concept known as access control which "...regulates the limitations about the access of the subject to the object, and controls the request of resource access according to the identity authentication. . . [It is] the most important and basic security mechanism in the computer system" (Quing-hai & Ying, 2011, p. 1).

Carter (2003) purports authentication "...is the mechanism whereby systems may securely identify their users," as opposed to authorization which "...is the mechanism by which a system determines what level of access a particular authenticated user should have to secured resources controlled by the system." Authentication is a set of ownership, knowledge, and inherence factors by which an entity proves its identity or property to the receiver using passwords, biometric scanners, one-time passwords (OTP), physical tokens, or radio-frequency identification (RFID) chips separately or in combination. Each is unconditionally trusted and if the procedure is finalized correctly, no further checks are made to determine whether the feature was compromised. Authorization works based on lists for both authenticated and unauthenticated user groups stating which permissions were granted for sensitive information assets. Physical procedures comprise signatures, identity documents, visual and human-facilitated verification, access lists, locks, and others.

The third element of confidentiality is encryption, closely related to a notion of privacy, defined as "... the condition of not having undocumented personal knowledge about one possessed by others," (Parent, 1983, p. 1) or "...the right to be left alone," (Warren & Brandeis, 1890) although its precise meaning is a matter of legal, philosophical, and social discussions (A. Moore, 2008) especially as technology increasingly integrates into society. Digital privacy emerged as a reaction to sensitive assets handled, processed, stored, accessed, and entrusted to connected computer systems where individuals lose direct control over how the information are secured. It is defined as "... [the] right to keep a domain around us, which includes all those things that are part of us, such as our body, home, thoughts, feelings, secrets and identity. The right to privacy gives us the ability to choose which parts in this domain can be accessed by others, and to control the extent, manner and timing of the use of those parts we choose to disclose" (Onn et al., 2005, p. 12). Personal data, albeit purportedly anonymized, is sold by brokers for variety of purposes (Opsahl & Reitman, 2013), turning privacy into economic good (Zhan & Rajamani, 2008).

Encryption is the process of encoding messages (or information) in a way that prevents eavesdroppers from reading them, but authorized parties can (Goldreich, 2004). The source (plaintext) is converted using a mathematical function (encryption algorithm) to output (ciphertext) unreadable to anyone not possessing a means (key) to invoke a reverse operation (decryption) in order to obtain the original message. Attempts to circumvent the need for a key are resource-intensive and despite being always finite, the time required for successful extraction is counted in centuries or millennia. Encryption is a subset of cryptography, "... the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication" (Menezes, van Oorschot, & Vanstone, 1996, p. 4). Menezes et al. (1996) mention many cryptographic algorithms exist to guarantee asset protection, differing in level of security, functionality, methods of operation, performance, and ease of implementation. Level of security, performance, and ease of implementation in particular are exploitable as summarized in Table 1.

Encryption augments authentication: entities possessing the key are permitted to view, modify, or execute operations on information, making them contingent on the key distribution scheme which can be used instead of access lists. The scheme increases processing overhead, though, as data for each user group has to be encrypted separately. The arrangement is thus suitable for small

Tab. 1: *Edge cases of selected encryption properties. Confidentiality is undermined or compromised if some are deliberately set to high or low levels, making the maximization objective of a single one counterproductive.*
 Source: own work.

Property	Low	High
Level of security	Resource-intensiveness skewed in favor of the attacker who can decrypt data in “reasonable time.”	Performance penalty incurred to the system
Performance	Repeated encryption commands can be issued to make hardware inaccessible or unusable for users due to saturation.	Advanced optimization required
Ease of implementation	Proneness to failures in production environments, expertise beyond organizational scope required.	Advanced settings hidden, opening attack vectors if the defaults set improperly

amount of disjunctive groups with storage and performance increasing linearly. Techniques has been developed which provide large-scale resource optimization: for example, single instancing (deduplication) “. . . essentially refers to the elimination of redundant data. In the deduplication process, duplicate data is deleted, leaving only one copy (single instance) of the data to be stored. However, indexing of all data is still retained should that data ever be required” (Singh, 2009). Concerns were raised about performance penalties of single instancing (Connor, 2007) which may strain hardware.

In summary, confidentiality means “. . . [a] requirement that private or confidential information not be disclosed to unauthorized individuals” (Guttman & Roback, 1995, p. 7). It mainly utilizes encryption and aims to create a balance between ease of access, use for legitimate parties, and security.

2.2.2 Integrity

Integrity is “. . . the representational faithfulness of the information to the condition or subject matter being represented by the information,” (Boritz, 2003, p. 3) and is partly related to a set of properties for database transactions known as ACID: Atomicity, Consistency, Isolation, and Durability. A database “. . . is a shared, integrated computer structure that stores a collection of [e]nd-user data, that is, raw facts of interest to the end user, [and] [m]etaddata, or data about data, through which the end-user data are integrated and managed” (Coronel, Morris, & Rob, 2009, p. 6). Instantiating a transaction, “. . . a short sequence of interactions with the database. . . which represents one meaningful activity in the user’s environment,” (Haerder & Reuter, 1983, p. 3) is useful for preserving integrity as it allows tracking discrete changes to the asset by entities performing operations on it simultaneously. Definitions of individual ACID axioms are provided in Table 2. Some database systems are fully ACID-compliant even in parallel, multi-transaction environment, others focus on strict subsets of the criteria.

Tab. 2: *Definition of ACID axioms. While Gray (1981) did not explicitly delimit isolation, he nevertheless described it.*

Source: own work.

Element	Definition	Source
Atomicity	“[The transaction] either happens or it does not. . .”	Gray (1981, p. 3)
Consistency	“The transaction must obey legal protocols.”	Gray (1981, p. 3)
Isolation	“Events within a transaction must be hidden from other transactions running concurrently.”	Haerder and Reuter (1983, p. 4)
Durability	“Once a transaction is committed, it cannot be abrogated.”	Gray (1981, p. 3)

While it is necessary to apply safeguards enforcing integrity on a database level, concurrency control, “. . . the activity of coordinating the actions of processes that operate in parallel, access shared data, and therefore potentially interfere with each other,” (P. A. Bernstein, Hadzilacos, & Goodman, 1987, p. 1) and strict ACID specifications led to arguments integrity cannot be maintained fully. Specifically, distributed computing systems were posited to have at most two of the three properties: consistency, availability, and tolerance to network partitions, the so-called CAP theorem (Gilbert & Lynch, 2002). A relaxed model, BASE (Basically Available, Soft-state, Eventual consistency) was devised sacrificing consistency and isolation for availability, graceful degradation, and performance (Brewer, 2000). Pritchett (2008, p. 4) states that “[w]here ACID is pessimistic and forces consistency at the end of every operation, BASE is optimistic and accepts that the database consistency will be in a state of flux. . . through supporting partial failures without total system failure.” This is a suitable approach for organizations where many entities perform tasks on identical data at the same time, and in Internet settings. Simple and concurrent database transactions are demonstrated in Figure 8.

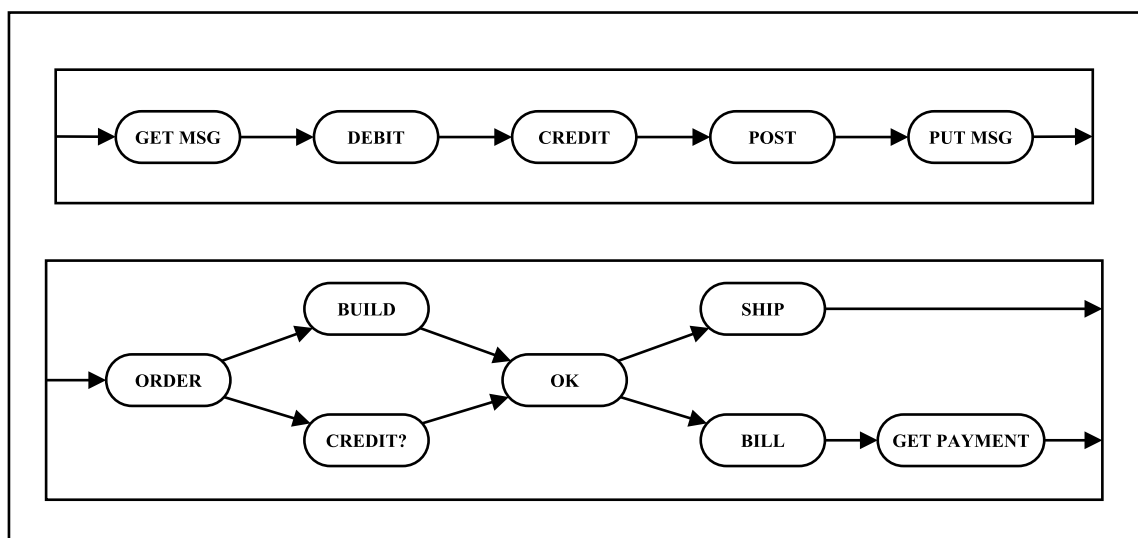


Fig. 8: *Types of database transactions. “A simple transaction is a linear sequence of actions. A complex transaction may have concurrency within a transaction: the initiation of one action may depend on the outcome of a group of actions.”*

Source: Gray (1981), modified.

Integrity can be considered a special case of consistency applied to physical and digital assets. Physical asset protection is sometimes treated as secondary but as strong encryption and best practices are implemented, perpetrators exploit alternative attack vectors to gain leverage into the system. Organizations dealing with sensitive assets should employ measures preventing disclosure, misappropriation, false asset introduction (e.g., supplanting a document with a modified variation) or destruction of information:

- hand-baggage screening,
- removable media policy,
- identification from and declaring length of stay for any third party who should be accompanied at all times while on the premises,
- closed-circuit television (CCTV) feed with real-time monitoring and response,
- perimeter control,
- suitable lighting in areas where assets are being processed,
- physical tokens and access codes to areas hosting critical ICT infrastructure,
- clean desk policy,
- printers, faxes, copy machines, and personal computers accessible exclusively to authorized parties,
- secure destruction, including information in original size, optical, magnetic, and electronic data media, information in reduced form, and hard drives with magnetic data media (HSM, 2012).

Two widely-used tools to prevent electronic asset corruption are metadata and verification using hash or checksum algorithms. Metadata “. . . refers to data about the meaning, content, organization, or purpose of the data. Metadata may be as simple as a relational schema and or as complicated as information describing the source, derivation, units, accuracy, and history of individual data items” (Siegel & Matnick, 1991, p. 3). Bagley (1968, p. 26) first introduced structural metadata: “As important as being able to combine data elements to make composite data elements is the ability to associate explicitly with a data element a second data element which represents data ‘about’ the first data element. This second data element we might term a ‘meta-data element’.” Alternatively, NISO (2004, p. 1) states structural metadata “. . . indicates how compound objects are put together. . .” as opposed to descriptive metadata which “. . . describes a resource for purposes of discovery, identification etc. It can include elements such as title, abstract, author, and keywords,” and administrative metadata which “provides information to help manage a resource, such as when and how it was created, file type and other technical information.” Alternative delimitations also exist for specialized applications such as data warehouse deployment (Kimball, Reeves, Ross, & Thornthwaite, 1998; Bretherton & Stingley, 1994).

The International Organization for Standardization understands metadata as “data that defines and describes other data,” ISO (2012b, p. 4) making them suitable for storage in databases either internally, embedded within the object they refer to, or externally in a separate instance. The former approach is favored when redundancy and tight coupling to the source are required, the latter for aggregation and analyses because metadata can be grouped and manipulated with easily. External storage in particular complements integrity because separating information assets from metadata reduces risk of unauthorized modifications, e.g., rewriting document’s author in the file and the descriptor field. When enhanced with checksums and designated as authoritative in case of inequivalence, metadata enable effective monitoring and version control. However, an adversary can automate metadata-enriched asset collection to map organizational structure, names, roles, software base, and other specificities during preparations for an attack.

Cryptographic checksums are a class of functions purpose of which is to generate numeric outputs uniquely fingerprinting input data. The product does not store any information about the

source and can only be used for comparison or error correction. F. Cohen (1987, p. 1) stipulates: “Historically, integrity protection mechanisms have been designed to protect transmitted information from illicit creation, deletion, and/or modification, but. . . integrity protection in information systems and networks may be a more pressing problem than was ever previously suspected.” They should have several features:

- checksum does not reveal any information about the data block on which it was calculated,
- different data blocks will generate different checksums with overwhelming probability,
- identical data blocks will produce identical checksums every time,
- low computational and storage demands for routine use,
- metadata integration,
- short-length checksum must be capable to validate long-length data block.

Optionally, the checksum algorithm should exhibit avalanche effect, described by Feistel (1973) and Kam and Davida (1979) but going back to Shannon (1949): a change in elementary unit (a bit) in a message produces a change of some units in the checksum. In case of a strict avalanche criterion (SAC), “. . . each output bit should change with a probability of one half whenever a single input bit is complemented,” (Webster & Tavares, 1986, p. 2) making the change on each position random and unpredictable. Avalanche effect for MD5 and SHA-1 functions, widely used to generate checksums, is demonstrated in Figure 9.

Data	MD5	SHA-1
000	C6F057B86584942E415435FFB1FA93D4	8AEFB06C426E07A0A671A1E2488B4858D694A730
001	DC5C7986DAEF50C1E02AB09B442EE34F	E193A01ECF8D30AD0AFFEFD332CE934E32FFCE72
011	84EB13CFED01764D9C401219FAA56D53	E7001334D9D19559A8BB0DD6015F16E31D15566C
111	698D51A19D8A121CE581499D7B701668	6216F8A75FD5BB3D5F22B6F9958CDEDE3FC086C2

Fig. 9: *Avalanche effect. A single-digit change in input results in vastly different outputs. The characteristic is desirable for integrity as it reduces probability of forging checksums based on unknown plaintext. Source: Own work.*

One-way hash function, “. . . a function, mathematical or otherwise, that takes a variable-length input string. . . and converts it to a fixed-length (generally smaller) output string (called a hash value),” (Schneier, 1996, p. 30) is a tool frequently employed to compute checksums. Supported in database management systems (DBMS), operating systems, and software, they have become popular due to their speed and ease of use. Because stored hash values do not leak anything about the data for which they were calculated, they are preferred for handling authentication requests based on passwords: when a candidate string is submitted, its hash is computed and compared to a database entry corresponding to the user account which initiated the request. If a positive match is made, it is highly probable the password is correct. Storing sensitive data in its original form for comparison purposes has been broadly discouraged as bad practice (Moertel, 2006; Atwood, 2007; Boswell, 2012; Evron, 2012; Nielsen, 2012; Kozubek, 2013) which leads to a compromise should the intruder penetrate the system and create a copy of the database. However, some algorithms previously considered suitable have been rendered susceptible to attacks resulting from advances in hardware performance, parallel computing, decreasing storage costs, and novel theoretical findings. Despite their shortcomings, MD5 and SHA-1 are routinely deployed as default solutions for protecting information assets even though neither complies

with a fundamental security axiom titled Kerckhoffs's principle. It states the system must not rely on secrecy but rather features of its parts⁴ (Kerckhoffs, 1883; Petitcolas, 2013). An attack on MD5 will be presented in chapter 2.4.2 and case study 1 in chapter 5.1. Unlike dedicated checksum algorithms, hashes do not provide error correction.

An important application area of hash functions is ensuring integrity of sensitive data in transit over the network or any channel which does not involve direct physical exchange between the sender and the receiver. The terms were introduced by Shannon (1948) in a communication system depicted in Figure 10. Any channel apart from direct exchange is considered noisy with non-zero probability of the transferred data modified by factors outside of sender's and receiver's control; in a security setting, the adversary can either passively intercept all communications, or actively attempt to change its content.

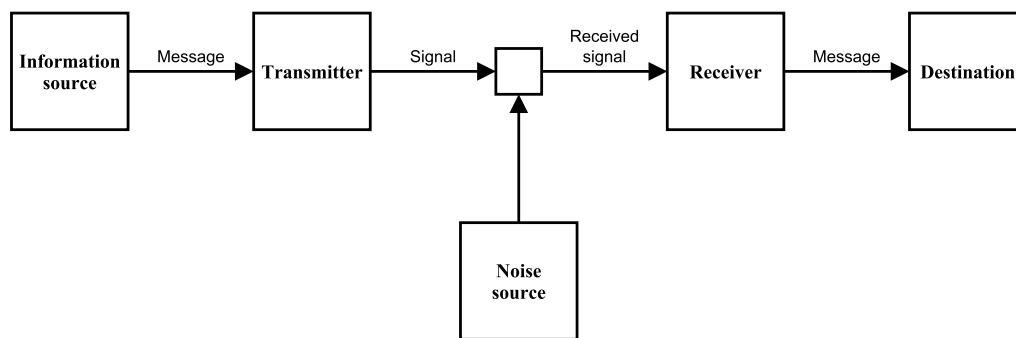


Fig. 10: *General communications model. In the scheme, sum of factors influencing communication over a channel is designated as noise.*
 Source: Shannon (1948), modified.

When a fingerprint is provided by the sender, the receiver can determine if the asset was modified during transmission. To demonstrate the technique practically, MD5 and SHA-1 hash values computed for the previous sentence sender-side are 2935753095b3d9c72407cfea7df4c370 and c231ceda79bf7364fc40991b87308cef557fcf59, respectively; if they differ receiver-side, they were modified without authorization and should be considered invalid.

One-way hash functions are advantageous only for assets which need not be stored in original form. For routine operations where information serves as inputs to business processes, handling it scrambled is not preferred or appropriate because it needs to be readily available. When an asset is requested, loaded, and transmitted from a protected medium (database, network storage), system integrity which "... seeks to ensure security of the system hardware, software, and data" (Microsoft, 2005) comes to the fore. A closed system inaccessible over the network may be assumed to automatically provide integrity of all information (signals) exchanged within due to non-existent third-party threat. As long as physical access is allowed, however, the threat is still relevant. Organizations are located in an internetworked environment (chapter 2.1.1), which in addition to the asymmetric security model (chapter 2.1.5) places high demands on ICT incident response and prevention. Tools and methods usually deployed to support system integrity are:

- access control,
- antivirus software,
- event auditing, logging,
- firewalls,

⁴Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi. (The system must not require secrecy and can be stolen by the enemy without causing trouble.)

- full-disk encryption (FDE),
- Intrusion Detection Systems (IDS),
- physical separation from the network (air gap),
- protocol extensions (DNSSEC, FTPS, HTTPS, IPsec),
- security-focused operating systems,
- virtualization.

Even in combination, they cannot guarantee system integrity as they only focus on technology, not the human element (customers, employees, suppliers, users, guests). Training, demonstrations, concise security policies, encouraging questions and suggestions as well as monitoring and incorporating preventive measures for novel threats must be also added to the ICT priorities.

In summary, Guttman and Roback (1995, p. 6) define data integrity as “. . . a requirement that information and programs are changed only in a specified and authorized manner,” and system integrity as a state where the system “performs its intended function in an unimpaired manner, free from deliberate and inadvertent unauthorized manipulation. . . .” Various programmatic means are used to ensure integrity which should be combined with measures dealing with physical access and handling of sensitive assets for optimal results.

2.2.3 Availability

Availability is the “[a]bility of an IT service or other configuration item to perform its agreed function when required. [It] is determined by reliability, maintainability, serviceability, performance and security” (ITIL, 2011, p. 7). COBIT (2012, p. 82) categorizes availability as a security/accessibility quality, and defines it “[t]he extent to which information is available when required, or easily and quickly retrievable.” Finally, IEEE (1990, p. 24) specifies availability as “[t]he degree to which a system or component is operational and accessible when required for use. [It is] often expressed as a probability.” This corresponds to Barlow and Proschan (1981, p. 190): “An important figure of merit for a system. . . is the probability that the system is operating at a specified time t .” Availability is also understood as “[a] measure of the degree to which an item is in an operable and [committable] state at the start of a mission when the mission is called for at an unknown (random) time” (DoD, 1981, p. 1). It belongs to reliability engineering, a field of study dealing with “[t]he ability of a system or component to perform its required functions under stated conditions for a specified period of time” (IEEE, 1990, p. 170). In ICT, a reliable system “. . . virtually never loses data, unless certain forms of catastrophic failures occur, and. . . it is always able to recover data to a desired consistent state, no matter what complicated forms of one or multiple failures arise” (Weikum & Vossen, 2002, p. 27).

An asset whose confidentiality and integrity has been assured is unusable if it’s not available as input to business processes. Tied to error tolerance, fault tolerance, and robustness, availability is a probabilistic concept taking into account both expected and unexpected events influencing system’s ability to deliver data in a desired form while spending minimum amount of time and resources. In many organizations, high availability “. . . which implies that recovery times after failures are short, and that failures that lead to total outages are infrequent” (Weikum & Vossen, 2002, p. 27) is a requirement with severe financial stake: The Standish Group (1999, p. 1) estimated that “. . . 6% of all application outages are caused by database failures. This equates to a \$30 billion cost-of-downtime per year.” A breakdown demonstrating downtime hierarchy is depicted in Figure 11. Both preventive and corrective measures must be factored in when considering maintenance: the former refers to a scheduled outage plan which need not affect resource availability, corrective measures occur due to hardware degradation, software instability, power failures, human error, and others.

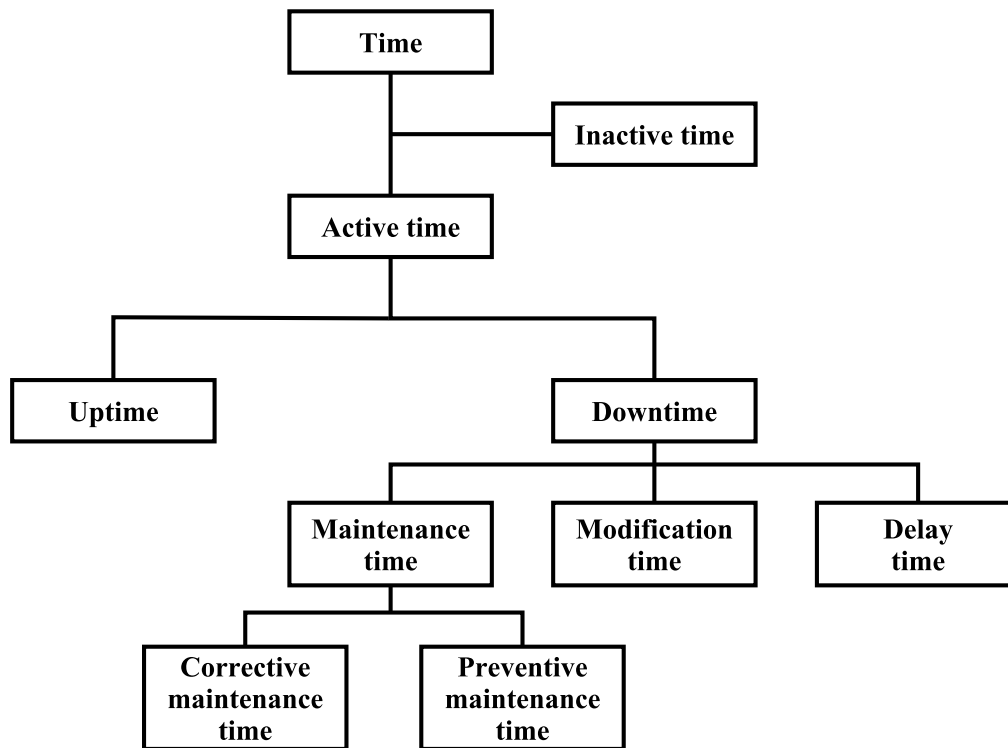


Fig. 11: Availability breakdown. Corrective maintenance is a period spent making the system operational after an unforeseen event caused a downtime, preventive maintenance is a period spent improving system resilience to reduce probability of future unscheduled downtimes. Source: DoD (1981, p. 13), modified.

Preventive maintenance “... generally requires between one and four hours of scheduled downtime per year. . .,” (Liebert Corporation, 2003, p. 1) which limits availability if no backup is in effect. For systems handling multiple services and large volume of requests simultaneously, e.g., financial markets, online shopping portals, search engines, and social networks, downtimes or increased latencies have tangible business consequences in decreasing revenue, user satisfaction, or lost sales, and should be avoided or reduced. Martin (2007) estimates that “[a] 1-millisecond advantage in trading applications can be worth \$100 million a year to a major brokerage firm. . .” but even organizations which do not require high speeds benefit from infrastructure supporting fast recovery mechanisms in case of partial or complete dropouts. System availability also “... depends on the status of its components, which should be reliable or available during the processing period” (Liu, Pitoura, & Bhargava, 1995, p. 7). A server running a database platform or mission-critical applications to which authenticated entities are authorized to send requests for data acquisition or processing has multiple hardware components prone to failures. These include central processing unit (CPU), cooling units, hard-disk drive (HDD), motherboard, network links, power supply unit (PSU), random-access memory (RAM) modules, and uninterruptible power supply (UPS).

Data centers housing “... a broad range of services such as Web search, e-commerce, storage backup, video streaming, high-performance computing, and data analytics” (Gill, Jain, & Nagappan, 2011, p. 1) experience frequent component replacements due to high concentration of servers (upwards of 100 000) located on the premises. For compatibility and cost reasons, clusters often leverage commodity hardware over specialized solutions (Al-Fares, Loukissas, & Vahdat, 2008). In fault-tolerant systems, the user does not know a failure occurred as asset availability is maintained despite changes in network routing topology, hardware defects, and software crashes; Greenberg, Lahiri, Maltz, Patel, and Sengupta (2008, p. 1) state that “... [i]nnovation

in distributed computing and systems management software have enabled the unreliability of individual servers to be masked by the aggregated ability of the system as a whole.” The most important techniques facilitating seamless access include backups, failover, load balancing, and virtualization. Cloud computing will be mentioned first as a cost-effective availability alternative to ICT ownership.

Outsourcing ICT-related activities outside the organization in part or whole is termed Business Process Outsourcing (BPO) or Knowledge Process Outsourcing (KPO). It “...refers to the process of consigning duties and accomplishing determined duties by an enterprise to other that usually accomplishes by a third provider” (Alipour, Kord, & Tofighi, 2011, p. 1). A framework for evaluating benefits of KPO has not been agreed upon yet (Willcocks, Hindle, Feeny, & Lacity, 2004). Outsourcing was first hinted at by Coase (1937, p. 11) who argued that “a point must be reached where the loss through the waste of resources is equal to the marketing costs of the exchange transaction in the open market or to the loss if the transaction was organised by another entrepreneur... [A] firm will tend to expand until the costs of organising an extra transaction within the firm become equal to the costs... of organising [it] in another firm.” Outsourcing has become practiced in administration, assembly, facility services, ICT, and research and development. Two issues to consider is how much of a given product or service should the firm outsource (degree of outsourcing or boundary of the firm), and in what manner should the firm manage its relationships with outside suppliers (governance structure) while respecting trends in ICT: decreasing average unit cost and increasing economies of scale, information availability, processing capacity, standardization and interconnection (Clemons, Reddi, & Row, 1993). Rightsourcing, “...knowing what activities to outsource and... how to structure these activities so that they can be outsourced most effectively” (Aron, Clemons, & Reddi, 2005, p. 1) reflects on these needs and lists outsourcing risks together with ways to mitigate them. Selective sourcing “... characterized by short-term contracts of less than five years for specific activities” (Lacity, Willcocks, & Feeny, 1996, p. 1) aims to avoid vendor lock-in, “... consumers’ decreased propensity to search and switch after an initial investment” (Zauberman, 2003, p. 1). Outsourcing risk is classified into several categories: information security and privacy, hidden costs, loss of management control, employees’ morale problems, business environment, and vendor issues. Benefits include cost savings, focus on core competencies, flexibility, access to skills and resources, service quality, and product and process innovations (Perçin, 1993).

Two main types of ICT outsourcing are on-site and off-site provisioning: in the former, an organization leases or purchases hardware and software from a third party while exercising physical control over where it is located, off-site provisioning is a model in which the infrastructure is maintained off the premises in one or several places. Both have benefits and challenges: downtime, initial and ongoing expenses, scalability, “... the measure of a system’s ability [to] respond to increased or decreased workload with minimal, or no manual intervention required,” (Lee, 2011, p. 3) security, single point of failure, speed of operation, and TCO has been mentioned (EPA Cloud, 2011). A combination of on-site and off-site provisioning “... can simplify both on-premise and off-site backups, while reducing the costs and increasing the reliability... Typically, these solutions keep large files, like databases and system state file backups on-site... [ensuring] a quick recovery to the latest versions of these files and reduce downtimes. All other files and data types are sent to the vendor’s remote cloud data centers” (Mueller, 2012). The cloud platform is “... a large pool of virtualized resources (such as hardware, software, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs” (Vaquero, Rodero-Merino, Caceres, & Lindner, 2009, p. 2). It is

an availability enabler for organizations in need of risk diversification stemming from loss of sensitive assets due to hardware failures and other unforeseen circumstances.

Cloud supports several technologies for availability assurance such as virtualization, load balancing, and backups. For the first one, Dong, Hao, Zhang, and Zhang (2010, p. 2) state that “. . . [w]hen a system is virtualized, its interface and resources visible through the interface are mapped onto the interface and resources of a real system actually implementing it.” Virtualization allows several independent virtual machines to share single hardware configuration, multiplying resource utilization by a factor of k , where $k \geq 1$ represents number of virtual instances on a single physical node so that “the failure of a single physical box will reduce the pool of available resources, not the availability of a particular service” (Rosenblum, 2004, p. 7). Simplified block diagram is depicted in Figure 12.

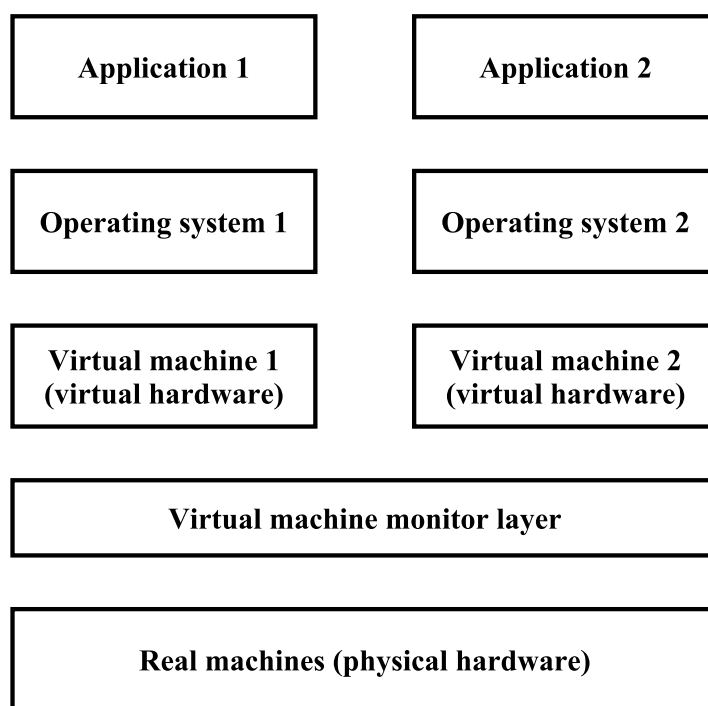


Fig. 12: Virtualization. Each underlying hardware can execute $k \geq 1$ number of virtual machines; for $k = 1$, the system need not be virtualized as resources are not shared with any other running instances. Here, $k = 2$.

Source: Rosenblum (2004, p. 3), modified.

Each virtual machine is strictly partitioned from others, although some data still tend to leak across, allowing the attacker to learn some information; the technique will be discussed in chapter 2.4.6. An organization wishing to minimize ICT-related costs can utilize single physical server running software and machines of all its users; however, this creates a single point of failure, “. . . [an] element that, if it failed, would have consequences affecting several things,” (Dooley, 2001, p. 31) which coupled with lacking backup policy could result in inadvertent data and productivity losses.

Load balancing is a concept in which databases are implemented “. . . in a clustered configuration in order to accommodate business requirements such as scalability, performance, high-availability and failure recovery,” (Hogan, 2009, p. 2) and is divided into two types: shared-nothing and shared-disk. Shared-nothing approach divides the database into discrete logical units (employees, payrolls, customers, products) with state of one independent on states of others. Aggregate availability is determined by availability of individual parts, “the load is spread

amongst servers or nodes on the basis of which server owns the data,” (Hogan, 2012, p. 8) and unplanned downtimes pose a serious challenge due to each database being a single point of failure. For high availability, shared-disk which “. . . enables any node to access the entire data set, so any node can service any database request,” (Hogan, 2012, p. 9) is optimal. The disadvantage is lower scalability: when n machines are added to the system pool, the number of inter-nodal messages increases to theoretically as much as $n \times (n - 1)$ as each server announces its presence to all others.

A remote cloud backup storage means “. . . delivery of virtualized storage and data services on demand over the network, based on a request for a given service level that hides limits to scalability, is either self-provisioned or provisionless, and is billed based on consumption” (SNIA, 2012, p. 20). This way, client’s assets are archived and geographically replicated as per cloud operator’s policy specified in a Service-Level Agreement (SLA) negotiated by both parties. Real-time accessibility, availability, and recovery necessitate active Internet connection, turning it into a single point of failure in case redundant links are not deployed. Since off-site backup utilizes cloud services, it faces the same challenges (downtime management, security and privacy, economic viability). Cloud offers inexpensive client-tailored settings but exponential progress of technology, first observed by G. E. Moore (1965) and later titled Moore’s law, continuously drives storage media prices downwards (Komorowski, 2009), making even large-scale hardware deployment an option for organizations not prepared to lose control over location of their data. Secure deletion and version management have been also mentioned as drawbacks (Rahumed, Chen, Tang, Lee, & Lui, 2011). Additionally, because the assets are duplicated across several data centers, proper deletion is not instantaneous, and unintended data losses at the cloud provider’s site are a possibility. An encryption-based method to make data inaccessible has been proposed (Geambasu, Kohno, Levy, & Levy, 2009) but the system can not be retrofitted to existing architectures. From a security standpoint, “[c]louds can comprise multiple entities, and in such a configuration, no cloud can be more secure than its weakest link. . . By their architecture’s inherent nature, clouds offer the opportunity for simultaneous attacks on numerous sites, and without proper security, hundreds of sites could be [compromised] through a single malicious activity” (Kaufman, 2009, p. 3). Another concern is unauthorized access to data by the cloud provider who “. . . will by definition control the ‘bottom layer’ of the software stack, which effectively circumvents most known security techniques” (Armbrust et al., 2010, p. 6). Furthermore, “. . . because moving large volumes of data quickly and cheaply over the Internet is still not practical in many situations, many organizations must send mobile data, such as an archive tape, to the cloud provider. It is critical the data is encrypted and only the cloud provider and consumer have access to the encryption keys” (IBM, 2011, p. 5). The provider is thus expected to be a trusted party.

Service-level agreements employ several metrics to quantify reliability: MTBF, MTTF, MTTR, and availability classes. Mean Time Between Failure (MTBF) “. . . is a reliability term used to provide the amount of failures per million hours for a product. This is the most common inquiry about a product’s life span, and is important in the decision-making process of the end user” (Stanley, 2011, p. 3). The metric is applicable to repairable and replaceable components (e.g, CPU, HDD) with finite repair time, whereas Mean Time to Failure (MTTF) is used to predict degradation in parts which are not replaced (infinite repair time). Both MTBF and MTTF were criticized for their simplicity, overly optimistic assumptions, and little resemblance to real-life conditions (Shroeder & Gibson, 2007; Elerath, 2000). Moreover, Schroeder and Gibson (2007, p. 3) observed that “. . . there is little indication that systems and their hardware get more reliable over time as technology changes.” The last metric, Mean Time to Repair (MTTR), is “. . . an estimated average elapsed time required to perform corrective maintenance, which consists of fault isolation and correction” (NASA, 1998, p. 2). Unlike MTBF and MTTF, MTTR should

be minimized as it is inversely tied to availability. All three metrics arithmetic mean sensitive to extreme values/outliers (i.e., low robustness), especially in small sample sizes. This renders mean “... not an appropriate measure of central tendency for skewed populations” (Manikandan, 2011, p. 2). Figure 13 demonstrates the classic hazard function to empirically model system life expectancy (servers, data centers) as well as hardware components, most of which operate in the second, intrinsic failure period ensuring the highest availability (Tobias, 2012).

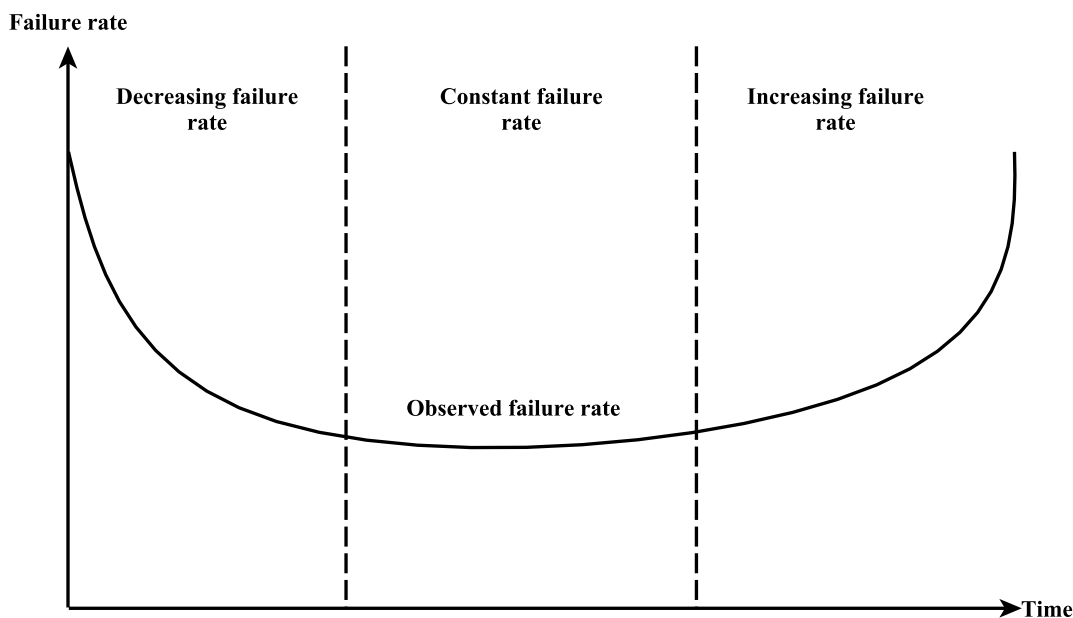


Fig. 13: Hazard function. The classic “bathtub curve” visualizes a three-stage failure probability model: decreasing (early failures), constant (random failures), and increasing (wear-out failures). Inverted curve would depict probability the system is available at a particular time. Source: Klutke, Kiessler, and Wortman (2003, p. 2), modified.

Availability classes discretize and mathematically express system’s ability to deliver assets at a given moment. Equation 2.2.1 demonstrates how a state can be uniquely described by elementary logical operators at time t with $X(t)$ representing the availability function. Downtime is often expressed in minutes or hours.

$$X(t) = \begin{cases} 1 & \text{if the system is available at time } t \\ 0 & \text{otherwise} \end{cases} \quad (2.2.1)$$

To aggregate availability (A) percentually, the following ratio is used:

$$A = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}} = \frac{\sum X(t) = 1}{[\sum X(t) = 1] + [\sum X(t) = 0]} \quad (2.2.2)$$

So-called availability classes “... each of which is defined by the number of leading nines in the availability figure for a system model; i.e.:

$$\left\lceil \log_{10} \left(\frac{1}{1 - A} \right) \right\rceil \quad (2.2.3)$$

where A is the system availability,” (Liu et al., 1995, p. 6) determine percentage of total time during which hardware and software perform up to specifications. The higher the availability class, the higher the system maintenance costs, over-provisioning, and redundancy which in turn increase the SLA-specified costs. To calculate service credits, a standard was devised which “...describes criteria to differentiate four classifications of site infrastructure topology based on increasing level of redundant capacity components and distribution paths” (Turner, Seader, & Renaud, 2010, p. 2). Benson (2006, p. 5) proposes four tiers:

- Tier 1 – Basic: 99.671% availability (annual downtime of 28.8 hours),
- Tier 2 – Redundant Components: 99.741% availability (annual downtime of 22.0 hours),
- Tier 3 – Concurrently Maintainable: 99.982% availability (annual downtime of 1.6 hours),
- Tier 4 – Fault Tolerant: 99.995% availability (annual downtime of 0.4 hours).

Alternatively, “number of nines” provides information on cloud and indirectly on asset availability: three, four, and five nines represent 99.9%, 99.99%, and 99.999% availability, respectively. Factors involve scheduled or unscheduled downtimes, network and power outages, targeted attempts to disrupt the service (denial-of-service attacks), and hardware failures.

In summary, availability means a “requirement intended to assure that systems work promptly and service is not denied to authorized users” (Guttman & Roback, 1995, p. 7). Organizations can effectively manage their electronic assets using cloud computing which eliminates single points of failure and enables high availability at a cost comparable to running hardware and software ICT infrastructure on the premises.

2.3 Bring Your Own Device

Information confidentiality, integrity, and availability should provide authorized and authenticated parties access and ability to use data in a timely fashion, in a desired form with minimum time and resources regardless of where the data is located. Even when adhering to the CIA triad, though, the data still cannot be considered secure due to multiple layers, devices, networks, protocols, software, hardware, services, and people between the endpoints on the transfer path each of which can be compromised to passively or actively intercept communication while user has little to no indication of such actions taking place.

Desktop stations, servers, and notebooks have been the core security focus since becoming ubiquitous in organizations: advances in detection, response, and containment protocols were automated in antivirus suites; intrusion detection systems (IDS), firewalls, egress filters, sandboxes, and data loss prevention software thwarted some risks associated with electronic processing. This forced perpetrators to either increase attack sophistication, or exploit alternative vectors such as human element or newly-emerging technologies making their way into corporate environments. No suitable policies have usually been set for this class of devices as a result of low flexibility and reactive approach to new trends, creating a window of opportunity with no countermeasures put in place.

Rapid adoption of advanced technology is known as consumerization and its defining aspect is “. . . the concept of ‘dual-use’. Increasingly, hardware devices, network infrastructure and value-added services will be used in both businesses and consumers” (Moschella, Neal, Opperman, & Taylor, 2004, p. 4). Consumer-grade ICT evolves rapidly due to shorter innovation cycles which leads to their gradual acceptance from firms: “At first, companies stop prohibiting personal devices, then they allow connecting to corporate Internet servers, next they connect personal devices to corporate applications” (Copeland & Crespi, 2012, p. 1). Consumerization is coupled with a move to cloud and desktop virtualization where “. . . [t]he operating system of desktops is installed on virtual machines, which are located on a virtualization infrastructure in a data center, and the user remotely operates the desktops via thin clients” (Man & Kayashime, 2011, p. 1). “In a thin-client computing environment, end users move from full-featured computers to thin clients, lightweight machines primarily used for display and input and which require less maintenance and fewer upgrades” (Nieh, Yang, & Novik, 2000, p. 1). While they suffer from a single point of failure if the virtualization platform is hosted locally, users are allowed to access them from any device irrespective of geographic location.

The idea of ubiquitous computing (pervasive computing, ambient intelligence, everywhere) as unrestricted, pervasive electronic resource availability, networks of interconnected devices, and scalability was coined by Weiser (1991, p. 1) who came up with “[t]he idea of integrating computers seamlessly into the world at large. . .” A more recent delimitation sees it as utilizing “. . . countless very small, wirelessly intercommunicating microprocessors, which can be more or less invisibly embedded into objects” (Friedewald & Raabe, 2011, p. 1). But security concerns were pointed out: “Pervasive computing will see the accumulation of vast amounts of data that can provide a comprehensive overview of an individual. . . [T]hese huge sets of data and the spontaneous networking of smart objects will make it impossible for the pervasive computing user to trace where one’s personal data are stored, how they are used and how they might be combined with one another. . . data protection is therefore an essential requirement for protecting privacy – even more so than in other IT systems” (FOIS, 2006, p. 15). Schmidt (2010, p. 3) further mentions that “[p]ervasive computing technologies are transparent to users until the system malfunctions. . . it is difficult for the end user to identify where the problem lies.”

One device class in particular has brought ubiquitous computing to consumers: small form factor devices, specifically smartphones and tablets. Ballagas, Borchers, Rohs, and Sheridan

(2006, p. 1) note that “[t]he emerging capabilities of smart phones are fueling the rise in the use of mobile phones as input devices to the resources available in the environment. . . The ubiquity of mobile phones gives them great potential to be the default physical interface for ubiquitous computing applications.”

Growth of smartphone ecosystem led to the introduction of BYOD (Bring Your Own Device) and more recently, BYOT (Bring Your Own Technology), subsets of consumerization aimed at mobile hardware (Scarfò, 2012) and hardware with software, respectively. Apart from privacy issues, security challenges arose, too and “[w]ith the advent of cloud storage with its partitions, care should be taken in-house to ensure that data is partitioned and individual users only get access to the information they need to perform their assigned duties,” (Miller, Voas, & Hurlburt, 2012, p. 3) as detailed in chapter 2.2.1. Miller et al. (2012, p. 3) also admit that “. . . little attention has been paid to this issue, but that’s a problem that will need to be addressed if BYOD and BYOT become adopted widely. . .” Irreversible modifications are not an option which “. . . stems from the fact that given the device does not belong to the enterprise, the latter does not have any justification – and rightly so – in modifying the underlying kernel of the personal device of the employee” (Gessner, Girao, Karame, & Li, 2013, p. 1).

2.3.1 Background

Mobile phones, alternatively titled cellular, cell or feature phones, have undergone considerable and rapid transformation. From devices capable of performing basic operations (Short Message Service, calls, contact manager) they evolved into dedicated computing, multimedia as well as social platforms, incorporating functionality on par with desktop stations and laptops. However, the form factors make them highly portable and inconspicuous when in operation. Mobile phones have become commercialized with advent of high-speed digital cellular (2G), mobile broadband (3G), and native IP (4G) networking standards. Block diagram showing hardware modules each of which needs to be supported in the mobile operating system is depicted in Figure 14.

Growing portfolio of features incorporated into mobile phones gave rise to the term smartphone: a device with dedicated operating system whose complexity and breadth of functions outstrip feature phones. Location-aware and streaming services, wireless access, VoIP (Voice over IP) as well as video telephony changed users’ lifestyles, entertainment and advertising industry, and provided empowerment to engage in various activities on the go with users exhibiting diversity in usage patterns (Falaki et al., 2010). Moreover, “. . . the phone is emerging as a primary computing device for some users, rather than as a peripheral to the PC,” (Karlson, Meyers, Jacobs, Johns, & Kane, 2009, p. 8) especially for information workers and freelancers. Future high-speed data transfer standards provide users with convenient way to consume streamed media, data-intensive applications while having superior responsiveness. A thin-client has been also proposed “. . . which provides cloud computing environment specifically tailored for smartphone users. . . to create virtual smartphone images in the cloud and to remotely run their mobile applications in these images as they would locally” (E. Y. Chen & Itoh, 2010, p. 1). Sales figures seem to confirm gradual migration from feature phones to smartphones. Even academia recognized the inclination; indeed, “. . . publication on research in subject related to adoption of [s]martphone technology is increasing continuously specially in the last five years which indicates importance of studying and understanding adoption of [s]martphone technology among scholars in various fields” (Aldhaban, 2012, p. 2).

As the user base grows, so do security concerns. Personally-identifiable data, GPS (Global Positioning System) coordinates, credit card information, data transfers, and others may be correlated to reconstruct history of physical location, financial transactions, and wireless network

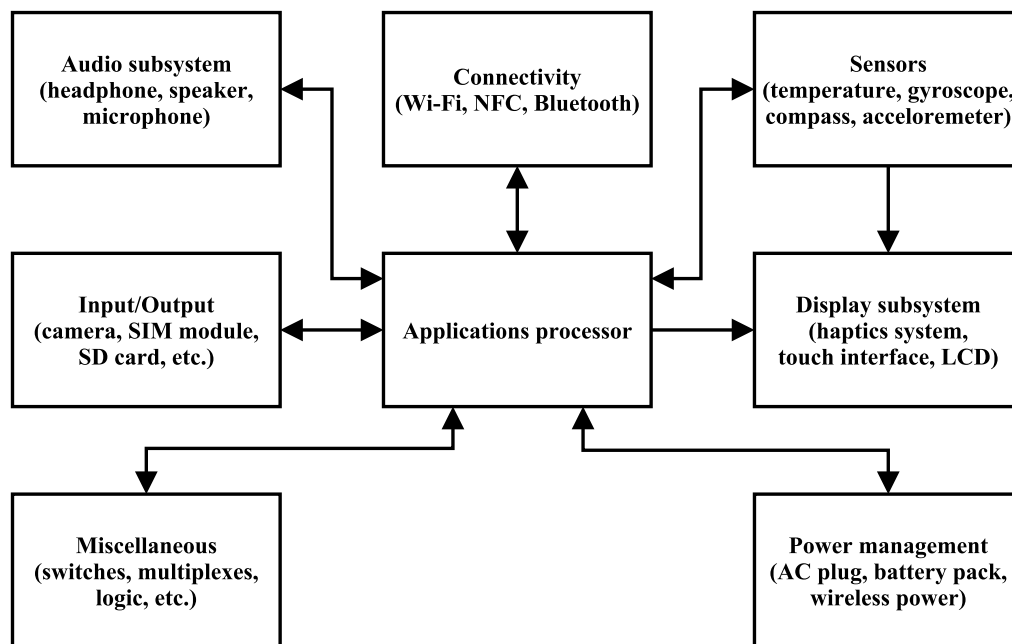


Fig. 14: *Smartphone block diagram. Each subsystem is designed and optimized for energy efficiency because battery capacity is a limitation of the small form factor employed.*
 Source: Texas Instruments (2013), modified.

trails for per-user electronic behavior profiling. At least a portion of users is aware of the implications: Chin, Felt, Sekar, and Wagner (2012, p. 7) found out study participants were “... less willing to perform tasks that involve money (banking, shopping) and sensitive data... on their phones and on their laptops... [and] are more concerned with privacy on their phone than they are on their laptop.” Dangers of incorporating advanced networking and processing capabilities into mobile phones had been discussed before smartphones became widely available (Guo, Wang, & Zhu, 2004). Dagon, Martin, and Starner (2004, p. 1) claimed that “... physical control of a computer doesn’t automatically guarantee secure control. Users tend to have a false sense of security with handheld or portable consumer electronics, leading them to trust these devices with more sensitive information.” It is unclear whether the sentiments will attenuate with continuing smartphone and tablet pervasiveness, or whether they will get more pronounced. BYOD increases the likelihood a security breach can propagate into the organization’s internal network as there is no clear separation between personal and work spaces if no suitable countermeasures and policies are implemented.

Malware (portmanteau of malicious software) makes it possible for the perpetrators to exfiltrate sensitive data without user consent and make further unsanctioned modifications to the device, sidestepping any input required from the victim. Developers, aware of such possibilities, incorporated safeguards and protective measures to mitigate or neutralize the most prominent attack vectors. Ranging from cryptographic instruments to hardware-imposed locks, they intend to keep the mobile ecosystem as secure as possible without incurring unnecessary user experience penalties.

2.3.2 Hardware

Smartphones incorporate elements similar to desktop stations: CPU, RAM, flash memory storage, I/O (Input/Output), LCD (Liquid-crystal Display)/LED-based (Light-emitting Diode) display technology, peripherals support, Bluetooth connectivity, and a WNIC (Wireless Network

Interface Controller) module. As more hardware is integrated onto the circuit boards, software has to be added to the mobile operating system (OS) which increases complexity and expands the attack surface. System complexity is defined as "... a property of a system that is directly proportional to the difficulty one has in comprehending the system at the level and detail necessary to make changes to the system without introducing instability or functional regressions" (Rosser, 2006). Software complexity in particular has generated a slew of metrics aiming to quantify the property: frequently mentioned (Weyuker, 1988) are number of program statements, McCabe's cyclomatic number, Halstead's programming effort, and knot measure. A highly-complex system creates more opportunities for exploitation; probability of breach could be reduced by omitting parts deemed inessential and in some cases, redesigning the system to comply with secure coding practices (Graff, 2001). This is not in line with practice described by Gjerde, Slotnick, and Sobel (2002, p. 1) for incremental innovators "... frequently introducing new models that are only slightly different from the previous ones and do not incorporate all possible technological advances," as compared to frontier innovators "... choosing not to introduce a new model until it is very different from the previous models and is at the leading edge of technology frontier." Smartphone developers combine both approaches and retain core resources across releases, fixating the attack surface.

Power efficiency is a primary factor in an environment restricted by battery capacity. Ascent of flash memory, a non-volatile erasable storage medium, brought about massive increases in speed and reliability with reductions in energy consumption. Data are written by applying electrical current and no mechanical system for storing and retrieving is necessary, a disadvantage of HDDs which "... are generally very reliable but they are also very complex components. This combination means that although they fail rarely, when they do fail, the possible causes of failure can be numerous" (Pinheiro, Weber, & Barroso, 2006, p. 1). Conversely, flash memory eliminated protracted seek times due to nearly uniform availability of each requested location. Its disadvantages are non-negligible wear and tear process deteriorating storage integrity over time, a concern mainly for enterprise-level solutions, not consumer electronics, the need for block erasure, read disturb, and write amplification (Hu, Eleftheriou, Haas, Iliadis, & Pletka, 2009). Smartphones use flash memory-based storage modules exclusively with massive economies of scale which decrease prices. Flash memory contributed to fast and efficient data storage and retrieval, and advances in CPU design and miniaturization assured adequate level of power-constrained computational resources. Smartphones can perform multi-threaded and multi-core operations including but not limited to gaming, scientific computations, media encoding, real-time high-resolution GUI (Graphical User Interface) rendering and refreshing, and high-definition content streaming. Haptic⁵ interfaces present users with tactile controls and direct device feedback via on-screen keyboard and gesture prompts.

Another function which differentiates smartphones from feature phones and at the same time poses imminent security risk is wireless connectivity, data transfers, and GPS location services. Constandache, Gaonkar, Saylor, Choudhury, and Cox (2010, p. 1) admit that "[w]hile GPS offers good location accuracy of around 10m, it incurs a serious energy cost. . .;" positioning using Wi-Fi and other sensors is more energy-efficient and may determine location with higher accuracy, especially in urban areas. Comparison of exploitable technologies integrated in smartphones is provided in Table 3.

To facilitate access to wireless Access Points (AP), smartphones are equipped with hardware modules supporting different Wi-Fi standards. The WNIC can reconnect to already-visited networks automatically: the OS "... scans for APs and then chooses the unencrypted one with the highest signal strength. . . [the method] which we call 'strongest signal strength', or 'SSS',

⁵[hæptɪk], *adjective*: relating to or involving sense of touch (Oxford University Press, 2011)

Tab. 3: *Wireless positioning systems. While GPS provides high accuracy, it is the least energy-efficient. Wi-Fi estimation accuracy and energy efficiency lie between GPS and cellular but it reliably works only in areas saturated with wireless networks. Signal from cellular base stations extend both indoors and outdoors but its location capabilities are limited to hundreds of meters. Bluetooth's low coverage is offset by high accuracy within the area.*

Source: Han, Qian, Leith, Mok, and Lam (2011, p. 2), modified.

	GPS	Wi-Fi	Cellular	Bluetooth
Lifetime [h]	10	40	60	60
Coverage [m]	Outdoor	50	Everywhere	10
Error [m]	10	40	400	10

ignores other factors that matter to then end user,” (Nicholson, Chawathe, Chen, Noble, & Wetherall, 2006, p. 2) a vulnerability which could be exploited to take over the communication channel. Also present is logic gauging signal strength by keeping track of nearby APs and transferring to the strongest one automatically. With wireless features on par with notebooks, it is necessary to ensure adequate protection of data bidirectionally transferred over unsecured networks, and data about the network itself saved on the device (usernames, passwords). Facing such challenges is no different in the mobile-based cyberspace than in the desktop-based.

2.3.3 Software

Mobile operating system (OS) is an extension of either free or proprietary kernel whose purpose “... is to implement these fundamental concepts: simulation of processes; communication among processes; creation, control, and removal of processes,” (Hansen, 1970, p. 1) supporting hardware and software of the particular platform. An OS is “[a] collection of software, firmware, and hardware elements that controls the execution of computer programs and provides such services as computer resource allocation, job control, input/output control, and file management in a computer system,” (IEEE, 1990, p. 145) or “... an intermediary between the user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a [convenient] and [efficient] manner” (Silberschatz, Galvin, & Gagne, 2012, p. 1). A high-level view of a generic OS is depicted in Figure 15.

As the kernel handles all interactions with the device hardware, it may appear to constitute OS's single point of failure. However, as the system itself issues commands to the kernel, any exploit allowing the attacker to execute arbitrary code must be treated as critical. Both parts are viable targets, although compromising OS is preferred due to wider attack surface (each application the user executes interacts with system resources) and technical expertise to bypass the OS layer and interact with the kernel directly. Mobile OS developers have implemented software safeguards to ensure kernel integrity, e.g., virtual memory, Kernel Patch Protection (KPP), (Microsoft, 2007) and ACL. For example, Ritchie (1979) admitted that “... UNIX was not developed with security, in any realistic sense, in mind; this fact alone guarantees a vast number of holes. (Actually the same statement can be made with respect to most systems),” which supports the insecurity argument raised in chapter 2.1.5. Android is a descendant of UNIX which draws on many of its components. An overview of the mobile OS market landscape as of 2014 is provided in Table 4.

Developers chose different ways to distribute their operating systems via OTA (Over-the-Air) programming. All allowed third parties access to API (Application Programming Interface),

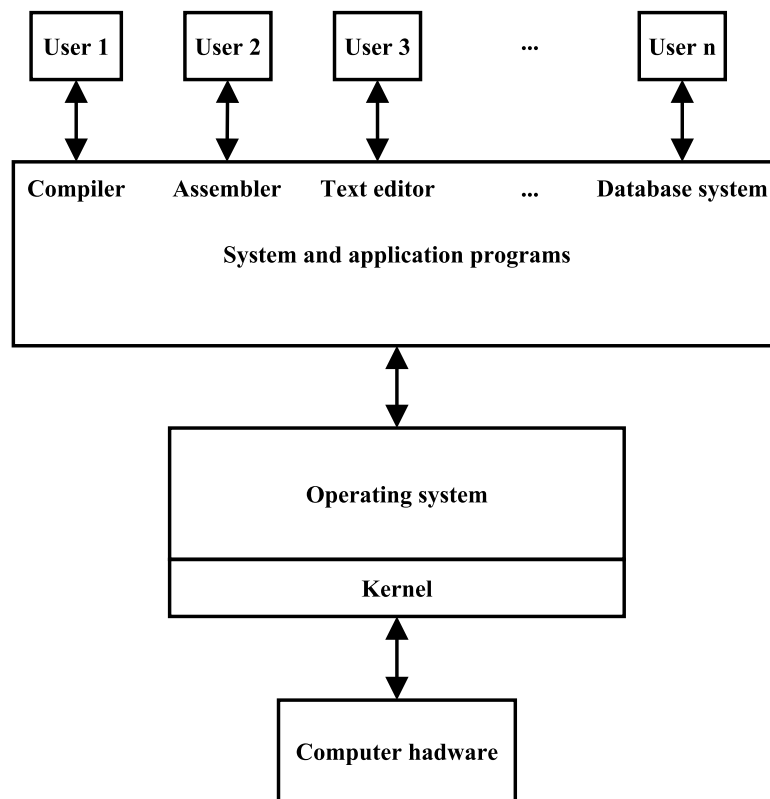


Fig. 15: Schematic view of an operating system. The kernel is a low-level core mediating requests from and to hardware (I/O), and distributing limited resources among active processes, Users do not interface with the kernel directly. The system further comprises GUI, libraries, utilities, and applications (user space) for convenient access and management.

Source: Silberschatz, Galvin, and Gagne (2012, p. 4), modified.

SDK (Software Development Kit), and documentation, opening the platforms to non-native code execution. Rapid hardware and software developments in the smartphone ecosystem and shifts in consumer preferences mean the information in Table 4 may become obsolete after some time as new products are introduced and others discontinued. Best practices and policies described in chapter 6 do not presuppose any particular OS and are applicable to most, if not all, current products. Security practices among vendors differ, though: while some focus on corporate security by providing native frameworks and procedures for device-level and system-level control, some aim at consumer sector with marginal BYOD support.

2.3.4 Summary

The chapter described hardware and software capabilities of smartphones to demonstrate how evolved the devices have become which turned them into a security vulnerability if not managed properly. The author is of the opinion BYOD should be addressed in organizational policies and enforced by profiles. Moreover, smartphones should be considered on par with desktop stations due to their capabilities and treated accordingly. BYOD management presents a challenge because the devices do not belong to the organization which limits the scope of measures and necessitates user consent.

Disregarding smartphone security when accessing sensitive data is a vulnerability the perpetrator can exploit to gain persistence on the internal network. Users should therefore be expected to make concessions if they demand integration of their device into the organizational ICT

Tab. 4: *Overview of mobile OS landscape. Market shares and discontinued products (i.e., Bada, Windows Mobile) are not included. Smartphone ecosystem has undergone major shifts and developments in consumer demand and preferences, current market data are therefore not indicative of future trends. Source: own work.*

Name	Developer	Model
Android	Google	free, open source
BlackBerry	BlackBerry	proprietary, closed source
iOS	Apple	proprietary, closed source
Nokia Asha	Nokia	proprietary, closed source
Windows Phone	Microsoft	proprietary, closed source
Windows RT	Microsoft	proprietary, closed source

infrastructure, and profiles are the least-intrusive means of ensuring best practices are being followed even in environments such as open Wi-Fi networks. Chapter 2.4.4 discusses attacks which can be mounted against unsecured wireless networks users regularly pair with for Internet connectivity. Profiles can mitigate the risks by creating encrypted tunnels through which data are passed to the organizational electronic resources (email servers, information systems, VoIP servers) while marginally impacting user comfort and convenience.

The questionnaire research presented in chapter 4 will map and analyze attitudes of respondents in a representative sample toward installing profiles on their mobile device. This will support the theoretical background with real-world observations which will then be used for devising best practices in chapter 6.

2.4 Techniques for Unauthorized System Access

This chapter provides a high-level overview of techniques adversaries can use to gain unauthorized system access. Specific instances, i.e., cases where victims were targeted will not be discussed nor presented as they become obsolete quickly. Some attacks are theoretical, “what if” scenarios, others are frequent in practice due to their effectiveness and pervasively vulnerable back-end platforms (operating and database management systems). The Internet protocol suite, largely unchanged for backward compatibility concerns, is extensively exploited, as are automated authentication tools. Wireless networks with default security settings and weak encryption give attackers opportunity to bypass perimeter defenses and access internal resources directly in case the network is an entry point for smartphone-enabled employees and no mechanism such as access control list (ACL) is utilized. Mobile device exploitation should be expected to increase as BYOD will accelerate in the future. The attacks can be accompanied by social engineering campaigns where employees themselves are designated as targets and scenarios created which trigger instinctive reactions anticipated and acted upon by the malicious third party. Finally, penetration testing will be introduced as a way to comprehensively audit ICT infrastructures, human element, security policies, and their resilience in real-world scenarios.

Asset confidentiality, integrity, and availability are essential for business continuity, and contingency plans must be developed for mission-critical ICT infrastructure and employees as part of an overall defense strategy. Perimeter defense, e.g., intrusion detection system, firewall, ACL, antivirus, event logging, air-gapping, and red/black separation should be deployed in combination. This is called the defense in depth principle and is outlined in chapter 6.2. Air-gapped systems are “. . . secured and kept separate from other local networks or the Internet and operate on specially-designed software to for their unique purposes. The implied conclusion from air-gapping is that these systems are safe from, and invincible to, computer network attacks” (Noor, 2011, p. 57). While effective against network threats, air-gapping has since been proven vulnerable to uncontrolled interconnects such as mass storage devices. Red/black separation “. . . views the world of communications in two categories of media: red and black. Each category refers to the type of information that can be transmitted over that media. A red communications system can handle and fully protect classified plaintext data. A black system can handle unclassified plaintext and classified ciphertext. . . . A red system is physically separated from black systems” (S. H. Bennett, 2001, p. 6).

The division is one way to thwart side-channel attacks, originally devised to target cryptographic hardware circuits. They rely on side channel information “. . . that can be retrieved from the encryption device that is neither the plaintext to be encrypted or the ciphertext resulting from the encryption process” (Bar-El, 2003, p. 2). Timing of individual operations, power consumption, vibrations, sound oscillations, light reflections, and electromagnetic emanations can be intercepted to partially or entirely reconstruct program’s control flow and duplicate data written, transferred, and displayed to the user. A general-purpose hardware side-channel attack was notably demonstrated as capable to reproduce Cathode Ray Tube (CRT) display image at a distance of 50 meters, “. . . [t]he set-up is simple and measurements do not take an unreasonably long time to be carried out” (van Eck, 1985, p. 8). Similar results later corroborated susceptibility of Liquid-crystal Display (LCD) panels to the same conduct at a 10-meter distance with no direct line of sight, concluding that “[t]he eavesdropping risk of flat-panel displays. . . is at least comparable to that of CRTs” (Kuhn, 2004, p. 17). The thesis will not further consider side-channel exploitation despite smartphones being at risk and “[t]he vulnerability. . . not specific to a particular mobile device, model, or cryptographic implementation. . .” (Kenworthy & Rohatgi, 2012, p. 4).

Security policies and safeguards by themselves do not guarantee impenetrability as the attack surface continually shifts, new vulnerabilities are discovered, documented, and patched, and novel threat vectors emerge and are exploited regularly. Complexity increases time to exhaustively map and resolve collisions, prolonging patch deployment which keeps the vector open and creates a window of opportunity, schematically depicted in Figure 16. Depending on the disclosure method (full, partial, non-disclosure), priority the vulnerability is assigned, support status and other factors, the prerelease phase may span months; vendors vary in speed with which their products get patched. On the other hand, the postrelease phase denotes the time until the patch is deployed according to the organizational ICT policies. Older systems in particular are often left untouched for stability reasons or due to lack of official support. In this case, the window of opportunity is not closed until the system is replaced.

Enterprises should strive to reduce their software footprint and streamline patch management. This necessitates Chief Information Security Officer (CISO) who monitors, assesses, mitigates, anticipates, and responds to risks associated with application, computer, data, infrastructure, network, physical as well as mobile security, and balances them with user comfort and productivity. Precautions which increase security may lower work effectiveness, create processing bottlenecks, and introduce delays and computational overheads. In production environment, patch management aiming "...to create a consistently configured environment that is secure against known vulnerabilities and in operating system and application software," (Chan, 2004) is strongly recommended. Both server- (backup software, database managers, web applications) and client-side (desktops, mobile devices, notebooks) benefit from patch management after prior testing to determine compatibility with critical ICT infrastructure elements. A patch is "...a piece of software code that is inserted into a program to temporarily fix a defect. Patches are developed and released by software vendors when vulnerabilities are discovered" (Dacey, 2003, p. 3). If n dedicated parts exist and one is updated, $n - 1$ unique interdependencies need to be evaluated. Moreover, "[k]eeping the system up-to-date with recently released patches results in higher operational costs, while patching the system infrequently for its vulnerabilities leads to higher damage costs associated with higher levels of exploitation. . . Therefore, the firm should define its patch-update policy to find the right balance between operational and damage costs considering vendor's patch-release policy" (Cavusoglu, Cavusoglu, & Zhang, 2008, pp. 11–12).

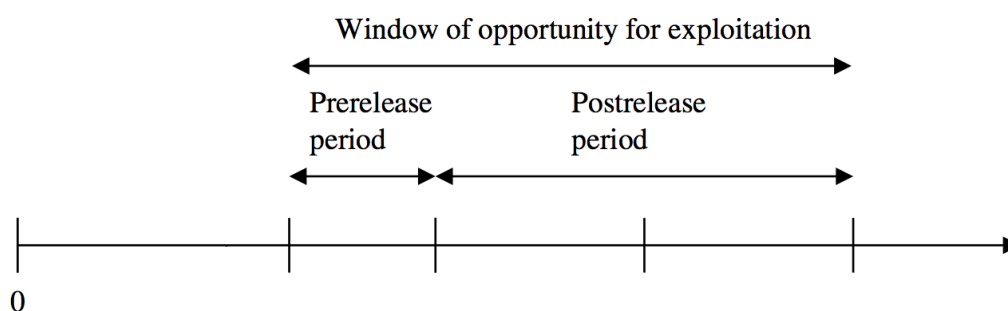


Fig. 16: *Window of opportunity. It is initiated when either a vendor discloses a vulnerability and releases a patch, or the attacker exploits the vulnerability. The window is closed when countermeasures are deployed in production environment and the threat is neutralized.*

Source: Cavusoglu, Cavusoglu, and Zhang (2008, p. 4), modified.

2.4.1 Modeling the Adversary

Before the techniques are discussed, it is necessary to model attacker's behavior and motivations which allows to analyze and harden avenues most likely taken to achieve their objective. When

security is perceived as approximately equal for all exploitable vectors, it is reasonable to assume none will be given priority and each is equally likely to be selected for the attack.

- **Anonymity.** The attacker must always be assumed to take steps to conceal their identity either through misappropriation (victim as a proxy) or misdirection (virtual private network services, Tor nodes). Host identification and pinning have made some progress and “[w]hile the Internet lacks strong accountability, this lack does not mean that users and hosts are truly anonymous. . . . The host-IP binding information can be used to effectively identify and block malicious activities by host rather than by IP address” (Xie, Yu, & Abadi, 2009, p. 11). False positive and false negative rates should be taken into consideration when assigning responsibility and initiating counteractions.
- **Bounded rationality.** The attacker is not expected to have perfect information or knowledge (H. A. Simon, 1957) of ICT infrastructure and processes, though they are able to reconnaissance and extrapolate some, e.g., only limited number of products provide firewall services, and the target can thus be fingerprinted or exploits prepared for all based on shared vulnerabilities. Bounded rationality seemingly limits the attacker but high proclivity toward particular vendors make the constraint less disadvantageous. Metadata extraction (chapter 2.2.2) can also provide actionable intelligence.
- **High computational capabilities.** The attacker either owns or rents hardware to execute repetitive operations in parallel, making resource-intensive ventures such as reverse password engineering feasible under realistic assumptions. Furthermore, they are able to amass resources, e.g., network traffic, IP addresses, and direct them to perform desired actions simultaneously. With legitimate users usually attempting to access the same service being targeted, bulk network filtering policies generate high false positive and false negative rates. Per-request analysis is viable, although computationally expensive with performance directly proportional to the number of requests. Alternatives such as neural networks exist but Buhari, Habaebi, and Ali (2005, p. 11) concluded that “. . . usage of neural networks for packet filtering is questionable because the inclusion of extra security features like local or hourly hits to take care of the security lapse in neural network system causes the performance gain to be affected.”
- **Malicious intent.** The attacker aims to cause damage or inconvenience to the victim tangibly or intangibly, e.g., corrective maintenance after internal network breach (chapter 2.2.3), initiation of backup procedures, security audit costs, implementation of best practices, reputation damage management. The last one presents a challenge due to the need to modify customer’s perceptions and regain their trust (Rhee & Valdez, 2009). Reputation can be also damaged indirectly “. . . because stakeholders cannot distinguish the relative performance or effect of each user, all users share a common stakeholder assessment of their character. Consequently, the action of one firm affects the reputation of another” (A. A. King, Lenox, & Barnett, 2002, p. 4). This is akin to the tragedy of the commons in microeconomics (Hardin, 1968).
- **Predictability.** Attacker’s objectives can be narrowed down to coincide with categorization of electronic assets: sensitive information (credentials, financial and personally-identifiable data) is highly interesting and should be protected; unclassified information available to general public need not be monitored. Networks and devices for accessing internal resources should also be a priority with air-gapped, isolated systems posing significantly lower risk of compromise. Miscalculating adversary’s preferences may leave seemingly benign vectors open, especially those related to social engineering. Removing unnecessary third-party software and disabling unused services is another security practice to reduce the attack surface, understood as follows: “A system’s attack surface is the subset of its resources that an attacker can use to attack the system. An attacker can use a system’s entry points and

exit points, channels, and untrusted data items to send (receive) data into (from) the system” (Manadhata & Wing, 2010, p. 5).

The list is not exhaustive as some attacks require relaxing or adding conditions, but it provides a baseline on abilities and motivations the perpetrators exhibit. Some incursions may be highly targeted and damage deprioritized in favor of long-term information gathering, enumerating vulnerabilities, observing patch deployment cycles, estimating windows of opportunity, and focusing on predetermined objectives: data exfiltration (industrial espionage) and establishing covert presence in the system. Social engineering techniques increase attack potency, for example by tailoring emails which contain psychological stimuli, “. . . the ways in which individuals intentionally or purposefully (although not necessarily consciously) alter, change, influence, or exploit others. . .” (Buss, 1987, pp. 4–5). Detecting an incursion in progress and alerting appropriate parties could thwart future attempts and increase security by closing the pertinent attack vector, updating existing policies, and employee training curricula.

2.4.2 Human Interaction Proofs

Users interacting with various online services are frequently subjected to checks designed so that humans can trivially pass them. Human Interaction Proof (HIP) requires some form of input from peripheral devices based on animation, graphics, text, and sound depending on the challenge presented. HIPs are commonplace when creating free email accounts which can be exploited for spamming campaigns; Figure 17 depicts a text-based challenge on Google successful response to which proves with very high probability that the entity requesting the service is human, not a machine. Malicious third parties have been attempting to devise automated means for bypassing HIPs while alternatives allegedly increasing their efficiency have been proposed as well.

Authentication was first discussed in chapter 2.2.1. It comprises tools to directly prove identity between two parties in real-world settings, e.g., identification documents at the airport collated with information printed on the ticket, or remotely without direct contact, e.g., login credentials compared with an entry in a database of valid accounts. While non-negligible probability of subverting authentication mechanisms exist in case of direct authentication (counterfeiting), the thesis will focus exclusively on the remote version as it is much more frequently abused. Apart from passwords which represent specific form of authentication, i.e., uniquely identifying the other side, general authentication techniques will be discussed the most well-known of which is Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA), aiming to discern legitimate entities eligible to access the service, and those who attempt to circumvent the mechanism. Passwords will be detailed in the following chapter.

CAPTCHA is a text HIP first mentioned in chapter 2.1.1. It was devised by Naor (1996, p. 2) as follows: “One of the key ideas in [c]ryptography is applying the fact that there are intractable problems, i.e., problems that cannot be solved effectively by any feasible machine, in order to construct secure protocols. . . . What should replace the keyed cryptographic function in the current setting are those tasks where humans excel in performing, but machines have a hard-time competing with the performance of a three years old child.” Obstructed or malformed characters, distorted sounds, animations, shapes, puzzles, answers to simple questions, mouse interactions, and counting are examples of operations trivially solvable by a human but challenging or infeasible for representation in computers. Naor (1996, p. 3) further lists properties such a scheme should possess:

- “[i]t is easy to generate many instances of the problem, together with their unambiguous solutions,”

- “[h]umans can solve a given instance effortlessly with very few errors,”
- “[t]he best known programs for solving such problems fail on a non-negligible fraction of the problems, even if the method of generating the instances is known,”
- “[a]n instance specification is succinct both in the amount of information needed to describe it and in the area it takes to present it to the user.”

Work by von Ahn, Blum, Hopper, and Langford (2003, p. 2) introduced practical implementation of such scheme and conceded that “. . . from a mechanistic point of view, there is no way to *prove* that a program cannot pass a test which a human can pass, since there is a program – the human brain – which passes the test. All we can do is to present evidence that it’s hard to write a program that can pass the test.” Lack of rigorous proof together with popularity CAPTCHAs gained for their simplicity and streamlined integration prompted research in and convergence of Artificial Intelligence and security. Mori and Malik (2003, p. 2) achieved a breakthrough when their tool correctly identified distorted words represented in production-environment CAPTCHAs with 92 % success rate even though they “. . . present challenging clutter since they are designed to be difficult for computer programs to handle. Recognition of words lend itself easily to being approached either as recognition by parts (individual letters or bigrams) or whole objects (entire words).” Chellapilla and Simard (2004, p. 3) attempted the following: “Our generic method for breaking. . . HIPs is to write a custom algorithm to locate the characters, and then use machine learning for recognition. Surprisingly, segmentation, or finding the characters, is simple for many HIPs which makes the process of breaking the HIP particularly easy.” Of interest are warped characters depicted in Figure 17 which does not contain any geometric shapes to thwart pattern recognition but instead presents user with two words whose shape should deter automated attacks.



Fig. 17: CAPTCHA. The image was presented as a HIP when creating a new email account. It contains two strings, “consider” and “erchop.” Characters in the second one are warped (elongated and purposefully not following a straight line) to break attempts at automated analysis which would allow to create a valid account.

Source: Google.

Surrogate HIPs have been also scrutinized. Audio CAPTCHAs, based on recording the characters the user is prompted to type while background noise is added to decrease automated speech recognition efficiency, serve as an alternative for the visually impaired. They were found vulnerable: the method “. . . extracts an audio segment from a CAPTCHA, inputs the segment to one of our digit or letter recognizers, and outputs the label for that segment. . . until the maximum solution size is reached or there are no unlabeled segments left” (Chellapilla & Simard, 2008, p. 4). Success rates ranged from 45 % to 71 %, well above the 5 % threshold frequently selected in statistical hypotheses testing as a probability attributable to chance. Later research showed 75 % success rate (Bursztein & Bethard, 2009, p. 5) and suggested “. . . to limit both the number of [CAPTCHA] downloads allowed for each IP address (download limit) and the number of times an IP address is allowed to submit an incorrect response (error limit).” Image-based CAPTCHAs have also been devised (Chew & Tygar, 2004; Datta, Li, & Wang, 2005) but the research results “. . . are more than sufficient to consider the [CAPTCHA] broken, or at least not safe” (Merler & Jacob, 2009, p. 9). Another proposed method “. . . clearly identifies the shortcomings of several currently existing image [recognition] CAPTCHAs” (Fritsch, Netter,

Reisser, & Pernul, 2010, p. 12). Regardless of technique, CAPTCHA systems are prone to faulty implementations allowing to bypass the authentication mechanism altogether (Yeend, 2005).

2.4.3 Passwords

Securely hashing user login credentials to make online and offline exhaustive reverse engineering time-dependent is often underestimated. The practice was discussed in chapter 2.2.2: instead of storing sensitive data in human-readable plaintext form, mathematical fingerprints uniquely identifying each string should be strongly preferred for comparison purposes. Cryptographic hash functions must have several efficiency properties such as being extremely fast to compute, and the length of the digest a fixed constant typically much smaller than the length of the message (Zimand, 2013, p. 2). Another property is high computational requirements to reverse the function.

Initially, hash functions were considered safe due to the number of mathematical operations necessary to produce a string hashing to the same output as the original. Advances in processing power as per Moore's law (chapter 2.2.3) and targeted scenarios against popular implementations (MD5, SHA-1) led to recommendations these hash functions be obsoleted in favor of more resilient ones. Many database management systems continue to offer insecure schemes as defaults, though. The author believes this is detrimental to security because cloud computing and dedicated hardware components can enumerate billions of keys in parallel with optimization techniques further capable to reduce the candidate search space.

Two ways to break hashes have been devised apart from side-channel vulnerabilities presented earlier in chapter 2.4. One is brute-force attack (exhaustive key search) that is "...based on a simple concept:... the attacker... has the ciphertext from eavesdropping on the channel and happens to have a short piece of plaintext, e.g., the header of a file that was encrypted. [He] now simply decrypts the first piece of ciphertext with *all possible* keys... If the resulting plaintext matches the short piece of plaintext, he knows that he has found the correct key... Whether it is feasible in practice depends on the key space, i.e., on the number of possible keys that exist for a given cipher. If testing all the keys on many modern computers takes too much time, i.e., several decades, the cipher is *computationally secure* against a brute-force attack" (Paar & Pelzl, 2010, p. 7). In a brute-force attack, a hash is obtained by breaching a database with user credentials via SQL injection (chapter 2.4.7) or some alternative exploit, determining the search space, systematically enumerating all candidates from the pool, and verifying the hashes against the target value. If not identical, a true negative was encountered, or a false negative with the candidate string selected correctly but the hash requiring additional data added server-side to increase security (salt), which were omitted. If the adversary has no prior knowledge about the hash composition, one case is unrecognizable from the other as the comparison procedure simply outputs a yes/no statement. Key length determines the worst-case (upper bound) time it takes to break a given hash: in general, for an n -bit key the maximum number of operations is 2^n and $\frac{2^n}{2}$ on average.

Chaney (2012) points out that "[t]he resources required for a [brute-force] attack scale exponentially with increasing key size, not linearly. As a result, doubling the key size for an algorithm does not imply double the required number of operations, but rather squares them." Extending the digest from 128b to 129b thus increases the time factor considerably. This makes cryptographic hash functions supporting longer products desirable security-wise as long as they conform to the requirements stated above (e.g., extremely fast to compute). Table 5 lists examples of functions for algorithmic efficiency analysis, denoting the number of operations necessary to successfully terminate its run. Complexity of some is directly proportional to their

Tab. 5: *Algorithmic growth functions. Brute-force attacks belong to a category of exponential algorithms whose order of growth outpace their input size even for small n. Cryptographic hash functions were purposefully designed for reverse engineering to be inefficient and computationally expensive to deter malicious attempts.*

Source: Levitin (2011, p. 46).

n	$\log_2 n$	n	$n \log_2 n$	n^2	n^3	2^n	$n!$
10	3.3	10^1	$3.3 \cdot 10^1$	10^2	10^3	10^3	$3.6 \cdot 10^6$
10^2	6.6	10^2	$6.6 \cdot 10^2$	10^4	10^6	$1.3 \cdot 10^{30}$	$9.3 \cdot 10^{157}$
10^3	10	10^3	$1.0 \cdot 10^4$	10^6	10^9		
10^4	13	10^4	$1.3 \cdot 10^5$	10^8	10^{12}		
10^5	17	10^5	$1.7 \cdot 10^6$	10^{10}	10^{15}		
10^6	20	10^6	$2.0 \cdot 10^7$	10^{12}	10^{18}		

input size while output of others outpace the growing input very early on. As Levitin (2011, p. 46) comments on exponential and factorial functions, “both... grow so fast that their values become astronomically large even for rather small values of n ... For example, it would take about $4 \cdot 10^{10}$ years for a computer making a trillion (10^{12}) operations per second to execute 2^{100} operations... Algorithms that require an exponential number of operations are practical for solving only problems of very small sizes.”

The MD5 Message-Digest algorithm is a popular cryptographic hash function routinely deployed to convert user credentials to digests. Designed by Rivest (1992), its summary states that “[t]he algorithm takes as input a message of arbitrary length and produces as output a 128-bit ‘fingerprint’ or ‘message digest’ of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest.” The assumption has since been proven incorrect in research by M.M.J. Stevens (2007), Marc Stevens, Lenstra, and de Veger (2012) who were able to forge a false digital certificate allowing them to impersonate arbitrary legitimate website on the Internet using parallel computations on commercially-available hardware. The finding prompted a proclamation that “. . . there is no proper excuse for continued use of a broken cryptographic primitive when sufficiently strong alternatives are readily available. . . [A] standard user will likely not notice anything. Therefore inspection of certificates is not a strong countermeasure” (Sotirov et al., 2008). C. R. Dougherty (2009) concluded that “[a]s previous research has demonstrated, [MD5] should be considered cryptographically broken and unsuitable for further use.” Chapter 6 lists suitable alternatives for MD5, namely PBKDF2, bcrypt, and scrypt all of which deliberately incur performance penalties by either iterating the function multiple times with added (pseudo)random data (salt), or necessitating large amount of memory.

Naïve brute-force attack assumes each string (password) is equally likely to occur, i.e., uniform password selection distribution function. Due to passwords’ ubiquity and “[a]lthough the user selects a password by combining characters or numbers than can be selected from the keyboard, [passwords consisting] of consecutive numbers, specific words or sentences are frequently used for the most part” (Kim, 2012, p. 1). The practice is not exclusive to mobile devices where convenience and typing speed is preferred over complexity, but in situations where full instead of virtual keyboard is available, such as desktop stations and notebooks. Empirical findings on patterns in passwords (D. V. Klein, 1990; Zviran & Haga, 1999; Yampolskiy, 2006) have corroborated the hypothesis “. . . that people’s choice of passwords is non-uniform, leading to some passwords appearing with a high frequency. . . [O]ne consequence of this: a relatively small

Tab. 6: Password mutation list. All rules are demonstrated on the string “password” which is modified accordingly, its message digest computed and compared to the target string value.
 Source: Korolkova (2009), modified.

Method	Examples
Case mutations	Password, pAssword, PAssword
Order mutations	drowssap, passwordpassword, passworddrowssap
Vowel mutations	psswrđ, pAsswOrd, paSSWord
Strip mutations	assword, pssword, pasword, passord
Swap mutations	psasword, paswsord, apswsord
Duplicate mutations	ppassword, paassword, passssword

number of words can have a high probability of matching a user’s password. To combat this, sites often ban dictionary words or common passwords. . . in an effort to drive users away from more common passwords” (Malone & Maher, 2012, p. 8). Expending computational resources on unlikely strings led to a dictionary attack, a faster version of brute-force enumeration sacrificing complete search space coverage for faster running times.

As password choices are strongly non-uniform and word lists can be used to compile likely candidates, brute-force enumeration covering the whole search space has been refined and optimized into a dictionary attack, assigning sequences such as “AZ@p)i#A” lower probability than “password” and other easily-memorable strings. Defined as “[a]n attack that uses a brute-force technique of successively trying all the words in some large, exhaustive list,” (Shirey, 2007) it is a subset of brute-force algorithms from which it inherits the successive iteration phase. To increase search space without resorting to large-scale enumeration, a sophisticated set of rules is employed to mutate the strings in the dictionaries; Table 6 provides an overview.

Additional rules (digit, year, border, delimiter, freak, and abbreviation mutations) are employed with numbers and special characters, further expanding the search space and include many best practices on how to create strong passwords. The mutators utilize dictionaries, text files of strings freely available on the Internet which make barriers of entry minimal, together with software automating the reverse engineering procedures. Coupled with extensive research and findings, e.g., that “a Zipf distribution is a relatively good match for the frequencies with which users choose passwords. . . . [P]asswords from one list provide good candidates when guessing or cracking passwords from another list,” (Malone & Maher, 2011, p. 13) little technical expertise is necessary to generate potent attack scenarios.

Both brute-force and dictionary attacks can be executed online or offline: it is trivial to prevent multiple requests in quick succession online by delaying the response after several consecutive failed attempts, or locking the account. A test similar to CAPTCHA (automated generation, easy for humans, hard for machines, small probability of guessing the answer correctly) was devised to separate legitimate users and machines (Pinkas & Sander, 2002). This substantially reduces effectiveness of online brute-force attacks but because the countermeasures work with fixed thresholds, “. . . when the system rejects the password as being incorrect for that particular user, the adversary picks a different password from the dictionary and repeats the process” (Chakrabarti & Singhal, 2007, p. 2). When targeting a specific account owner, their access can be blocked purposefully to initiate lockdown, disrupting the service for a specified amount of time after which the process is repeated, turning the attack into a primitive denial-of-service described in the following chapter. In the offline scenario, the password digest can be tested

indiscriminately as no safeguards are in place; the hash can be even transferred over to custom hardware circuits or cloud and the workload distributed among several virtual machines.

Compared to brute force, dictionary attack is not guaranteed to succeed in retrieving the sensitive string from the message digest due to reduced search space. Probabilistic Context-Free Grammar "...incorporates available information about the probability distribution of user passwords. This information is used to generate password patterns... in order of decreasing probability. [They] can be either password guesses themselves or, effectively, word-mangling templates than can be later filled in using dictionary words" (Weir, Aggarwal, de Medeiros, & Glodek, 2009, p. 1). Per-word or per-character probability templates are thus generated which are later populated with characters.

Dictionary attacks are thwarted by choosing suitable cryptographic hash functions, salting the data before generating digests, selecting passwords randomly, storing them in dedicated containers which ensure they remain encrypted when unused, and enforcing one-password-per-site policy to limit damage in case of compromise. As will be demonstrated practically in case study 1 (chapter 5.1), the vector is effective when uncovering longer strings in real-world situations where brute-force search would be prohibitively long.

2.4.4 Communication and Encryption Protocols

All devices communicating on the Internet have to conform to a set of protocols. Moreover, specifications of the core suite, Transmission Control Protocol/Internet Protocol (TCP/IP) are freely available for any party to inspect. Even though this resulted in many vulnerabilities discovered and addressed theoretically, TCP/IP cannot break backward compatibility because vendor support for a vast array of endpoints has been discontinued, and patching the ICT infrastructure therefore presents a significant challenge. In many cases, the only solution is to replace the outdated hardware and software which the operators are unwilling or unable to do, necessitating protocol compatibility across heterogeneous hardware and software. Moreover, adversaries found numerous ways to invoke unintended behavior by purposefully deviating from expected or implied routines.

Attacks are thus frequently leveled against widely-deployed communication and encryption protocols not assumed to change due to compatibility concerns, TCP/IP in particular because it forms a backbone for majority of Internet communication. A protocol is "[a] set of conventions that govern the interactions of processes, devices, and other components within the system" (IEEE, 1990, p. 161). As all devices need to comply with them to receive and transmit data, the attack surface is extensive which coupled with publicly-available documentation enable detailed understanding of their inner workings. This led to high-impact vulnerabilities sidelining specific implementations and instead focusing on the underlying structures. The most prominent type of attack is denial-of-service.

McDowell (2009) asserts that "[i]n a denial-of-service (DoS) attack, an attacker attempts to prevent legitimate users from accessing information or services... The most common and obvious type of DoS attack occurs when an attacker 'floods' a network with information." Carl, Kesidis, Brooks, and Rai (2006, p. 1) add that "[t]he Internet was designed for the minimal processing and best-effort forwarding of any packet, malicious or not... DoS attacks, which come in many forms, are explicit attempts to block legitimate users' system access by reducing system availability... The malicious workload in network-based DoS attacks consume network buffers, CPU processing cycles, and link bandwidth. When any of these resources form a bottleneck, system performance degrades or stops, impending legitimate system use." Consequences range

from inconvenience to financial, material, and other losses, e.g., when online banking systems and critical infrastructure are targeted and made unavailable through coordinated action of colluding devices, or from a single host. Multiple hosts can saturate victim's bandwidth by sending requests in volumes the server is unable to process simultaneously, depriving it of resources to distribute across incoming connections. A single perpetrator can exploit vulnerabilities to generate scenarios outside the protocol bounds to cause instability, unsanctioned code execution, or a crash.

Abliz (2011, p. 2) admits that “[p]reventing denial of service attacks can be very challenging, as they can take place even in the absence of software vulnerabilities in a system. Meanwhile, it is extremely hard, if not impossible, to precisely differentiate all attacker's requests from other benign requests. Thus, solutions that rely on detecting and filtering attacker's requests have limited effectiveness.” This is especially true for high-latency connections where delays are caused by the time it takes the packet to reach a destination rather than malicious intent. Several mechanisms have been proposed (Abliz, 2011; D. J. Bernstein, 2002; Mirkovic, Dietrich, Dittrich, & Relher, 2005) but none is universally preferred or widely deployed.

Distributed denial-of-service (DDoS) attacks also exist which “. . . use multiple systems to attack one or more victim systems with the intent of denying service to legitimate users of the victim systems. The degree of automation in attack tools enables a single attacker to install their tools and control tens of thousands of compromised systems to use in attacks. Intruders often search address blocks known to contain high concentrations of vulnerable systems,” (Householder, Houle, & Dougherty, 2002, p. 2) such as unsecured internal corporate networks. The adversaries “. . . simply exploit the huge resource asymmetry between the Internet and the victim in that a sufficient number of compromised hosts is amassed to send useless packets toward a victim around the same time. The magnitude of the combined traffic is significant enough to jam, or even crash, the victim (system resource exhaustion), or its Internet connection (bandwidth exhaustion), or both, therefore effectively taking the victim off the Internet” (Chang, 2002, p. 1). DDoS filtering techniques are prone to false positives and false negatives: a false positive occurs when a legitimate request is denied service due to it being classified as malicious, a false negative when attacker's connection is allowed through and determined benign. Both situations are harmful: economically, false positives increase opportunity costs, relinquishing profit the denied user would have generated (e.g., in electronic shopping) for security, false negatives increase the risk of unauthorized system access. High-profile servers have redundant capacities absorbing elevations in network activity which makes DoS challenging despite amplification, enabling a single host to multiply network traffic as much as 50 times (D. J. Bernstein, 2010). Migrating electronic operations to the cloud during DDoS and thus preserving quality of service was recommended (Latanicki, Massonet, Naqvi, Rochwerger, & Villari, 2010). The solution offers superior resilience to maintain all elements of the CIA triad discussed in chapter 2.2, although infecting virtual instances and launching DDoS attacks is a threat introduced as a side effect of the cloud. Monitoring and analyzing anomalies, e.g., high request count to a single IP address should thwart basic DoS when no additional protections are used. Malicious techniques directed at virtual machines will be the focus of chapter 2.4.6.

Another set of attacks exploits cryptographic protocols in devices expected to operate unmaintained over extended time periods, namely routers and wireless access points. Three encryption algorithms are routinely deployed on wireless networks to ensure protection of data: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and Wi-Fi protected Access II (WPA2), depending on the network operator and hardware. From a security standpoint, WEP, “[t]he original security standard used in wireless networks to encrypt the wireless network traffic,”

(Wi-Fi Alliance, 2013a) was shown to contain serious vulnerabilities (Fluhrer, Martin, & Shamir, 2001; Stubblefield, Ioannidis, & Rubin, 2004) which can reveal the secret encryption key and intercept any packets sent from and received by the victim in at worst two hours using consumer-grade hardware. An optimized version can recover the key in 60 seconds. Authors state that with “[t]he number of packets needed. . . [is] so low that opportunistic attacks on this security protocol will be most probable. Although it has been known to be insecure and has been broken by a key-recovery attack for almost [six] years, WEP is still seeing widespread use. . . While arguably still providing a weak deterrent against casual attackers in the past, the attack. . . greatly improves the ease with which the security measure can be broken. . .” (Tews, Weinmann, & Pyskin, 2007, p. 15). This makes WEP obsolete and insecure for sensitive corporate data access and interaction. A new scheme was devised, titled WPA, “[a]n improved security standard for wireless networks that provides strong data protection and network access control. . . [It] addresses all known WEP vulnerabilities” (Wi-Fi Alliance, 2013a). However, (Tews & Beck, 2007, p. 11) assert that “. . . even WPA with a strong password is not 100% secure and can be attacked in a real world scenario. Although this attack is not a complete key recovery attack, we suggest that vendors should implement countermeasures against the attack.” An extension was made available in 2004 in the form of WPA2, “[t]he follow on security method to WPA for wireless networks that provides stronger data protection and network access control. It provides enterprise and consumer Wi-Fi users with a high level of assurance that only authorized users can access their wireless networks” (Wi-Fi Alliance, 2013a). Since 2006, all devices certified by the Wi-Fi Alliance have to support WPA2 encryption in order to be granted conformation of compliance.

Even strong encryption protocols become vulnerable when insecure implementations open novel attack vectors. Wi-Fi Protected Setup (WPS) was devised for “. . . typical users who possess little understanding of traditional Wi-Fi configuration and security settings to automatically configure new wireless networks, add new devices and enable security” (Wi-Fi Alliance, 2013b). Walker-Morgan (2011) adds that “[it] simplifies the process of connecting a device to the Wi-Fi network by pushing a button to start the authentication, entering a PIN number from the new client into the access point, or entering an eight digit PIN number (usually printed on the device) from the access point to configure the connection.” A flaw was found in the PIN verification mechanism which “. . . dramatically decreases the maximum possible authentication attempts needed from $10^8 (= 100.000.000)$ to $10^4 + 10^4 (= 20.000)$. . . [T]here are at most $10^4 + 10^3 (= 11.000)$ attempts needed to find the correct PIN,” (Viehböck, 2011, p. 6) making brute-force attack with 100% success rate trivial in less than four hours with half the time needed on average. WPS is turned on by default in many devices supporting the technology, some with no apparent way to switch it off. Still, users are advised to disable WPS as “[a]n attacker within range of the wireless access point may be able to brute force the WPS PIN and retrieve the password for the wireless network, change the configuration of the access point, or cause a denial of service” (Allar, 2011). Neither WPA nor WPA2 defend against WPS exploitation.

Wireless networks rely on radio communication and electromagnetic radiation; any party within the signal range can intercept any and all data the access point broadcasts. Passive reception and analysis on unsecured channels, i.e., those without any encryption protocol set, presupposes compatible hardware (WNIC) and presence within the area covered by the radio-frequency signal. Due to no data being modified and sent from the WNIC, the sole means to detect an eavesdropper is to physically locate them, a challenge if the wireless network spans wide radius, e.g., university or corporate campuses. The monitoring device can be inconspicuous: smartphones can perform network traffic analysis identical to notebooks, ensuring almost complete anonymity. Users should be urged to request electronic resources using HTTPS protocol with Secure Sockets Layer/Transport Layer Security (SSL/TLS) encryption when on

unsecured networks, and never perform sensitive operations on HTTP connections. Implemented properly, HTTPS hampers real-time data analysis. But as Sanders (2010) admits, “[o]ne of the most prevalent network attacks used against individuals and large organizations alike are man-in-the-middle (MITM) attacks. Considered an active eavesdropping attack, MITM works by establishing connections to victim machines and relaying messages between them. In cases like these, one victim believes it is communicating directly with another victim, when in reality the communication flows through the host performing the attack. The end result is that the attacking host can not only intercept sensitive data, but can also inject and manipulate a data stream to gain further control of its victims.” When full control of the packet flow between the two parties is obtained, HTTPS can be stripped by tools readily available on the Internet. Schematic representation of MITM is depicted in Figure 18.

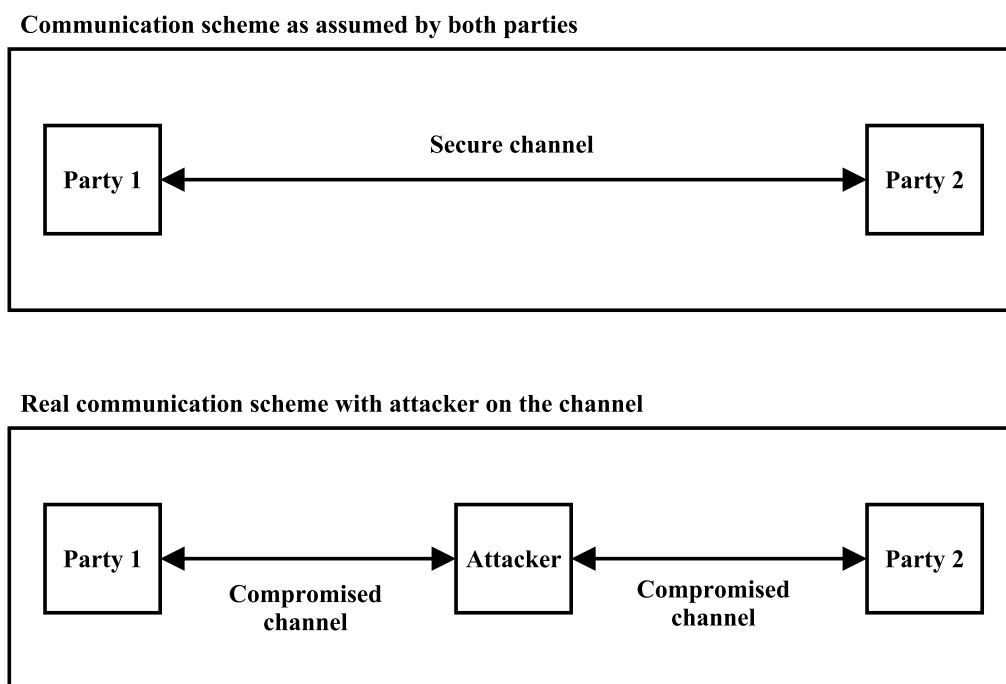


Fig. 18: *Man-in-the-Middle attack. The perpetrator injects themselves into the communication channel and passes messages between the two parties, arbitrarily modifying their content. Unless both victims possess knowledge of networking principles, MITM attack is not trivially detectable.*
 Source: own work.

Detecting MITM on unencrypted channels is not viable due to lack of reliable mechanisms indicating interference by a third party. In case of encrypted traffic, inspecting the digital certificate issued by a trusted Certificate Authority may provide user with a forewarning the connection is being redirected instead of going directly to the server. To automate the process, browser plugins supporting the functionality are offered. However, “[c]omputer users have been unconsciously trained for years that the absence of warning messages and popups means all operations were successful and nothing unexpected happened... In the SSL stripping attack,... the browser is never presented with any illegal SSL certificates since the attacker strips the whole SSL connection before it reaches the victim. With no warning dialogues, the user has little to no visual cues that something has gone wrong. In the case of SSL-only websites (websites that operate solely under the HTTPS protocol) the only visual cue that such an attack generates is the absence of lock icon somewhere on the browser’s window...” (Nikiforakis, Younan, & Joosen, 2010, p. 5). This makes MITM attack difficult to recognize without additional tools.

2.4.5 Social Engineering

Assuming a rational attacker (chapter 2.4.1) who wishes to maximize their utility, increases in security lower their payoff function and force them to choose different strategy. The scenario is similar to a game-theoretic imperfect information model which "...encompasses not only situations in which a player is imperfectly informed about about the other players' previous actions, but also, for example, situations in which during the course of the game, a player forgets an action that he previously took and situations in which a player is uncertain about whether another player has acted" (Osborne & Rubinstein, 1994, p. 197). Applying the approach to security is understandable as "[t]here is a need to predict the actions of both the defenders and the attackers. Since the interaction process between attackers and defenders is a game process, game theory can be applied in every possible scenario to predict the actions of the attackers and then to determine the decision of the defenders" (X. Liang & Xiao, 2013, p. 1). A degree of subjectivity must be expected due to assumptions about players' utility for each available option. As ICT infrastructure is monitored, hardened, and upgraded to meet the latest performance and security requirements (albeit in a delayed manner, opening windows of opportunity), one element remains relatively stable and unchanged over time: people.

Even though detractors emphasize that "... [i]nstead of spending time, money and human resources on trying to teach employees to be secure, companies should focus on securing the environment and segmenting the network. It's a much better corporate IT philosophy that employees should be able to click on any link, open any attachment, without risk of harming the organization," (Aitel, 2012) proponents stress that "... every social engineering attack will work on someone. Training simply raises the bar; it's not an impermeable shield... Ultimately, we believe awareness training is something all smart CSOs will continue to invest in, whether it is for the entire staff to understand the hostile environment around them or for developers..." (McGraw & Miguez, 2012). To subvert the victim, variety of tactics based on emotional stimuli (fear, distress, interest, joy) can be used to illicit desired response and manipulate the target into taking arbitrary action. R. J. Anderson (2008, p. 18) claims that "[d]eception, of various kinds, is now the greatest threats to online security. It can be used to get passwords, or to compromise confidential information or manipulate financial transactions directly... [One] driver for the surge of attacks based on social engineering is that people are getting better at technology. As designers learn how to forestall the easier techie attacks, psychological manipulation of system users or operators becomes even more attractive. So the security engineer simply must understand basic psychology and 'security usability'..."

Social engineering refers to "... various techniques that are utilized to obtain information in order to bypass security systems, through the exploitation of human vulnerability... [T]he human element is the 'glitch' or vulnerable element within security systems. It is the basic 'good' human natured characteristics that make people vulnerable to the techniques used by social engineers, as it activates various psychological vulnerabilities, which could be used to manipulate the individual to disclose the requested information" (Bezuidenhout, Mouton, & Venter, 2010, p. 1). Utilizing cognitive biases such as those presented later in chapter 6, social engineering aims to direct the target toward particular behavior sometimes contradicting their usual patterns to either help the attacker unwittingly, or act to avoid perceived harm. It comprises various methods; the two most widely-used are pretexting and phishing.

Pretexting is defined as "... getting private information about an individual under false pretenses," (Schwartz, 2006, p. 1) or as "... the background story, dress, grooming, personality, and attitude that make up the character you will be for the social engineering audit" (Hadnagy, 2010, p. 77). By accumulating and strategically presenting background information, the attacker

establishes a sense of legitimacy through impersonation to gather insights from the target. For example, publicly-available manuals and reference books may provide enough technical details and jargon to contact an employee who can provide temporary system access with the help of persuasion. After obtaining the login credentials, malware can be deployed on internal network, completely bypassing perimeter defenses. A significant risk of pretexting is that “[c]ompanies that fail to fully safeguard themselves against the pretexting tactics of others can compromise confidential data (including that entrusted to them by their customers), expose intellectual property, and prematurely reveal their plans to the outside world. By allowing themselves to fall prey to pretexting, these companies can lose the confidence of the market, suffer financial losses, and open themselves up to legal and regulatory exposure” (Leonard, 2006, p. 2). Pretexting in legal context (criminal investigations) has been known to take place, although it remains controversial whether information obtained in such way are ethically justifiable (S. C. Bennett, 2010). Countermeasures include authentication, data non-disclosure policy over the phone and email, training, assessing employees most likely to fall victim to social engineering attacks, and human vulnerability assessment as part of penetration testing (chapter 2.4.8). False positives and false negatives must be expected to occur, though.

Phishing attacks “. . . typically stem from a malicious email that victims receive effectively convincing them to visit a fraudulent website at which they are tricked into divulging sensitive information (e.g., passwords, financial account information, and social security numbers). This information can then be later used to the victim’s detriment” (Ramzan, 2010, p. 433). One of the first documented cases of phishing occurred in 1994; it “. . . involved tricking someone into trusting you with their personal information. In this case, a person who had just logged on to cyberspace for the first time would be fooled into giving up their password or credit card information” (Rekouche, 2011, p. 1). Primary communication medium for phishing is email (although instant messaging is also viable) purportedly coming from reputable entities, e.g., auction portals, banks, electronic mail providers, financial institutions, insurance agencies, online retailers, payment processors, and social networks. They prompt users to visit a link included in the email body apparently pointing to a legitimate website. There, the victim is asked for personally-identifiable information under false pretenses, such as to validate their account, receive compensation, etc. The data is, however, collected for immediate or later impersonation on sites whose look and user experience the attack emulated. Another result is malware infection for unfettered access to the victim’s station. Findings suggest “. . . phishing is evolving into a more organized effort. It is part of a larger crime eco-system, where it is increasingly blended with malware and used as a gateway for other attacks” (Sheng, Kumaraguru, Acquisti, Cranor, & Hong, 2009, p. 12). Phishing attempts have become more sophisticated to include formally- and grammatically-correct text and corporate logos which makes it harder for both automated systems (filters) and users to discern whether an attack is being attempted.

Mathematical and statistical methods have been employed to counter phishing. For instance, fraudulent emails were observed to contain predictable words and phrases, and automatically penalizing incoming messages based on this criterion is a naïve phishing classification rule. Ma, Ofoghi, Watters, and Brown (2009, p. 5) posit that “. . . content only classification is not sufficient against the attack. Ortographic features reflect the author’s style and habit so that the features are also informative as discriminators. Derived features are mined and discovered from emails which also provide clues for classification.” Apart from content, links are also scanned against blacklists: when a match is made, the email is discarded because the locator is provably malicious. As will be mentioned in chapter 6, managing blacklists becomes more complex with increasing size, rendering the measure only partially effective due to the need for regular updates. Heuristic methods have been proposed which forgo accuracy for speed and efficiency: Whittaker, Ryner, and Nazif (2010, p. 12) implemented a scalable machine learning classifier “. . . which maintains

a false positive rate below 0.1%... By automatically updating our blacklist with our classifier, we minimize the amount of time that phishing pages can remain active. . . Even with a perfect classifier and a robust system, we recognize that our blacklist approach keeps us perpetually a step behind the phishers. We can only identify a phishing page after it has been published and visible to Internet users for some time.” This reactive security approach is a serious disadvantage; even very brief windows of opportunity are enough to generate considerable amount of phishing emails. Yue Zhang, Hong, and Cranor (2007) devised a system combining eight characteristics, e.g., age of domain, suspicious Uniform Resource Locators (URLs) and links, number of dots in the URL, and others into a single score with a detection rate of 90 % and a false positive rate of 1 %. A real-time URL scanner was proposed, too, achieving accuracy of more than 93 % on training data sets (J. Zhang & Yonghao Wang, 2012). Phishing detection and classification remains an active research field.

Despite the tools presented above, it is often up to individuals to decide if the message is a phishing attempt. Several precautions exist to increase security when dealing with suspicious emails: inspecting the link closely and entering it as a search term into a search engine, using a browser plugin supporting blacklist validation as well as training and testing (Kumaraguru et al., 2007). Nevertheless, Dhamija, Tygar, and Hearst (2006, p. 9) point out that “. . . even in the best case scenario, when users expect spoofs to be present and are motivated to discover them, many users cannot distinguish a legitimate website from a spoofed website. . . . [p]hishers can falsify a rich and fully functioning site with images, links, logos and images of security indicators. . . ,” further stressing the need to combine end-user education with automated phishing classification tools.

2.4.6 Virtual Machines

Cloud computing, mentioned in chapter 2.3 is an umbrella term for infrastructure, platform, and application layers connected to facilitate convenient access to virtualized resources. Instances of varying configurations are at customer’s disposal as virtual machines (VMs) representing physical devices in software. VMs do not constitute single points of failure: by consolidating multiple configurations to run on a single set of hardware, they reduce the attack surface by decreasing probability of failure for each physical server replaced by its equivalent virtual substitute. This exposes two previously non-existent vulnerabilities:

- hardware shared among the VMs,
- software managing the VMs.

Exploiting hardware requires physical access but even without malicious attempts, server components have limited lifespan, as discussed in chapter 2.2.3. If no backup or failover mechanisms are present and thoroughly tested for correct functionality, they create a weak point with significant negative consequences ranging from decreased work productivity due to inaccessible electronic assets to complete infrastructure breakdown. Human factor is not considered vulnerable but social engineering can be used to manipulate cloud operator into helping the attacker, a situation mitigated by employee training and authentication procedures which identifies any party requesting remote assistance.

Targeting the software responsible for creation, management, and termination of VMs is more plausible as it can cause damage to as many as k machines, where k represents the number of nodes under a single master called Virtual Machine Monitor (VMM) or hypervisor. Popek and R. P. Goldberg (1974, p. 2) delimited its properties: “As a piece of software a VMM has three essential characteristics. First, [it] provides an environment for programs which is essentially

identical with the original machine; second, programs run in this environment show at worst only minor decreases in speed; and last, [it] is in complete control of system resources.” Figure 19 schematically depicts hypervisor controlling activities of subordinate machines. Attackers focus on hypervisors because they control each and every system resource. S. T. King et al. (2006, pp. 1–2) assert that “[c]ontrol of a system is determined by which [software] occupies the lower layer in the system. Lower layers can control upper layers because lower layers implement the abstractions upon which upper layers depend,” and present a tool showing “. . . how attackers can install a virtual-machine monitor (VMM) underneath an existing operating system and use that VMM to host arbitrary malicious software.” Such attack renders all security measures in the operating system useless as the hypervisor can manipulate resources to hide arbitrary activity. Achieving the same for multiple VMs gives unfettered control over the space the VMM manages. Hypervisor-based malicious software detection was demonstrated by Z. Wang, Jiang, Cui, and Ning (2009) who presented a legitimate application for such high-privileged software.

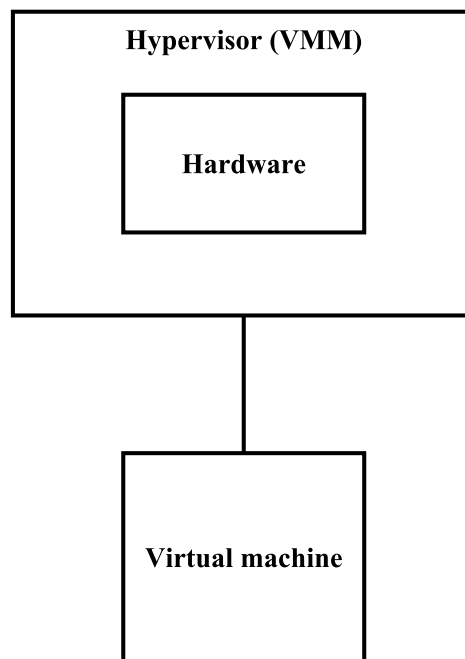


Fig. 19: *Hypervisor. Each virtual machine is dependent on the hypervisor which constitutes a single point of failure for all nodes under its control. Being a software tool, it is prone to bugs, exploits, and instabilities resulting from improper patch management policy.*

Source: Popek and R. P. Goldberg (1974, p. 2), modified.

In a data center, VMs from multiple customers are collocated and run on single physical hardware. The multi-tenant architecture pits together mutually distrusting parties with potentially malicious intents toward the rest with a reasonable assumption the hypervisor will be preferentially targeted as “[e]xploiting such an attack vector would give the attacker the ability to obstruct or access other virtual machines and therefore breach confidentiality, integrity, or availability of other virtual machines’ code or data” (Szefer, Keller, Lee, & Rexford, 2011, p. 1). Despite Roscoe, Ephinstone, and Heiser (2007, p. 6) claiming virtualization is “short-term, immediately applicable, commercially relevant, but cannot be described as disruptive, since it leaves most things completely unchanged,” it has become a tool of choice in many fields including business, and hypervisor protection thus an important issue. Ways have been proposed minimizing the number of VMs the VMM handles down to a single instance (Seshadri, Luk, Qu, & Perrig, 2007) which is less error-prone with at most one machine affected by hypervisor compromise. Another approach guarantees integrity for a subset of virtual machine control software tools (Z. Wang &

Jiang, 2010) while yet other strips non-essential parts “. . . to remove attack vectors (in effect also reducing the hypervisor) while still being able to support the hosted cloud computing model” (Szefer et al., 2011, p. 10). Virtualization security is an active field of research but as pointed out by Christodorescu, Sailer, Schales, Sgandurra, and Zamboni (2009, p. 1), “[w]hile a large amount of research has focused on improving the security of virtualized environments,. . . existing security techniques do not necessarily apply to the cloud because of the mismatch in security requirements and threat models.”

Users should treat VMs as untrusted components helping them get access to physical resources such as CPU cycles, storage capacities, and networks. I. Goldberg, Wagner, Thomas, and Brewer (1996, p. 2) assert that “. . . an outsider who has control over the helper application must not be able to compromise the confidentiality, integrity, or availability of the rest of the system. . . [W]e insist on the Principle of Least Privilege: the helper application should be granted the most restrictive collection of capabilities required to perform its legitimate duties, and no more. This ensures that the damage a compromised application can cause is limited by the restricted environment in which it executes.” This principle was discussed in chapter 2.2.1. If the attacker takes control of some part of the VM, they can disrupt the service only for a single instance because their permissions should not span manipulation of other VMs. Data center operators prevent external intrusions but must be also able to detect them in progress internally which requires tools to remotely monitor the machines in search for patterns associated with malicious activities. “Intrusion preventers work by monitoring events that enter or occur on the system, such as incoming network packets. Signature-based preventers match these input events against a database of known attacks; anomaly-based preventers look for input events that differ from the norm” (P. M. Chen & Noble, 2001, p. 3). Virtual Machine Introspection (VMI), penetration testing, and forensic analysis are utilized to pinpoint vulnerabilities but “[a]ll these tasks require prior knowledge of the [exact] guest OS version. . . Although the cloud users may provide the information about the OS version, such information may not be reliable and may become out dated after the guest OS is patched or updated on the regular basis” (Gu, Fu, Prakash, Lin, & Yin, 2012, p. 1). Precise and usable fingerprinting techniques are needed for virtualization security.

Two notable attacks have been presented leveraging vulnerabilities in the cloud environment. The first presupposes co-location on the same physical server as the victim. By exploiting the hypervisor or a side channel, various data (estimates of traffic rates, keystrokes) are collected in real-time. Ristenpart, Tromer, Shacham, and Savage (2009, p. 14) argue that “. . . fundamental risks arise from sharing physical infrastructure between mutually distrustful users, even when their actions are isolated through machine virtualization as within a third-party cloud compute service.” They recommend allowing customers to select locations for their VMs to blind side-channel attacks, and obfuscating VM placement policy. The second attack also requires neighboring instances and extends the previous one by exploiting “. . . side-channel attacks with fidelity sufficient to exfiltrate a cryptographic key from a victim VM. . .” (Yingian Zhang, Juels, Reiter, & Ristenpart, 2012, p. 11). Using the technique, data about a key for electronic mail encryption was used to reconstruct it in whole, rendering the security measure ineffective.

2.4.7 Web

Internet infrastructure has grown rapidly along with the complexity of the tools designed to maintain and manage it. Simultaneous resource centralization and decentralization lead to data aggregated into large collections geographically distributed for redundancy, backup, error correction, and fast recovery in case of disruptions. To enforce consistency in presentation, manipulation, and management, a standard was needed which would make managing the collections

(databases) efficient. Historically first database management system (DBMS) was presented by Chamberlin and Boyce (1974, p. 2): "... Structured English Query Language (SEQUEL)... is consistent with the trend to declarative problem specification. It attempts to identify the basic functions that are required by data base users and to develop a simple and consistent set of rules for applying these functions to data. These rules are intended to simplify programming for the professional and to make data base interaction available to a new class of users... Examples of such users are accountants, engineers, architects, and urban planners." SEQUEL was based on work of Codd (1970, p. 2) who had devised a relational model and had argued that "[i]t provides a means of describing data with its natural structure only – that is, without superimposing any additional structure for machine representation poses. Accordingly, it provides a basis for a high level data language which yield maximal interdependence between programs on the one hand and machine representation and organization of data on the other. A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations..." Due to licensing issues, SEQUEL was later renamed to SQL. Multiple commercial implementations had been marketed before standardization by the National Institute of Standards and Technology (NIST) and the International Organization for Standardization (ISO). This introduced incompatibilities which limit database portability among different products, leading to vendor lock-in (mentioned in chapter 2.2.3).

SQL has become widely deployed as a back-end solution for web applications, accompanied by additional software tools. A popular open-source web platform consists of Linux operating system, Apache HTTP Server, MySQL database manager, and PHP, Perl, or Python programming languages, and is shortened as LAMP (D. Dougherty, 2001). The programs work in conjunction to deliver resources to entities who remotely requested them; each, however, forms a complex system with exploitable attack vectors. By combining and making them interdependent with bi-directional interactions, the threat surface is expanded considerably. Vulnerabilities in web server applications can be highly destructive, and must be given priority in policies to ensure patches are deployed with minimum delay. Malicious techniques exist targeting fundamental configuration omissions in many SQL-supported web servers which are trivial to automate, extending their reach and potency.

The most well-known is SQL injection, defined as "... an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives... The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed" (Microsoft, 2013). It was first described as a means of "... piggy-backing SQL commands onto a command that will work... If the normal page can get to the SQL server through a firewall, VPN, etc[.], then so can this command. It can, and will, go whenever the normal page/SQL can go... [T]here's a stored procedure in SQL that lets you email results of a command to anywhere..." (Puppy, 1998). The author also suggested a remedy to harden the database infrastructure and render SQL injection ineffective. This form of vulnerability reporting is called full disclosure and contrary to non-disclosure, the proponents argue that "[p]ublic scrutiny is the only reliable way to improve security, while secrecy makes us only less secure... Secrecy prevents people from accurately assessing their own risk. Secrecy precludes public debate about security, and inhibits security education that leads to improvements. Secrecy doesn't improve security; it stifles it" (Schneier, 2007).

The attack is consistently mentioned as a major threat to web applications and classified as having easy exploitability, common prevalence, average detectability, but severe impact which "... can result in data loss or corruption, lack of accountability, or denial of access. Injection can

sometimes lead to complete host takeover,” (OWASP, 2013, p. 7) and organizations are urged to “[c]onsider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted” (OWASP, 2013, p. 7). SQL injection can affect the constituents of the CIA triad:

- confidentiality: if the targeted database stores sensitive assets (login credentials, financial data), they can be exfiltrated; if encryption is used, offline brute-force and dictionary attacks described in chapter 2.4.2 can be mounted,
- integrity: the attacker is able to arbitrarily and selectively change database contents and introduce modified data which is subsequently used as input to processes and treated as genuine if metadata or cryptographic checksums (chapter 2.2.2) do not exist,
- availability: SQL syntax contains statements to delete specific entries or the whole table, view, and database which prevents legitimate queries from retrieving data, limiting availability and necessitating corrective measures.

Various types of SQL injection exist but all share crafting a query conforming to the standard and passing it to the database server either through URLs, forms embedded on a page, or any element which sends “. . . an SQL query in such a way that part of the user’s input is treated as SQL code. . . . The cause of SQL injection vulnerabilities is relatively simple and well understood: insufficient validation of user input. To address this problem, developers have proposed a range of coding guidelines. . . such as encoding user input and validation. . . . However, in practice, the application of such techniques is human-based and, thus, prone to errors. Furthermore, fixing legacy code-bases that might contain SQL injection vulnerabilities can be an extremely labor-intensive task” (Halfond, Viegas, & Orso, 2006, p. 1). Restricting database permissions by means of ACLs detailed in chapter 2.2.1 to disable reading from and writing into sensitive tables can limit impact of SQL injection. Backward compatibility concerns and unpatched LAMP stack components are the most common root causes of why SQL injection dominates network security threat landscape. Compatibility concerns can be addressed by gradual, long-term infrastructure upgrades, vulnerable software requires diligent patch management policy for a flexible response to existing and novel exploits.

Targeting web applications has become commonplace because the TCP/IP suite is kept backwards compatible and largely unchanged so that legacy devices can operate despite their moral or technological obsolescence. The attacks are generally categorized as follows: URL interpretation, input validation, SQL injection, impersonation, and buffer overflow (Shah, 2002, p. 10). While none will be discussed further, automated tools were developed to assist in carrying them out.

Threat actors are recruited from inside organizations as well, giving rise to insider attacks dangerous primarily due to their knowledge of ICT processes, countermeasures, and internal network topology. Randazzo and Cappelli (2005, p. 30) studied implications of insider threat in banking and finance sectors and concluded that “. . . insider attacks. . . required minimal technical skill to execute. Many of the cases involved the simple exploitation of inadequate practices, policies, or procedures. . . . Reducing the risk of these attacks requires organizations to look beyond their information technology and security to their overall business processes. They must also examine the interplay between those processes and the technologies used. . . . Comprehensive efforts to identify an organization’s systemic vulnerabilities can help inform mitigation strategies for insider attacks at varying levels of technical sophistication.” Security auditing in the form of penetration testing focuses on discovering open attack vectors exploitable by both internal and external parties.

2.4.8 Penetration Testing

Finally, penetration testing (also shortened as pentesting) will be defined and introduced. A part of information technology security audit, it “. . . goes beyond vulnerability testing in the field of security assessments. Unlike [vulnerability scanning] – a process that examines the security of individual computers, network devices, or applications – penetration testing assesses the security model of the network as a whole. Penetration testing can reveal to network administrators, IT managers, and executives the potential consequences of a real attacker breaking into the network. . . . Using social-engineering techniques, penetration tests can reveal whether employees routinely allow people without identification to enter company facilities and gain unauthorized access to a computer system” (EC-Council, 2010, p. 2). Penetration testing is defined as “. . . a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure. The process includes probing for vulnerabilities as well as providing proof of concept (POC) attacks to demonstrate the vulnerabilities are real” (Engebretson, 2011, p. 1). Herzog (2010, p. 37) specifies it as a double blind test where “[t]he Analyst engages the target with no prior knowledge of its defenses, assets, or channels. The target is not notified in advance of the scope of the audit, the channels tested, or the test vectors. [It] tests the skills of the Analyst and the preparedness of the target to unknown variables of agitation. The breadth and depth of any blind audit can only be as vast as the Analyst’s applicable knowledge and efficiency allows.” A test type scheme is depicted in Figure 20.

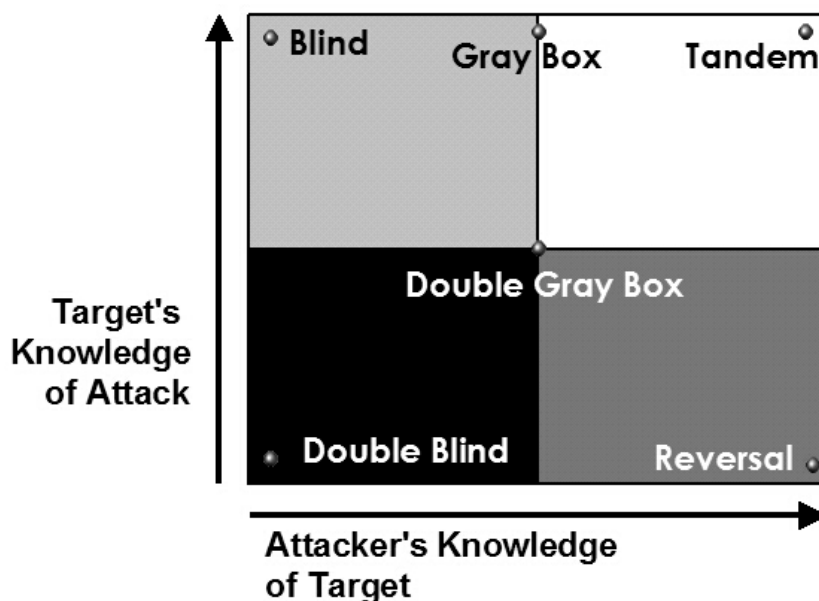


Fig. 20: Common test types. They are based on agreement between the analyst and the target requesting the audit. Double blind (black-box) assessment where the analyst has no prior knowledge of the system and the operator lacks information on how or when the test will be performed, is a penetration test. Source: Herzog (2010, p. 36).

Penetration testing is the most accurate representation of real-life scenarios, but perhaps also the most costly and time-consuming: the perpetrator does not have any knowledge of system capabilities, security measures, and policies the target enforces, propensity to social engineering manipulation, physical and network deterrents, ICT infrastructure readiness and resilience, and other exploitable vulnerabilities. At the same time, the entity under investigation does not know intruder’s capabilities, tools they will use, nor the time when the attack will be launched, and must be continuously vigilant to patterns indicating perimeter or internal breach.

Several reasonable assumptions can be made with respect to adversary profile (chapter 2.4.1): identical tools available to the attacker will be utilized, multiple vectors analyzed and combined to increase probability of success, employees likely targeted, and the system breached if no patch management, risk mitigation, and security policies have been put in place. The black-box approach contrasts with a white-box methodology presupposing full information on both sides which simulates insider threat; its purpose is to evaluate damage such intruder could cause. Gray-box test models the attacker as possessing partial information prior to audit commencement. This is corroborated by Scarfone, Souppaya, Cody, and Orebaugh (2008, p. 39) who state the “[t]ests should reproduce both the most likely and most damaging attack patterns – including worst-case scenarios such as malicious actions by administrators. Since a penetration test scenario can be designed to simulate an inside attack, an outside attack, or both, external and internal security testing methods are considered.”

Several frameworks for penetration testing exist. The National Institute of Standards and Technology, which defines it as a “. . . security testing in which assessors mimic real-world attacks to identify methods for circumventing the security features of an application, system, or network,” (Scarfone et al., 2008, p. 36) recommends segmenting the test into four phases, as demonstrated in Figure 21.

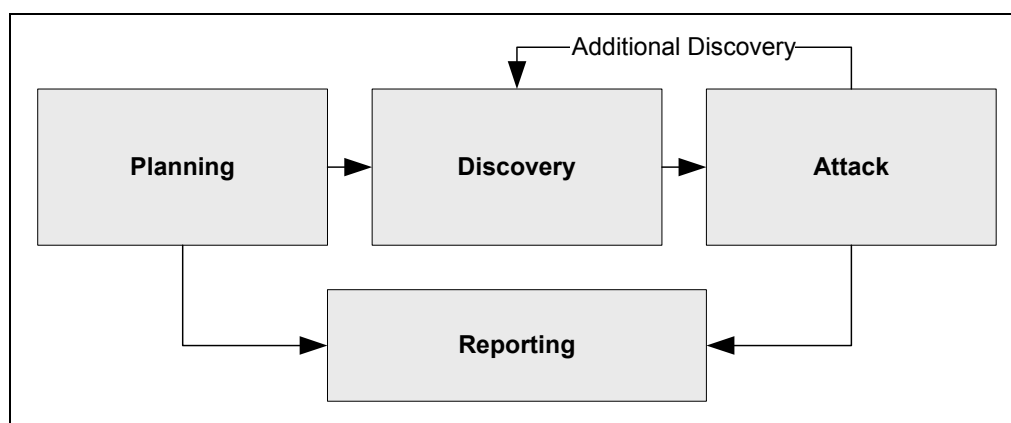


Fig. 21: *Penetration testing phases. After delimiting rights and responsibilities, information gathering which leads to exploitation of discovered vulnerabilities, is performed. The target is extensively informed on the findings to initiate mitigation procedures and infrastructure hardening.*
 Source: Scarfone, Souppaya, Cody, and Orebaugh (2008, p. 37), modified.

In the first phase, specifics, scope, legal implications, and other details pertaining to the test are negotiated and contractually agreed upon by both parties. Discovery phase consists of reconnaissance, information gathering, system fingerprinting, harvesting publicly-available data about the target (ICT infrastructure specifics, employee contact details, metadata extraction, physical security) which lead to a comprehensive target profile and a strategy for the attack stage. Discovery phase is a “. . . roadmap for a security testing effort, including a high-level overview of the test cases, how exploratory testing will be conducted, and which components will be tested” (H. H. Thompson, 2005, p. 2). The Social-Engineering Toolkit (SET), Metasploit, and Nessus Vulnerability Scanner offer unified environments for information gathering, vulnerability scanning and identification as well as basic and advanced offensive capabilities.

Even when penetration testing is not employed, reputable sources, e.g., National Vulnerability Database⁶, Open Sourced Vulnerability Database⁷ (OSVDB), and Common Vulnerabilities and

⁶<https://nvd.nist.gov/>

⁷<http://www.osvdb.org/>

Exposure⁸ (CVE) should be consulted for up-to-date information. Security advisories from vendors acknowledge defects in products and release updates which prevent further exploitation, closing the respective window of opportunity if tested and deployed immediately. Failure to do so opens vectors for unauthorized system access. The portals aggregate actionable code samples for software commonly found on target systems, adhering to the full-disclosure policy mentioned previously.

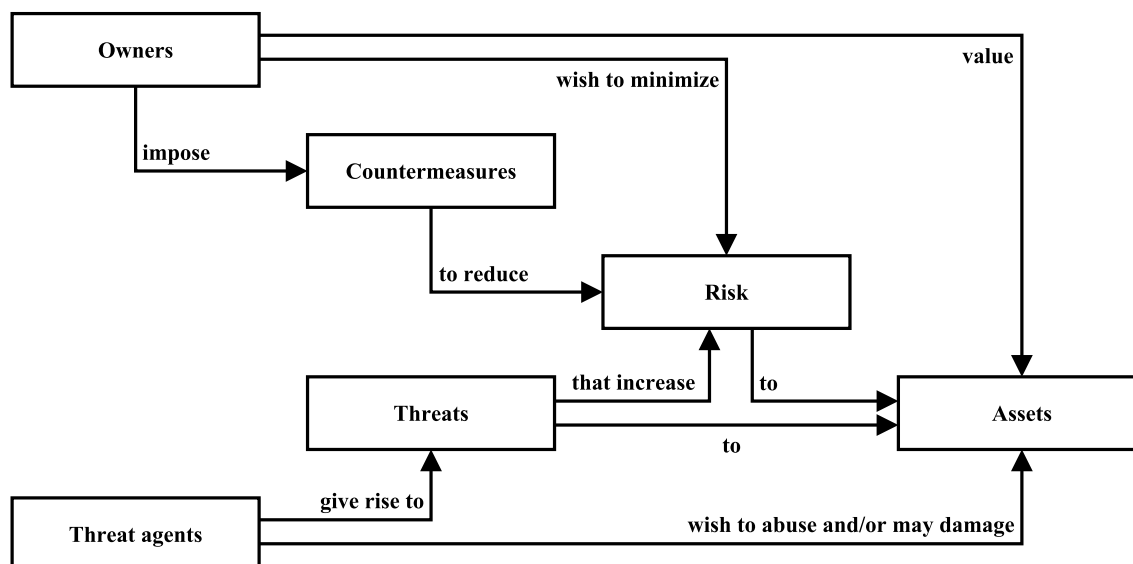


Fig. 22: Security concepts and relationships. Risk is a central point where owners and threat agents meet: owners strive to minimize risk by imposing countermeasures while threat agents seek to maximize it by threatening asset confidentiality, availability, and integrity. Source: Common Criteria (2012, p. 39), modified.

The attack stage then consists of initiating a connection, injecting the malicious payload, e.g., an SQL query which forces the DBMS into an unhandled state where the adversary escalates their privileges above those for which they are authorized. Internal analyses may reveal further vulnerabilities, creating a feedback loop back to the discovery phase. Maintaining system presence for easy future access is frequent. Findings are reported throughout the whole testing cycle; at the test conclusion, a comprehensive report is delivered to the victim. It is “. . . a critical output of any testing process. For security problems, report formats can vary, but they must at least include reproduction steps, severity, and exploit scenarios” (H. H. Thompson, 2005, p. 3). Steps to close the particular attack vector are usually provided as well.

Another framework is Open Source Security Testing Methodology Manual (OSSTMM) whose primary purpose is “. . . to provide a scientific methodology for the accurate characterization of operational security. . . through examination and correlation of test results in a consistent and reliable way” (Herzog, 2010, p. 13). Available free of charge, it is used in diverse situations, e.g., measuring security benefits of networking solutions (Endres, 2012).

In security concepts and relationships presented in Figure 22, “[s]afeguarding assets of interest is the responsibility of owners who place value on those assets. Actual or presumed threat agents may also place value on the assets and seek to abuse the assets in a manner contrary to the interests of the owner. Examples of threat agents include hackers, malicious users, non-malicious users (who sometimes make errors), computer processes and accidents” (Common Criteria, 2012, p. 39). The analyst is also classified as a threat agent due to methods employed

⁸<https://cve.mitre.org/>

which increase risk to assets and threaten to impair them. This necessitates the owner to apply countermeasures, e.g., firewall, IDS, policies, plans, training, and auditing to disallow asset tampering or misappropriation. However, “[m]any owners of assets lack the knowledge, expertise or resources necessary to judge sufficiency and correctness of the countermeasures. . . These consumers may therefore choose to increase their confidence in the sufficiency and correctness of some or all of their countermeasures by ordering an evaluation of these countermeasures” (Common Criteria, 2012, p. 40). Security auditing and penetration testing are specifically designed to meet these demands.

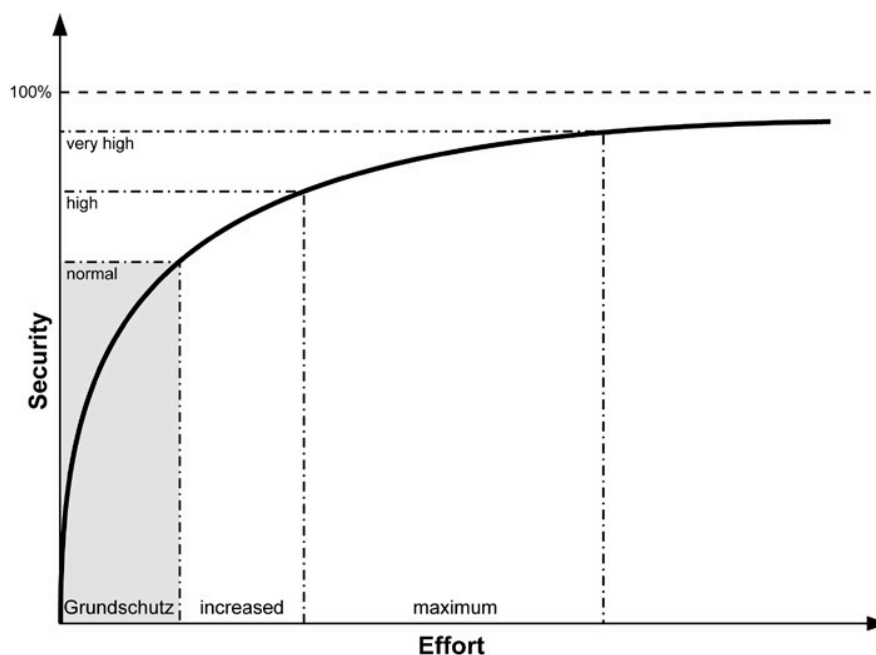


Fig. 23: Cost/benefit for information security. As the expenditures increase, gains in security gradually diminish to the point where additional spending to attain higher level of security does not produce noticeable gains.

Source: BSI (2009, p. 30).

When the victim is presented with an assessment report suggesting ways to improve the infrastructure, cost-effectiveness should be taken into consideration. If the target system operated with limited attention to security, it benefits from any additions substantially as measured by the number of exploitable attack vectors or metrics such as *rav*, introduced in chapter 2.1.5. However, if additional costs are incurred for the same purpose, the utility decreases akin to the law of diminishing marginal utility. Moreover, “[e]xperience has shown that the relationship between the expense required to increase the security level and the actual gain in security attained through this expense leads to diminishing returns as the desired security level increases. It is not possible to achieve perfect information security. . . .” (BSI, 2009, p. 30). Changing or establishing ICT processes may also result in benefits: “Frequently, simple organisational rules that can be implemented without great additional cost or additional technical equipment make a substantial contribution to improving the security level. . . . Above all, it is important to point out that investing in human resources and organisational rules is often more effective than investing in security technology. Technology alone does not solve any problems since technical safeguards always need to be integrated into a suitable organisational framework” (BSI, 2009, p. 30).

The cost/benefit relation diagram for information security is depicted in Figure 23. The curve is asymptotic, indicating perfect security is unachievable. As new vulnerabilities are discovered and novel attack vectors exploited, converging toward the ideal state necessitates continuous

assessment, monitoring, mitigating, and responding to threats detrimental to security as well as nurturing preparedness in employees because human element forms a critical part of any ICT infrastructure.

3 GOALS, METHODS

In this chapter, the complex topic of ICT security will be broken down into manageable parts, areas of interest for the doctoral thesis, goals and hypotheses specified, and methods used to achieve them detailed. For example, security is a multifaceted concept which cannot be covered in full without undue generalization which would make the results unlikely to ever be picked up and implemented because of lacking specificity. A balance must therefore be found between how many details to include (low-level approach) while retaining enough common features for organizations to be able to implement the results without considerable modifications (high-level approach). The author believes that combination of both will ensure all stakeholders are thoroughly informed: preferring one approach over the other would inadvertently lead to the work either quickly falling out of date due to technological advances, or it having little relevance because of overly general conclusions. While some parts will exhibit imbalance toward one or the other, the thesis will draw from both equally when it comes to setting up the ICT security governance model.

Visualizing order of actions undertaken to achieve each goal will demonstrate dependencies and provide overview of how the selected methods will be applied in real-world research settings to obtain primary data. The data will then be utilized to support or reject hypotheses corresponding to individual goals which support the model, serving as the thesis' primary scientific output. The diagram with milestones is depicted in Figure 24. The process can be divided in two parts: in the first part, an area of interest was selected and literature review conducted to narrow down the space of candidate topics out of which a suitable option was picked and expanded to include goals and methods. This resulted in a preliminary research plan. The thesis defense proposed changes which were then incorporated along with findings from a second round of literature review focusing on BYOD, ICT, penetration testing, and security to coincide with the newly-devised goals and methods: questionnaires and case studies. Each will serve as a means to evaluate the particular hypothesis and provide data to extract factors pertaining ICT security. Finally, the findings will be synthesized to create a model based on situations encountered in practice to ground it in real-world observations.

Analysis of secondary sources, especially the first round, covered broad range of topics to gauge different research avenues. A fusion of economics and computer science, the latter defined as "... the systematic study of the feasibility, structure, expression, and mechanization of the methodical processes (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information," (Sheppard, 2003, p. 1) offers many opportunities to explore but also assumes prior knowledge of fundamental principles which is not always met.

A common ground needed to be found emphasizing elements understandable to both economists and IT managers who may expect low-level description of the challenges the thesis aims to address. Despite the author's best effort, however, some aspects will be new to both groups, requiring further study to fully grasp the underlying concepts. The intended audience are middle managers dealing with ICT in various industries, though financial and health will not be considered as legislative acts which regulate sensitive data processing there are beyond the thesis' purview. General principles can be applied to any organization regardless of form and size without extensive modifications because the security principles are largely identical. Regardless, many details will have to be omitted to maintain the balance.

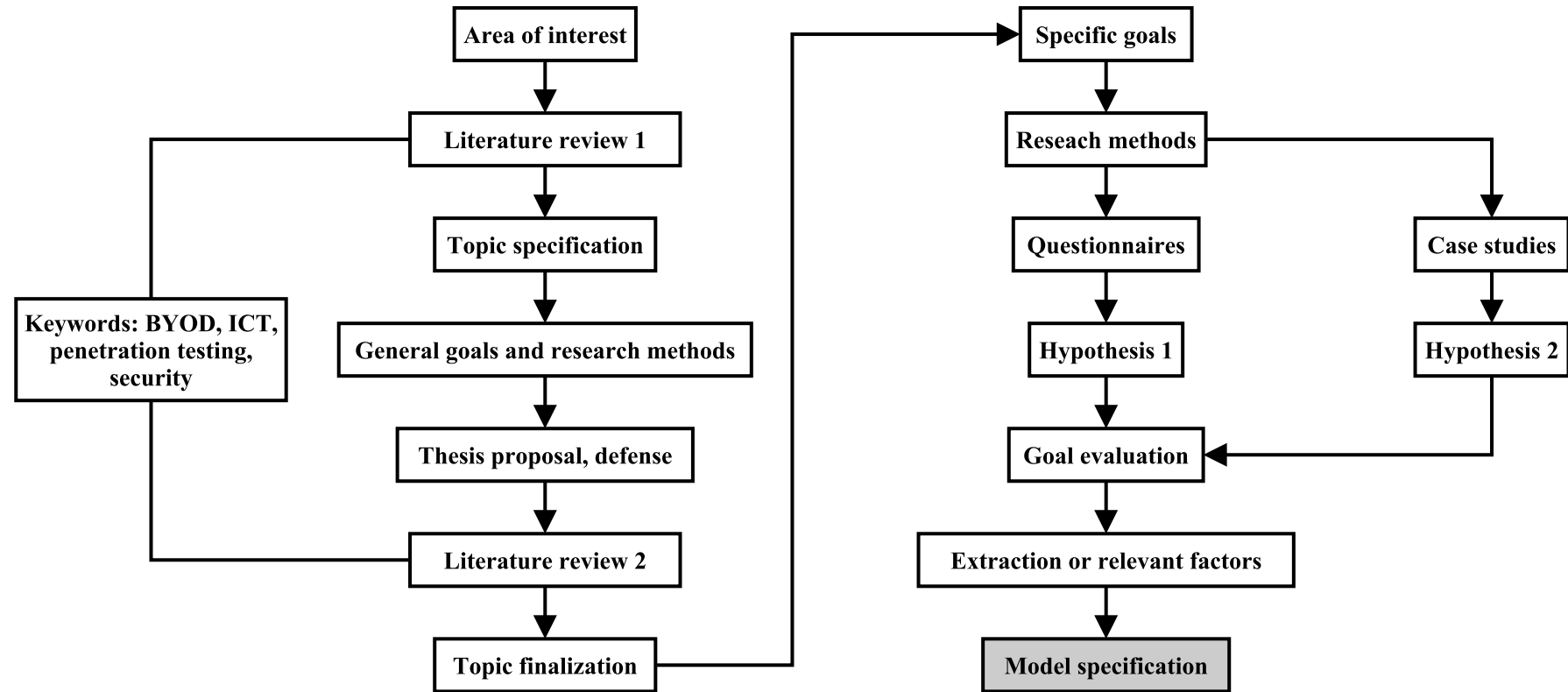


Fig. 24: Thesis milestones. Topic modifications reflecting changes in focus were performed during the literature review phase and after the thesis defense. At first, goals were devised in general terms and later specified when suitable options identifying areas which in the author's opinion allowed to pursue viable, interesting as well as practical research efforts, were found.

Source: own work

The model and best practices outlined in later chapters should not be deployed to supplant ICT risk process management in full but rather serve to re-evaluate security in key areas which the model addresses, particularly focusing on the human element as it is frequently targeted when other aspects of ICT infrastructure (hardware, software, processes) were hardened.

Disclaimer: While including a disclaimer may seem excessive, the author is of the opinion academic literature often contains data originating from legal entities who explicitly requested it to remain anonymous. Considering the importance of ICT security, priority should be assigned to remove any and all pieces of information which can be used to circumvent security. This premise will be respected and intentionally pursued in the thesis. The author does not condone any malicious practices demonstrated herein, they serve only to demonstrate vulnerabilities in real-world electronic systems and do not constitute endorsement or encouragement to execute them on live websites on the Internet. The author is likewise not responsible nor liable for any damage caused by emulating the techniques and running the tools against targets without prior consent. All testing was conducted with prior electronic and/or verbal consent from the affected parties which were also notified of the results and assets accessed during the tests, if applicable. All exploitable vulnerabilities were reported to appropriate personnel along with recommendations on how to mitigate the associated risk. No electronic nor physical copies of sensitive data were retained; graphical representations (screenshots) demonstrating proofs of concept were obtained but key textual information sanitized (blacked out) to ensure reverse image processing algorithms cannot be used to recover the information which may help to mount identical attacks. Lastly, legal counsel was sought regarding the case studies, and it was established the data used for analyses do not constitute personally-identifiable information as per the Personal Data Protection Act (101/2000) enforced in the Czech Republic.

3.1 Goals

The doctoral thesis has one primary goal and two auxiliary goals whose fulfillment will support achievement of the primary one. Both auxiliary goals will in turn be evaluated using data collected in field research and analyzed using statistical or other tests described further in chapter 3.2. As each link in the research chain builds on scientific methods, erroneous results should be kept to minimum except for instances where data is not representative. While care will be taken to ensure such situations do not occur, the possibility cannot be completely ruled out particularly in the questionnaire research as the sample size will be finite and inferences drawn from a collection which may exhibit skewed properties. Analyzing deviations from “reality” in the data through critical thinking and confronting the findings with expected, ideal scenarios could help to partially alleviate incorrect results. However, in case the tendencies are statistically significant even after accounting for limited sample size, another explanation, namely that the results represent a newly-emerging trend, will be also considered.

The **main goal** is to **strengthen the organizational sensitive electronic data and ICT security processes by addressing selected cybernetic risks and techniques for unauthorized access tied to mobile and other devices interacting with the ICT infrastructures and accessing electronic assets by devising a model and introducing best practices applicable to real-world, practical conditions.** The terms and concepts were all delimited previously:

- cybernetics: chapter 2.1.1
- data: chapter 2.1.2
- process: chapter 2.1.3
- risk: chapter 2.1.4

- security: chapter 2.1.5
- accessing assets: chapter 2.2
- mobile devices: chapter 2.3
- techniques for unauthorized access: chapter 2.4.

The goal is thus to create a model and provide a list of best practices which can be integrated into organizations dealing with processing and security of sensitive data. The goal formulation was devised by evaluating a complex array of interconnects and interactions between threat actors, users, objectives, policies, and attacks which are schematically depicted in Figure 25.

Considering the scope and breadth of the ties, the thesis will deal only with their subset: if all elements from the map were selected, the model and best practices would have to present them on a high level, omitting majority of specifics which are crucial in practice as they constitute exploitable attack vectors when dealt with improperly. By limiting the research to several closely-related areas, a low-level analyses could be performed and synergic¹ effects achieved: definitions common for multiple areas, concepts overlapping, reinforcing, and supporting one another, challenges in one area present in others, etc. Instead of several unrelated and loosely-connected topics, effort was made to group them based on a common stem: ICT security policy. Despite belonging to the same category, blacklisting/whitelisting and patch management will not be included, although they will be mentioned. Blacklisting and whitelisting can be automated while patch management heavily depends on the industry the organization operates in, is heterogeneous, and many scenarios would have to be covered. Systems requiring continuous availability (electronic shops, finance-related activities) necessitate conservative patch deployment strategies as any stability issues, downtimes, or disruptions result in substantial losses. On the other hand, ICT of smaller enterprises is comparatively more resilient, especially when it only supports business processes instead of being its primary, mission-critical component.

The main goal takes two inputs (findings from the auxiliary goals) and produces one output: a model with best practices and policies applicable to organizational ICT. An evaluation on why the topics were selected is presented in chapter 3.3.

¹['sɪnərdʒi], [*uncount., count.*] (*technical*): the extra energy, power, success, etc. that is achieved by two or more people or companies working together, instead of on their own (Oxford University Press, 2011).

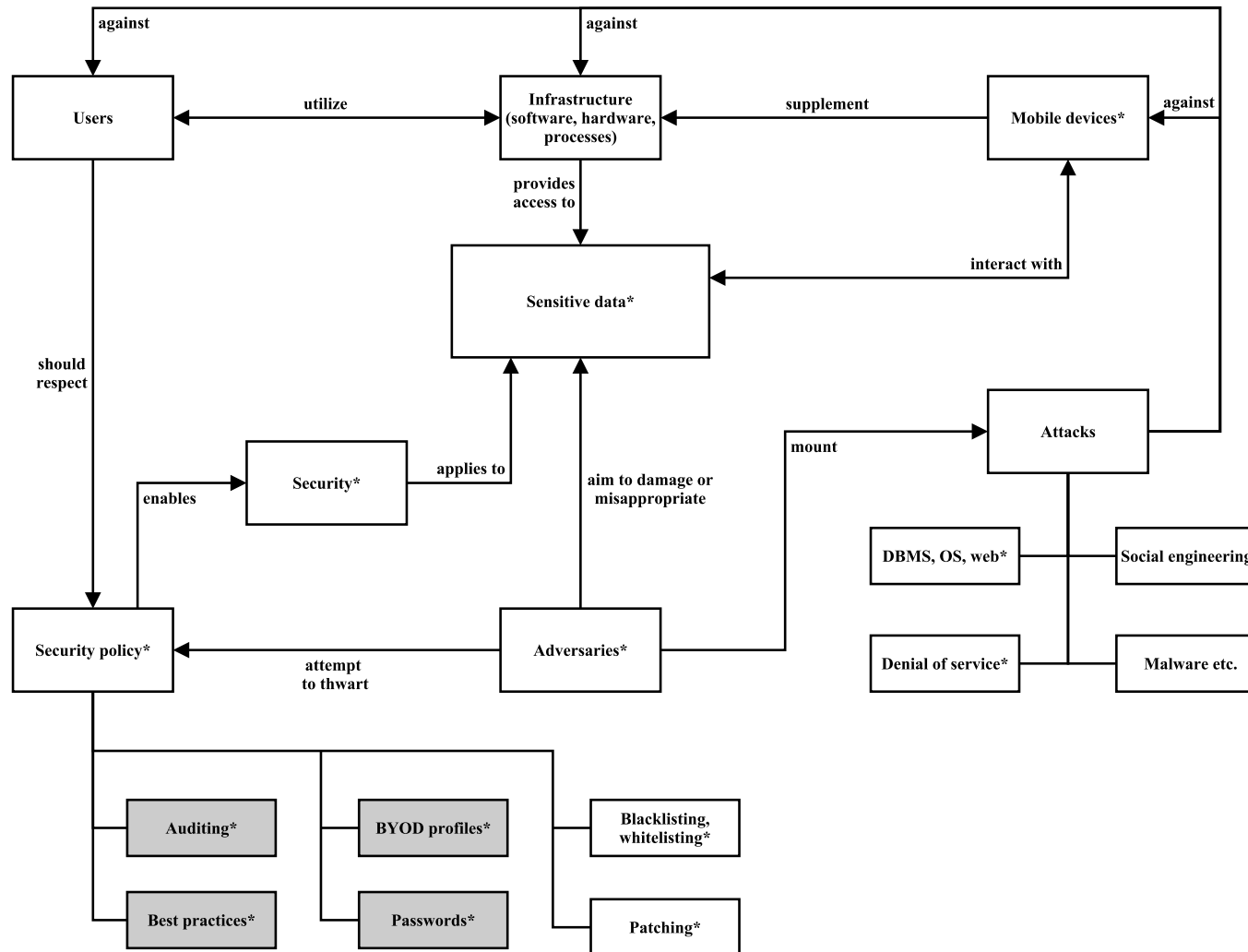


Fig. 25: Elements influencing security. Attacks affect sensitive data negatively while for users, infrastructure, and mobile devices, opportunities and threats exist simultaneously based on security policy features. Terms marked with (*) were discussed in previous chapters, gray boxes denote keywords used in the proposed ICT model.

Source: own work

The first auxiliary goal was formulated based on the following scientific question: “What levels of computer literacy and tendencies to obey security rules when on mobile devices and personal computers are reasonable to expect in a representative sample of users?” A hypothesis presupposing a particular outcome is: “Users handle electronic devices (PC, mobile phone) without enforcing baseline security practices even though a notion of what they constitute and what the potential threats could be are known.” To test its validity, questionnaires were selected as a suitable research method. They are briefly outlined in chapter 3.2 and more thoroughly in chapter 4. The output will be **a collection of best practices and recommendations understandable to users with little to no security background**. The practices will comprise BYOD, Internet, passwords, and social engineering. Exploits and mitigation procedures will be mentioned as well.

The second auxiliary goal was formulated based on the following scientific question: “Do selected areas of user-side and ICT-side security contain vulnerabilities which would allow even a low-skilled adversary to gain unauthorized system access should suitable techniques be applied?” A hypothesis presupposing a particular outcome is: “Users and organizations underestimate risks posed by unsophisticated adversaries and engage in substandard security practices which could be targeted and exploited with minimum level of knowledge and freely-available tools.” To test its validity, case studies were selected as a suitable research method. They are briefly outlined in chapter 3.2 and more thoroughly in chapter 5. The output will be **a comprehensive report detailing shortcomings found during the testing which will form a basis for the ICT security governance model**. Each case study will be focus on a topic deemed important by the author.

Auxiliary goals complement each other: questionnaires will supply information about how various aspects of ICT are perceived. The findings will be incorporated into the proposed security model because the answers may uncover relevant, up-to-date vulnerabilities in the human element of security. The delay between data analysis and its transformation into concise, usable form will be at best several weeks which will make the vectors pertinent to test in practical situations. The questionnaires and the case studies will be limited geographically to the Czech Republic. The model will thus reflect on what takes place in the consumer and organizational sectors, and the proposal will address vulnerabilities found while incurring minimal costs and user comfort penalties. Neither user-side nor ICT-side will be preferred, and the proposed framework balances best practices for both so that the security landscape is explored comprehensively.

The goals are summarized in Table 7. The amount of auxiliary goals were kept to a minimum by focusing on the most important aspects of security. Because each goal should be supported by original research, inflating their number would prolong planning, organizing, executing, and analyzing results to discover patterns and tendencies then used to produce the output. In the author’s opinion, the structure is sound and does not exhibit redundancies while at the same time, hypotheses share common ground with each element supporting others. The author also believes the thesis is of imminent practical benefit and reflects shifts currently in progress in industries and among consumers.

3.2 Methods

To integrate data from different sources into a cohesive unit, various methods will be employed ranging from general (abstraction, analogy) to specific (case studies, questionnaires): general methods are not rigorously defined and can be custom-tailored for different situations, specific methods necessitate that the researcher sets what they want to achieve beforehand, and make

Tab. 7: Summary of thesis goals. Auxiliary goals support completion of the main goal by analyzing data obtained from consumers and case studies.

Source: own work.

Goal	Definition	Input	Research method	Output
Main	Strengthen the organizational sensitive electronic data and ICT security processes by addressing selected cybernetic risks and techniques for unauthorized access tied to mobile and other devices interacting with the ICT infrastructures and accessing electronic assets by devising a model and introducing best practices applicable to real-world, practical conditions	Best practices, security audit report	Abstraction, analogy	Model
Auxiliary 1	A collection of best practices and recommendations understandable to users with little to no security background	Real-world data	Questionnaires	Best practices
Auxiliary 2	A comprehensive report detailing shortcomings found during the testing which will form a basis for the ICT security governance model	Real-world data	Case studies	Security audit report

appropriate steps. Indeed, it was a challenge to determine the objectives before setting out to devise a plan and its timeline. The plan–execution approach was chosen even though some authors start with research and then formulate a plan which sees them ending up with large volumes of unstructured data out of which patterns need to be extracted ex-post. While the outcome may be identical, a substantial risk exists the observations will not cover everything the research was supposed to address, leaving little choice but to repeat it again to obtain the missing data. Conversely, when a definitive plan is outlined ex-ante, such a risk is mitigated.

Abstraction transforms complex phenomena by omitting parts deemed inessential, and then works only with the reduced version. It opens ways to apply insights from one subject to another which possess some common features but differ in others. In the thesis, abstraction has its place in questionnaires and case studies which both work with primary data. In the questionnaire research, results from a representative sample will be generalized and applied to the whole population, a process which involves probabilistic reasoning, i.e., it cannot be reasonably assumed the two groups will exhibit identical features in all instances. At best, statements such as: “We can conclude the population tends to favor... with arbitrary probability,” can be constructed. In the case studies, by abstracting from specifics in an organizational environment, corollaries uncovered during the execution phase will be ported over to the best practices. Irrespective of the organizational type, the findings are to be general enough so that any can be deployed with

minimal modifications, aiming for the common ICT infrastructure denominators. One pitfall of abstraction lies in incorrectly stripping essential system properties; in such a case, the product no longer faithfully represents the original and any conclusions suffer from bias.

Analogy considers properties of a physical object or event as fittingly explaining properties of another with the two not necessarily being in any way related or sharing common features. Applied consistently, it allows for easy explanation of abstract ideas by choosing known things as sources. To demonstrate: in chapter 2.4.8, utility from increasing expenditures to security was likened to the law of diminishing marginal utility observed in microeconomics. The two scenarios are similar, and the method is thus a potent tool for explaining technical principles to the audience not familiar with security. Criticism could be leveled against analogy, especially when little overlap exists between the two elements, but in conjunction with other methods such as abstraction, it could make some parts of the thesis more accessible.

Historical method is prevalent throughout chapter 2 and aims to gain insights into, understanding of, and extrapolate from events in the past. By analyzing secondary literature sources and combining them logically, the method enables achieving the thesis' main and auxiliary goals. To demonstrate: chapter 2.1.1 introduced cybernetics which was found to be overlapping with security, discussed in chapter 2.1.5. Both will be used in the model to argue each system component adds a vector of compromise to the overall security assessment if not adequately protected. This implicitly hints ICT infrastructure should comprise the smallest number of sub-systems which grants it the ability to perform in a reliable, predictable fashion without imposing needless constraints, akin to the principle of least privilege mentioned in chapter 2.2.1. Historical method is a way to combine data into units which are then further processed. It presupposes event continuity to be applicable, though: when the past cannot explain what happens in the present or may happen in the future because of changes which render the historical perspective unusable, alternative methods must be explored instead.

Observation refers to a technique by which intangible signals are recorded, stored, and interpreted by an impartial entity with a subjective set of preferences, viewpoints, and cognitive² distortions. To compensate for the propensity of human mind to abstract and selectively disregard details not conforming to the preconceived notions, measurements utilizing objective scientific instruments were created to ensure research reproducibility. Care must be also taken that the observer does not alter the system under investigation which would change the data. Along with abstraction, case studies will benefit from observations, although controlled conditions cannot be realistically assumed and experiments will have to be performed with some variables left unfixed. This creates a set of circumstances unique for every field run which precludes complete reproducibility but increases data validity because of real-world predicaments under which it was collected. Questionnaire research uses observations indirectly via the answers which themselves introduce bias as questions are perceived differently by each respondent.

Questionnaire is an instrument by which opinions are sought from respondents on a mass scale to be processed, aggregate patterns extracted, and general statements produced which should apply to the whole population with arbitrary probability. Broad distribution, minimal cost, and standardized questions which allow efficient analysis are the advantages while low return rates and subjectivity rank among the downsides. Furthermore, the options out of which the respondent chooses may not exhaustively cover all possibilities, especially when written input (open answer) is not permitted; determining sample size for fixed effect size needs to be done beforehand to estimate experiment parameters in order to reach the desired level for

²[kagnəʃɪv], [*usually before n.*]: connected with mental processes of understanding (Oxford University Press, 2011).

the desired statistical power. Otherwise, the results would not have scaled reliably to the population. Despite the disadvantages, the method is frequently employed in social sciences. In the thesis, questionnaires will supply data on user habits pertaining to BYOD and ICT as well as their knowledge of terms and principles commonly encountered there. An informal tone was deliberately set because the author opines popularity of the method led to desensitization: presented with a questionnaire, people are assumed to take little time and care thinking about the answers, decreasing data validity. Effort was therefore made to augment the experience with elements of dialogue. Chapter 4 is dedicated to the questionnaire survey results.

Case study refers to a descriptive method which helps to answer “how?” and “why?” research questions, does not require control of behavioral events, and focuses on contemporary events (Yin, 2008). The second condition, namely that some variables are free to change, was a strong argument for including it in the thesis as real-world situations involving human factor rarely conform to theoretical, idealized probability distributions or mathematical descriptions. This was cited in chapter 2.1.1 as one reason for establishing higher-order cybernetics which broadened its original scope. Case studies delineate the problem-solving process from initial assessment, experimental conditions, hypotheses, selecting proper tools and tests, obtaining data, analyzing them, and evaluating the hypothesis formulated at the beginning. Some case studies engage the reader by establishing a narrative and asking thought-provoking questions which can be addressed in many ways with one or several answers suggested. Generalizing from case studies is a controversial topic: from a statistical point of view, making assumptions about a population based on a single observation is tied with a high error rate. However, when the case study deals with a common issue, its findings are easily testable on other instances which ensures reproducibility. Moreover, as long as the research conforms to rigorous scientific standards, the results should not be disregarded. Case studies may incorporate a combination of quantitative and qualitative methods. Chapter 5 is dedicated to the results.

3.3 Topic Selection Rationale

The topic was selected to coincide with the challenges organizations currently face with BYOD and ICT security. The decision-making process is detailed below. Each situation (a premise) was assigned one or more problems together with mitigation strategies (responses). A preference scale was subsequently constructed as a measure to assign relative importance of each response, and to categorize prerequisites for main/auxiliary goals outlined above. While it could be argued alternative countermeasures would sometimes be more appropriate, the author believes the challenges have been addressed to the best of his abilities based on extensive secondary sources literature review together with his practical experience gained from analyzing ICT infrastructures and conducting penetration tests against them.

Situation 1: Organizations want to protect sensitive electronic assets.

- Challenge 1-1: Little insight into ICT security on part of users (hypothesis 1).
- Response 1-1: Profiles, security audits.
- Challenge 1-2: Integrating BYOD for work-related tasks from personal mobile devices.
- Response 1-2: Best practices, demonstrations, real-world training.

Situation 2: Adversaries (insiders, outsiders) want to access sensitive assets in an unauthorized fashion.

- Challenge 2-1: Exploitable vulnerabilities in perimeter/internal ICT infrastructure elements.
- Response 2-1: Security policies, penetration tests.

- Response 2-2: Patch management.

Situation 3: Users access sensitive assets using their mobile devices.

- Challenge 3-1: Piggybacking to internal networks.
- Response 3-1: Separating work and personal space, profiles, security audits.
- Challenge 3-2: Closing attack vectors by timely patch deployment.
- Response 3-2: Patch management defined in profiles.

Situation 4: Complex ICT infrastructure management.

- Challenge 4-1: Multiple potentially unknown attack vectors.
- Response 4-1: Best practices mitigating vulnerabilities (defense in depth), security audits.
- Challenge 4-2: Delays between patch releases and production environment integration.
- Response 4-2: Infrastructure hardening, advanced tweaking to reduce the attack surface.

Situation 5: ICT infrastructure comprises human factor (employees).

- Challenge 5-1: Ingrained behavior patterns.
- Response 5-1: Repeated demonstrations and real-world training over long term.
- Challenge 5-2: Social engineering, breach of trust.
- Response 5-2: Best practices, long-term training, suspicion-based behavior toward third parties.

Demonstrations and real-world training are beyond the scope of the thesis. They both draw from psychology, defined as “. . . the study of the mind and behavior. The discipline embraces all aspects of human experience – from the functions of the brain to the actions of nations, from child development to care for the aged. In every conceivable setting from scientific research centers to mental healthcare services, ‘the understanding of behavior’ is the enterprise of psychologists” (APA, 2013). Importance of systematic employee training plans cannot be overstated in addition to the technology-based defense layers, but it will not be discussed further.

ICT security landscape provides many avenues to explore, particularly application of practically-proven theoretical findings to organizations, and enhancement of security policies by combining proactive and reactive measures. Security audits could uncover, analyze, and document non-obvious exploitable attack vectors, and suggest ways to mitigate them. As mentioned in chapter 2.4.8, investments in hardware and software may not result in observable increase of security because new components introduce vulnerabilities from increased system complexity. Long-term employee training could be a better alternative. An interplay of best practices, policies, education, and security auditing is therefore crucial to harden ICT from electronic asset misappropriation.

4 QUESTIONNAIRE RESEARCH

Questionnaires are unique because they are not predominantly focused on qualitative or quantitative research aspects but integrate both by harnessing their strengths for maximal effectiveness. Qualitative research is “. . . a situated activity that locates the observer in the world. It consists of a set of interpretive, material practices that makes the world visible. These practices transform the world. They turn the world into a series of representations, including field notes, interviews, conversations, photographs, recordings, and memos to the self. At this level, qualitative research involves an interpretive, naturalistic approach to the world. This means that qualitative researches study things in their natural settings, attempting to make sense of, or to interpret, phenomena in terms of the meanings people bring to them” (Denzin & Lincoln, 2005, p. 3). Qualitative researchers “. . . are interested in understanding the meaning people have constructed, that is, how people make sense of their world and the experiences they have in the world” (Merriam, 2009, p. 13). Conversely, Creswell (2002, p. 21) defines quantitative research as “. . . one in which the investigator primarily uses positivist claims for developing knowledge (i.e., cause and effect thinking, reduction to specific variables and hypotheses and questions, use of measurement and observation, and the test of theories), employs strategies of inquiry such as experiments and surveys, and collects data on predetermined instruments that yield statistical data.” A combination of qualitative and quantitative approaches is a mixed research, “. . . one in which the researcher tends to base knowledge claims on pragmatic grounds (e.g., consequence-oriented, problem-centered, and pluralistic). It employs strategies of inquiry that involve collecting data either simultaneously or sequentially to best understand research problem. The data collection also involves gathering both numeric information (e.g., on instruments) as well as text information (e.g., on interviews) so that the final database represents both quantitative and qualitative information” (Creswell, 2002, p. 21).

Despite being classified in the quantitative category, the thesis will consider and utilize questionnaires as a mixed method which itself spawned some discussion as to the preferred definition: Johnson, Onwuegbuzie, and Turner (2007, p. 19) analyzed 19 mixed methods and offered the following delimitation: “Mixed methods research is an intellectual and practical synthesis based on qualitative and quantitative research; it is the third methodological or research paradigm (along with qualitative and quantitative research). It recognizes the importance of traditional quantitative and qualitative research but also offers a powerful third paradigm choice that often will provide the most informative, complete, balanced, and useful research results. . . . This type of research should be used when the nexus of contingencies in a situation, in relation to one’s research question(s), suggests that mixed methods research is likely to provide superior research findings and outcomes.”

Unlike purely quantitative methods, questionnaires should account for bias caused by subjective wording and researcher’s bias using neutral, objective, and unequivocal constructions. Also, the respondents should be given an option to express their own opinion if none of the answers fully express their sentiment.

Assumptions cannot be made in advance about what tests will be used for analyzing the responses, but parametric statistical procedures will not be supplanted by non-parametric alternatives if possible. This is because asymptotic transformation under central limit theorem allows to convert unknown probability distributions of a variable to Gaussian distribution provided some conditions, e.g., sufficient sample size, are met. Especially with small samples, the procedure generates non-negligible errors which decrease to zero as number of observations increase. So, if error is indeed generated, it will be treated as sufficiently small, not influencing the results. Questionnaire template can be found in Appendix A.

4.1 Background

Questionnaire research will answer the following question: **What habits and knowledge pertaining to ICT and mobile security can be observed in a representative sample of respondents, and what are the main aspects in need of improvement?** It is predicted that users will strongly prefer comfort and ease of use, and fundamental recommendations, e.g., periodic password rotation policies and protecting mobile devices, will be sidelined. The results will serve as a basis for best practices in the proposed ICT model, presented in chapter 6, specifically for user-side security improvements.

Data for the questionnaire research was sourced from Master degree students at the Faculty of Management and Economics, Tomas Bata University in Zlin in full-time and distance forms. Each student was instructed to distribute and return 10 copies in paper form. Questionnaires were handed back during October 2013–December 2013. The template was provided in the Czech language only as no foreign participants were included. The PDF questionnaire was placed in a virtual course on the Moodle e-learning portal. A total of 784 copies was returned.

To produce the layout, a L^AT_EX class style from the SDAPS framework¹ was used. While the package covers the entire questionnaire research (QR) process from design to scanning, optical character recognition and report generation, the thesis opted for manual data input rather than automated methods due to manageable sample size. Should the QR comprise more than a thousand respondents, the SDAPS would have been utilized in full. Optical character recognition (OCR) “. . . is used to translate human-readable characters to machine-readable codes. Although the characters to be recognized may be printed or handwritten, most applications deal with the conversion of machine-print to computer readable codes. . . The prime purpose for such a process is to provide a fast, automatic input of documents (or forms) to computers either for storage, or for further processing” (Cash & Hatamian, 1987, p. 1). Although the technique would reduce data processing to input validation and corrections, the questionnaire contained multiple form fields where written Czech language input was exclusive. Handwriting recognition coupled with specifics of the language (accute accents, carons) would require extensive work, and it was decided the data would be inputted by hand.

The data was inputted into a Microsoft Excel 2007 spreadsheet and saved with an .xls extension to preserve backward compatibility. The convention used for variable coding is based on two main patterns found in the questionnaire answers: straight and layered, depicted in Figure 26.

Missing values were denoted as “999.” The sole question (Q) where the code may have caused confusion was Q2.11 where the respondent was asked for a number; inspecting the answers beforehand shown no such answer was entered. Uniform missing value tracking system helped with specifying the code in IBM SPSS Statistics which treated “999” in any field as a symbol rather than a figure. Otherwise, the results would be skewed and not justifiable because the majority belonged to limited code ranges, e.g., 1–5, 1–7. Imputation of missing values was not attempted despite the mode, median, or mean being suitable; alternatively, the least-occurring value may have been supplied instead of the mode. Missing value analysis is outside the thesis’ scope and will not be discussed further. Some questions were designed to give the respondent opportunity for open-ended answers, i.e., half-open questions. In case the the option had been picked, the text was transferred over to the Microsoft Excel spreadsheet verbatim.

The questionnaire structure consists of four parts: general IT overview, mobile phones, additional questions, and personal information. The first category, general overview of IT,

¹Scripts for data acquisition with paper-based surveys: <http://sdaps.org/>

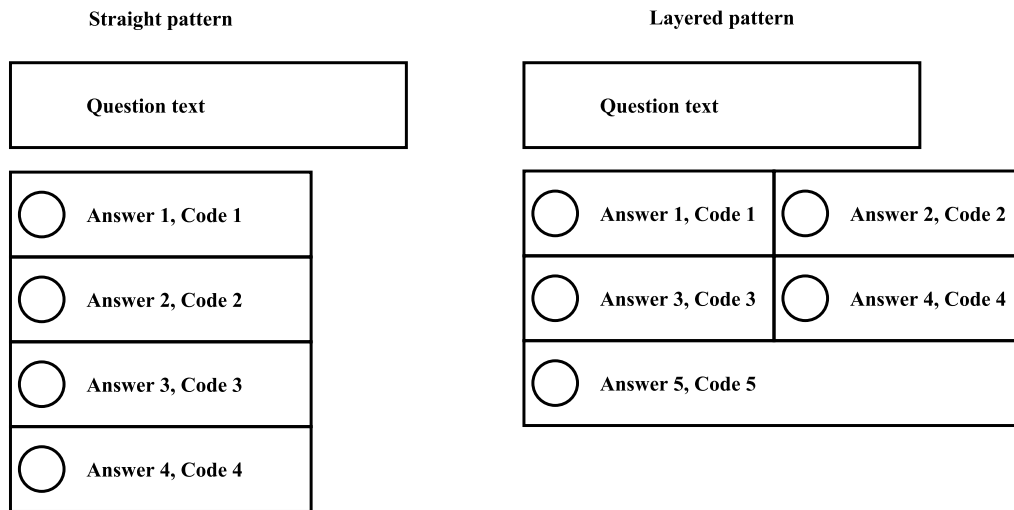


Fig. 26: Questionnaire answer patterns. For the straight pattern, the answers were coded sequentially from the top, for the layered pattern, the Latin writing system (left→right, up→down) was used as a baseline.

Source: own work.

comprises 11 questions polling the respondent about information pertaining to passwords and software, interspersed with inquiries about how subjectively knowledgeable they rate themselves. Specifically, Q1.5 was included to check whether the computer literacy level specified in question Q1.1 was inflated: if answer code 3–6 (highly skilled, guru, IT guy) was selected and the purpose of HyperText Transfer Protocol Secure (HTTPS) stated incorrectly, it can be assumed the user is either highly specialized in other aspects of ICT, or the computer proficiency was overstated. Relation between Q1.1 and Q1.5 may hint at one (both) questions interpreted subjectively, or systematic overconfidence in the sample. This could imply people cannot be trusted when making security-related decisions without having a baseline against which they are compared.

Each option in Q1.1 contains one or more keywords which should help decide on the answer quickly: Facebook; Word, Excel; proxy, VPN; Linux, Bash; and binary code. It was hypothesized the respondent would unconsciously seek known concepts, and the keywords directed them to quickly determine which computer literacy level would be suitable. Supposing the user fluently operates Microsoft Word and Excel, spotting the two words in the text body should give a clue the “moderately skilled” option is the correct one. It was further hypothesized a moderately-skilled person can also use web browser, implicitly assuming “beginner” competencies are present, too. The bottom-up (all skills up to and including the preferred answer) contrasts with the top-down down approach: rather than analyzing what they can do, the respondent may start by excluding skills above their level until they arrive at those which accurately reflect the skill set they actually possess. Regardless of the mental process employed, the answer establishes groundwork for Q1.5.

Questions 1.2–1.4 are aimed to identify respondent’s browser, when was the last time it was updated, and the operating system, respectively. Q1.2 recognized some users’ unfamiliarity with names but their focus on distinctive features, namely program icons used to access the Internet. Therefore, brackets provided description of the icons so that an image was invoked in user’s mind. Some browsers stream updates, forgoing the need to ask for a permission from the user. The trend was acknowledged and a respective option listed as the very first. Otherwise, intervals denoting approximate time since the last prompt or notification that the browser had been updated, were included. Lastly, under the assumption inexperienced users cannot remember even approximate

date, answer 7 referenced a popular science-fiction film saga, Star Wars, using the opening text, “A long time ago in a galaxy far, far away. . .” to denote the software was not updated for more than a year. Q1.4 asked about the operating system but visual cues were not described. To exhaustively cover all possibilities, form fields for a Linux distribution, an alternative system, or a combination (dual-boot, triple-boot) were added. As mentioned previously, Q1.5 served as a check for whether the proficiency stated in Q1.1 can be considered inflated or representative of the user’s ICT knowledge.

Questions 1.6–1.9 probes for password composition and rotation practices. Q1.6 and Q1.7 follow the establish→expand principle seen in Q1.2 and Q1.3, first setting the background (password) before asking about the frequency of change. The principle was chosen so that the same thought approach can be employed as previously, reducing time per answer. Q1.6 excluded electronic banking authentication strings as the most important password which the respondent was asked for. This is because Czech banks still predominantly use a system where the client is assigned a short numeric sequence without the possibility of changing it afterwards. This would make answering Q1.7 impossible, and so the option was explicitly disallowed. Q1.7 is comparable to Q1.3 because the intervals are identical except for answer 2 which indicates the user does not remember. A form field for specifying time not conforming to any option was included. Q1.8 then drilled down and polled for the amount of characters from alphabet sets (lowercase, uppercase, symbols, and spaces) while purposefully omitting numbers. After much consideration, a compromise was made between user’s willingness to answer and data representativeness. While password composition metadata is not classified personally-identifiable information by itself, combined with an (optional) email address and a name, legitimate concerns may arise as to whether the data can be maliciously used for profiling purposes. Therefore, numbers were not included in line with the assumption they constitute a portion of the password on which its security critically depends, and without which the string cannot be reverse engineered. This introduced ambiguity because some respondents summed numerical characters together with symbols while some created a separate entry, and data for Q1.8 must be assumed polluted at least for the “symbols” portion. Q1.9 then attempted to quantify price per password using a financial incentive (100 CZK; 5 USD using a fixed 20 CZK/USD conversion rate) offered in exchange for the password the recipient was asked to work with in the four-question block. An alternative was to specify an arbitrary realistic amount, although this proved in counter-intuitive in hindsight as many patently inflated figures were recorded. Hence, the financial data will not be included in the analysis, and Q1.9 will be understood as a dichotomous (yes/no) question.

Questions 1.10 and 1.11 finalized the password section and examined password generation and storing practices. Q1.10 recounted the most popular composition rules identified in chapter 2.4.3, all of which are integrated in password-cracking software and susceptible to reverse engineering. Two form fields were provided, addressing the tendency to rely on third-party local software or remote services for generating random strings, and for alternative means of password creation. Q1.11 uncovered two pieces of information if either of the first two answers was selected: handling and reuse of passwords on sites. Reuse being a widespread phenomenon, both options explicitly mentioned the practice along with another substandard habit, password memorization. Even though the approach does away with a single point of failure, the encrypted database for storing the sensitive data, it is hypothesized complexity is negatively correlated with length and memorability. Entropy, discussed in chapter 6.2.4, a general measure of complexity, would likely reflect passwords are made conducive for later recall with lower values which indicate proneness to reverse engineering techniques, e.g., mutators (chapter 2.4.3). A form field allowed to mention a dedicated program encrypting the strings which strongly suggests it is also used for credentials management.

The second category, mobile phones, comprises 11 questions polling about pervasiveness of mobile technologies in financial management and work-related activities. Throughout, smartphones and tablets are treated equally, and the recipient does not have to specify which device they have in mind, leading to ambiguity as the two device classes may differ. Q2.1–Q2.4 targeted preference for a particular operating system and usability. Q2.1 is a dichotomous inquiry about possession of a smartphone or a tablet; however, a hypothetical scenario is supplied in case of negative response, stating further questions should be answered under the pretense the respondent owns one. The data is still relevant because it classifies security practices regardless of physical hardware ownership. Q2.2 listed popular mobile operating systems at the time of writing, omitting BlackBerry. It was assumed should the need arise to include an alternative system, the form field will be used. A combination of multiple systems on one device is rarely seen and was not included. Negative correlation between Q2.1 and Q2.2 may hint at dissatisfaction with the current OS: if an individual purchased a smartphone (Q2.1=1), it can be assumed some research was conducted prior to the decision and a favorite was selected out of the existing variants. Should Q2.2=1, their preference may have shifted during day-to-day use, and the system is no longer perceived to be the right choice. Q2.3 extended Q2.2 with one or more reasons for inclination toward the particular OS, supposing Q2.2≠1; otherwise, Q2.3 was likely skipped. A form field complemented the choice pool because exhaustively encompassing all possibilities was not the objective. Multiple choices may be common, denoting a blend of considerations. Q2.4 presented typical use cases. Correlation between the breadth of utilized functions and the operating system may uncover significant differences, and the adversary could exploit the intelligence by prioritizing a particular OS due to its wider attack surface. A form field provided space to list additional actions, although they were seldom used.

Questions 2.5–2.8 moved on to security aspects of smart mobile devices. Q2.5 pertained to online banking and the respondent was queried about their preference to rather use smartphones over desktop computers, implicitly expecting such actions are performed. A form field allowed for numeric input denoting the frequency per month with which mobile phones facilitate access to sensitive data. Electronic banking is considered a high-risk activity and if conducted over unsecured channels without encryption (HTTPS, Q1.5), may result in passive or active data interception where the adversary listens or actively modifies the messages passed between the client and server. It is therefore crucial to validate the entire communication chain for signs of tampering by malicious parties which requires technical sophistication beyond what the average user can be expected to possess. Q2.6 polled for the span of functions mobile devices have compared with personal computers: while missing or additional features are not mentioned, the purpose was to discern whether users overestimate or underestimate under the assumption both classes are equal in features. Q2.7 returned back to password management, specifically Q1.11, and extended it with storing credentials on mobile phones. Substandard practices such as writing passwords on sticky notes attached to a monitor may have been replaced by typing the string in a reminder unencrypted, or using a note-taking software. It was posited the majority of users prefer comfort over security, and at best attempt to conceal their password. Thus, Q2.8 further probed for security measures deployed on the device, particularly lock screen passcodes because their complexity, length, and uniqueness (chapter 6.2.4) are the features preventing the adversary from accessing the phone and expropriating sensitive personal data off it for later analysis. The combination of unencrypted credentials and no passcode constitutes a potent attack vector exploitable to hijack individual's electronic identity, giving the adversary foothold into organization's internal network should the records contain system domain logins.

Questions 2.9–2.11 finalized the section. Q2.9 polled the recipient about an element of BYOD management discussed in chapter 2.3. One way how IT personnel can unify diverse hardware and software base is through profiles, collections of permissions and restrictions installed onto

the device and enforced whenever it interacts with sensitive electronic assets. Nevertheless, as the user is the legitimate owner, it is up to their discretion whether the profile will be permitted to run. ICT policies can restrict the smartphones not enrolled in the BYOD management program from accessing internal networks, a measure which hampers productivity for remote workers and decreases convenience when checking emails and other actions. Attitude of respondents toward profiles is an important indicator of the willingness to give consent for such action. Although it was expected the perspectives would differ, denial may have been the result of lacking information about the benefits and disadvantages of profiles as much as fundamental opposition to third-party control. Q2.10 utilized Likert scale for gauging subjective importance of security, price, functions/applications, look, and brand in smartphones. The scale was originally devised as an attempt “. . . to find whether social attitudes. . . can be shown to be measurable, and if an affirmative answer is forthcoming, a serious attempt must be made to justify the separation of one attitude from others” (Likert, 1932, p. 9). It was used with answers grouped into 5 classes: strongly approve, approve, undecided, disapprove, and strongly disapprove. Despite the argued propensity of some respondents to choose the middle option when in doubt, the scale format was preserved unmodified. Coding convention was established as follows:

- strongly approve: 1,
- approve: 2,
- undecided: 3,
- disapprove: 4,
- strongly disapprove 5.

No prior assumptions have been made regarding the answers except for one: security would not likely be the preferred choice. Q2.11 was the only inquiry strictly requiring written input: the participants were asked to put forward a figure for newly-discovered vulnerabilities in 2012 as reported by Symantec in its 2013 Internet Security Threat Report, Volume 18². The question statement clearly indicated the answer is an estimate the purpose of which is to analyze whether users systematically underestimate or overestimate the number of threats. The correct answer, 5291, was hinted at to be in thousands. Regrettably, some respondents patently inflated their answers, others entered figures which may or may not have been meant seriously, and while every effort was made to discern between the two, the source data is with high probability non-representative and skewed upwards. Due to this reason, it was decided Q2.11 will not be considered for further testing.

The third category, additional questions, comprises three entries not suitable for any other part which investigate knowledge of terminology pertaining to and views on electronic crime. Q3.1 started off by priming the respondent toward the subject matter: a real-world scenario was presented where clicking a malicious link or an attachment in an email causes endpoint malware infection, and the user was asked whether they are more observant if such a situation unraveled previously. Personal experience was assumed and no option on the contrary added; while the assumption can be criticized as overly strong, the results indicate the four answers were comprehensive. The question statement did not delve into technical details and kept the example on a general level for accessibility. Q3.2 surveyed on spam and phishing, the latter a type of social engineering campaign (chapter 2.4.5) launched against individuals with the intention of obtaining sensitive data through impersonation of legitimate services, e.g., electronic banking. It is tangentially related to Q1.5 as the attack can be thwarted by HTTPS encryption. While spam, unsolicited electronic distribution of bulk messages, should be relatively well-known, phishing represents an attack vector where websites are cloned with sophistication varying from

²http://www.symantec.com/security_response/publications/threatreport.jsp

low to being indistinguishable save from inspection using advanced tools the average user cannot be expected to employ. Thus, the results may hint at the need for more information, training, and real-world examples. Q3.3 reversed the perpetrator→victim narrative and hypothesized the respondent has the ability to launch malicious campaigns. As chapters 5.1 and 5.2 will demonstrate, this is well within the realm of possibility because the tools are freely available, and the readiness to engage in illegal conduct is postulated to be primarily governed by ethics. Even though moral grounds will not be investigated further, they are contrasted with financial incentives such actions may generate. The form field for alternative answers was used only marginally, suggesting the scope of answers was sufficient for the participants.

The fourth category, personal information, identifies the participant according to selected demographic and socio-economic criteria. Q4.1–Q4.4 did not include any form fields as answers should be exhaustively covered in the options. Gender, age, economic status, and monthly income in CZK were selected while marital status was deemed inessential for the nature of the research. Some participants did not answer Q4.4 but included information about password composition (Q1.8) and selection rationale (Q1.10), evidence security has lower preference than personally-identifiable information. The criteria were moved near the end because it was believed the respondent would be unwilling to continue knowing they imparted identifying information at the beginning. However, after 25 previous inquiries, the time already spent could have been a factor for completing the survey rather than abandoning it. Q4.4 was the only one which a subset of respondents chose to ignore.

Finally, a closing statement was appended on the penultimate page which delineates the chain of events after the questionnaire is returned. Data anonymization, confidentiality, and shredding the physical copies after the research is concluded were stressed to instill a sense of confidence the information, some of which might be considered sensitive, will be used solely for legitimate purposes. The author is of the opinion the participants were entitled to know analyses will be performed over aggregate data despite the full name and signature required to ensure validity. Both can be trivially faked but similar handwriting may hint at several copies filled out by a single individual. Author's contact information were added but no contact was initiated, e.g., to verify the questionnaire's origin. The very last page comprised a form field numbered Q4.5 where criticism, feedback, and suggestions could be optionally inputted. The information was reviewed but was not included in the analysis due to its informative, subjective nature. A frequent complaint was excessive length which the author concedes has merit.

Questionnaire template in Appendix A has been revised for mistakes and grammar omissions based on comments in Q4.5. Otherwise, the content does not differ from the original and is identical to the version each respondent was asked to fill out.

IBM SPSS Statistics 22 64-bit was used as the analytic tool of choice. The data set was first imported from a Microsoft Excel file and saved with .sav extension native to SPSS. While no performance improvements were expected by using the format, it ensured compatibility for features such as labeling of variables, missing values specification, value labels, and others to streamline data processing and interpretation. Hardware and software configuration was as follows:

- OS: Windows 7 Professional Service Pack 1 64-bit,
- CPU: Intel Pentium Dual-Core 2.20GHz,
- RAM: 4GB DDR3 SDRAM.

The setup is thoroughly described in chapter 5.1.1. No component created a bottleneck during testing mainly because resource-intensive operations were not executed, and the data set was

relatively small: the .xls source file size was 315 392 bytes, well within the capabilities of the hardware and software to process efficiently.

SPSS has facilities to export textual and graphical results. PDF was selected for both due to its ability to handle vector images without loss of visual information, i.e., lossy compression. All fonts were embedded in the output files, ensuring the documents were self-contained and independent from the operating system font pool. Tables and graphs were not modified from their original form in any way except for omitting titles and associated log messages. Colors were also left at their default settings to maintain uniformity in appearance. Tables were exported as graphical objects and will therefore be denoted as figures. Lastly, the results will not mention sequences of steps taken to produce the output, sidelining them in favor of interpretation. Literature sources on statistical testing in SPSS contain details and descriptions the thesis does not aim to reproduce.

Note: throughout the questionnaire research, plural addressing “we” instead of singular “I” is preferred. Unless otherwise stated, significance level for statistical hypotheses testing is set to $\alpha = 0.05$.

4.2 Results

The chapter presents main results of the QR both within and across the four categories. Prerequisites for statistical hypotheses testing are mentioned immediately prior to calculation, interpretation is provided immediately following the results. We will mainly focus on descriptive statistics, selected findings will be used as input to case study 1 in chapter 5.1 and the ICT security governance model in chapter 6. This establishes a sequence where the output forms basis for additional research which will contribute to the theoretical framework as well. The discussion aims for brevity and clarity, forgoing most of the details in favor of streamlining the text. Inferences pertaining to security will be constructed which do not utilize formal tests but hint at possibilities, user habits, real-world scenarios, and possible attack vectors. Rooted in author’s experience and extensive literature review from previous chapters, validity of the claims cannot be ascertained and should be understood as conjectures. They nevertheless represent likely developments in an environment where security and user comfort are postulated to be at odds, with strong preference for the latter.

4.2.1 Personal Information, General IT Overview

We will first analyze gender and age structure of the respondents; this will allow us to determine whether the distribution of the sample was skewed toward particular group. Because university students handled the questionnaires, it is reasonable to assume a non-negligible part of the recipients would be in the 19–25 category; moreover, family members were also expected to be given copies to fill out. This means another two groups, namely 36–45 and 46–55, could be strongly represented in the sample as well. Figure 27 lists the results, graphical depictions in Figures 28 and 29 were constructed from the frequency tables.

Three age groups were indeed more frequently included compared to others, namely 19–25 which suggests students filled out one questionnaire themselves, 26–35, and 46–55. While origin of the former can only be hypothesized about (friends, siblings), 46–55 are likely parents and older family members. The results will be primarily indicative of the opinions and practices of the three groups. To further break down the age structure per gender, a contingency table in Figure 30 confirms the 19–25 age group contributes to the results by 40 % for both genders,

		q4.1	q4.2
N	Valid	784	783
	Missing	0	1

q4.1

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Female	422	53,8	53,8	53,8
	Male	362	46,2	46,2	100,0
Total		784	100,0	100,0	

q4.2

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	<18	55	7,0	7,0	7,0
	19-25	318	40,6	40,6	47,6
	26-35	170	21,7	21,7	69,3
	36-45	78	9,9	10,0	79,3
	46-55	118	15,1	15,1	94,4
	56-65	22	2,8	2,8	97,2
	65>	22	2,8	2,8	100,0
	Total	783	99,9	100,0	
Missing	999	1	,1		
Total		784	100,0		

Fig. 27: Age and gender frequency tables. Female respondents prevailed with age structure following a predictable trend, with the 19–25 group being the most frequent. Source: own work.

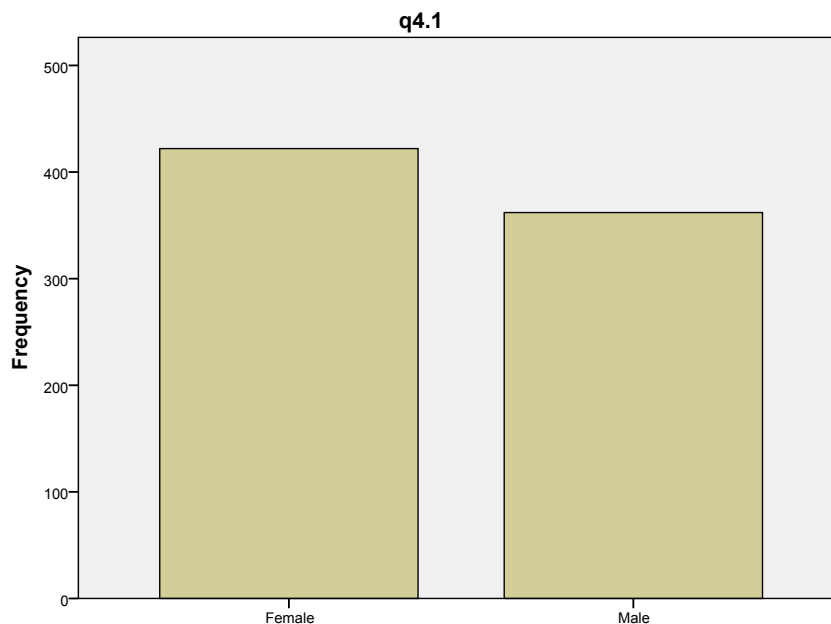


Fig. 28: Gender frequencies bar chart. Female respondents slightly surpassed male even though the disproportion is marginal and should not influence the results. Source: own work.

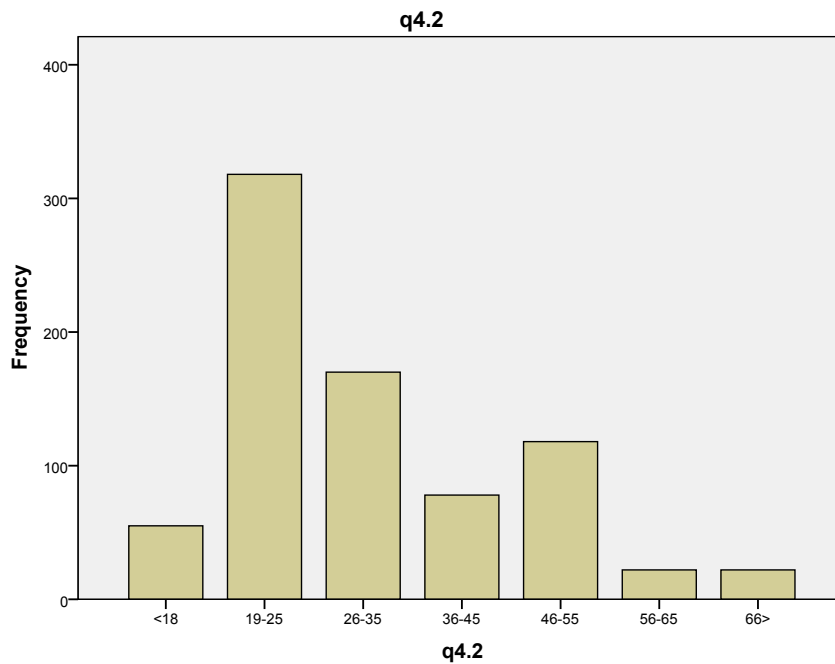


Fig. 29: Age frequencies bar chart. The 19–25 group dominated, indicating students who distributed the questionnaires filled them out themselves and also included their peers.
Source: own work.

with 55–65 and 66+ having the lowest influence in female and male respondents, respectively. Figure 31 presents a graph sourced from the data in the contingency table.

With the exception of 56–65 group, female participants were more numerous. We can calculate Pearson’s chi-squared test which analyzes whether two categorical variables conform to a particular theoretical probability distribution. The four assumptions for the test are simple random sampling, sufficient sample size, minimum expected cell counts, and independence of observations. The prerequisites have been met even though the random sampling criterion can be questioned because the respondents were not selected randomly as documented by the age structure skewed toward the 19–25 group. However, the chi-squared test will report whether the observed and theoretical distributions significantly differ, hinting at at least one factor deforming the data, i.e., random sampling violation. We will utilize p-value for hypotheses testing because SPSS calculates it by default. Pearson’s chi-squared test hypotheses:

- null H_0 : No statistically significant association exists between age structure and gender, i.e., men and women are equally likely to belong to any age category.
- alternative H_1 : Statistically significant association exists between age structure and gender, i.e., men and women are not equally likely to belong to any age category.

Results are depicted in Figure 32. The first table lists observed and expected frequencies; should the two differ substantially in multiple cells, it can be postulated the source data was affected by at least one factor. In our case, the counts do not exhibit large differences, suggesting the two probability distributions are a close match. This is corroborated by asymptotic significance (2-sided) in the second table which specifies p-value for Pearson’s chi-squared test on the first line. As $p\text{-value} > \alpha$, there is only weak evidence against the null hypothesis, and we fail to reject it in favor of the alternative one. Thus, no statistically significant association exists between age structure and gender, and if an additional statistical unit was added to the sample, it could equally likely belong to any age category, and vice versa. The result supports

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
q4.2 * q4.1	783	99,9%	1	0,1%	784	100,0%

q4.2 * q4.1 Crosstabulation

			q4.1		Total
			Female	Male	
q4.2 <18	Count	29	26	55	
	% within q4.1	6,9%	7,2%	7,0%	
	% of Total	3,7%	3,3%	7,0%	
19-25	Count	172	146	318	
	% within q4.1	40,8%	40,4%	40,6%	
	% of Total	22,0%	18,6%	40,6%	
26-35	Count	89	81	170	
	% within q4.1	21,1%	22,4%	21,7%	
	% of Total	11,4%	10,3%	21,7%	
36-45	Count	44	34	78	
	% within q4.1	10,4%	9,4%	10,0%	
	% of Total	5,6%	4,3%	10,0%	
46-55	Count	66	52	118	
	% within q4.1	15,6%	14,4%	15,1%	
	% of Total	8,4%	6,6%	15,1%	
56-65	Count	9	13	22	
	% within q4.1	2,1%	3,6%	2,8%	
	% of Total	1,1%	1,7%	2,8%	
66>	Count	13	9	22	
	% within q4.1	3,1%	2,5%	2,8%	
	% of Total	1,7%	1,1%	2,8%	
Total	Count	422	361	783	
	% within q4.1	100,0%	100,0%	100,0%	
	% of Total	53,9%	46,1%	100,0%	

Fig. 30: Age and gender contingency table. The drill down shows age categories per gender and how they are represented in the sample.

Source: own work.

the prior assumption that violation of the random sampling criterion was not severe enough to cause significant shift in observed frequencies compared to their theoretical counterparts.

Moving from personal characteristics to Q1.1, Figure 33 depicts pie chart for how proficient the respondents classified themselves in ICT. Each option provided examples of technologies the user can operate and configure, as described in chapter 4.1.

Almost half (exactly 49.6 %) of the respondents estimated their skills to be lower intermediate with 84.8 % belonging to either unskilled, basic skills, or lower intermediate categories, 15.2 % classified themselves as upper intermediate, guru, or geek. The finding strongly suggests users cannot discern attack scenarios crafted by moderately-skilled adversaries. Corporate ICT management should reflect on the fact and strongly focus on preventative measures, e.g., training. A sample curriculum will be presented in Table 19 later in the thesis, specifically addressing social engineering which targets the human element of security. The gap between complexity of the technology and the level of general ICT knowledge mentioned in chapter 1 is prominent in the statistical sample covered in the research, similar trend could very probably be observed in

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
q4.2 * q4.1	783	99,9%	1	0,1%	784	100,0%

Bar Chart

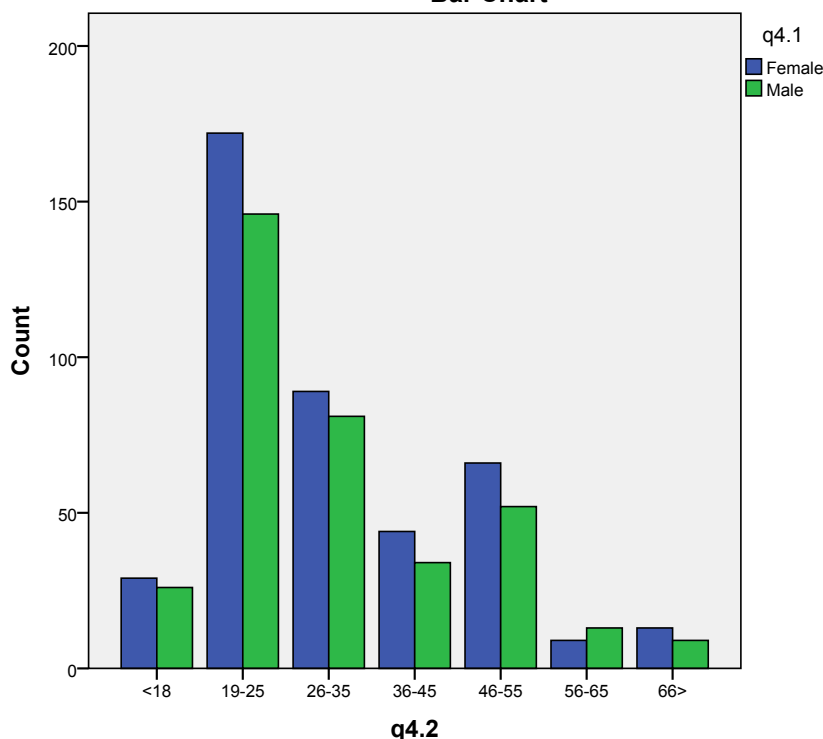


Fig. 31: Age and gender clustered bar chart. A single missing value, denoted “999,” cannot be expected to skew the results substantially.
Source: own work.

the population as well. It can be further argued the trend is reflected in browser selection (Q1.2), whose frequency table is depicted in Figure 34. The table makes browser selection preferences apparent but Figure 35 nevertheless presents a bar chart of descending values to demonstrate the point graphically.

Chrome, Mozilla Firefox, and Internet Explorer were the most popular choices with Opera, Safari, and combination of browsers lagging. Even without analyzing Q1.3 (frequency of browser updates), some inferences can be drawn. Chrome is known for distributing patches automatically, and the software is kept current at all times which increases security and decouples users from the deployment process. The feature is heavily emphasized in the proposed model in chapter 6 because it eliminates delays and closes windows of opportunities for the attacker to penetrate the system. Conversely, Internet Explorer is hardened through Microsoft Update every second Tuesday of the month, although some critical vulnerabilities have been mitigated via out-of-cycle patches. Compared to Chrome, the scheme is inflexible and prone to delays when users deliberately disable or ignore system warnings about new updates being available, especially when granted full permissions on their workstations. In Chrome, disabling automatic updates is possible through a fairly advanced procedure beyond capabilities of the average user. We can safely assume browsers are used with default settings in place most of the time, which means automatic updates are left enabled in Chrome.

For Mozilla Firefox, the conclusion is ambiguous. Prior to version 16, manual update checks

q4.2 * q4.1 Crosstabulation

			q4.1		Total
			Female	Male	
q4.2 <18	Count	29	26	55	
	Expected Count	29,6	25,4	55,0	
19-25	Count	172	146	318	
	Expected Count	171,4	146,6	318,0	
26-35	Count	89	81	170	
	Expected Count	91,6	78,4	170,0	
36-45	Count	44	34	78	
	Expected Count	42,0	36,0	78,0	
46-55	Count	66	52	118	
	Expected Count	63,6	54,4	118,0	
56-65	Count	9	13	22	
	Expected Count	11,9	10,1	22,0	
66>	Count	13	9	22	
	Expected Count	11,9	10,1	22,0	
Total	Count	422	361	783	
	Expected Count	422,0	361,0	783,0	

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,325 ^a	6	,887
Likelihood Ratio	2,325	6	,888
Linear-by-Linear Association	,029	1	,865
N of Valid Cases	783		

a. 0 cells (0,0%) have expected count less than 5. The minimum expected count is 10,14.

Fig. 32: Age and gender Pearson’s chi-squared test. The footnote specifies no cell count is lower than the threshold value of 5; if such situation occurred, Monte Carlo simulation or complete enumeration would have to be used instead.

Source: own work.

were necessary which left the browser open to attacks if the action was not performed periodically. In newer iterations, the procedure is identical to Chrome and no user input is required. The behavior can be changed from within the browser but at default settings, automatic updates are turned on and likely remain unchanged for the majority of users. In Q1.3, 115 respondents answered their Mozilla Firefox browser is kept current by itself; the group runs version 16+. Figure 36 lists all categories in a contingency table.

Respondents who answered “last month” need not be necessarily running older versions of Mozilla Firefox; during October and November 2013 when the questionnaires were being answered, the browser indeed received two updates. Regardless of whether users patched manually or were notified of the version change, they have been running the newest version with automatic updates turned on since then. The two answers are thus equivalent, bringing the total number to 156. The same could be asserted for the rest of the answers because Mozilla Firefox 16 was released in October 2012, i.e., every browser updated since then has automatic updates turned on under the assumption default settings were not changed; this sums to 180. The remaining two answers can be legitimate, or selected to acknowledge the popular culture reference described in chapter 4.1. If the former, any vulnerability mitigated in later versions of Mozilla Firefox can be exploited to run unsanctioned code, effectively taking control of the victim’s machine and using it for arbitrary malicious purposes.

q1.1		
N	Valid	784
	Missing	0

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Unskilled	29	3,7	3,7	3,7
	Basic skills	247	31,5	31,5	35,2
	Lower intermediate	389	49,6	49,6	84,8
	Upper intermediate	83	10,6	10,6	95,4
	Guru	27	3,4	3,4	98,9
	Geek	9	1,1	1,1	100,0
	Total	784	100,0	100,0	

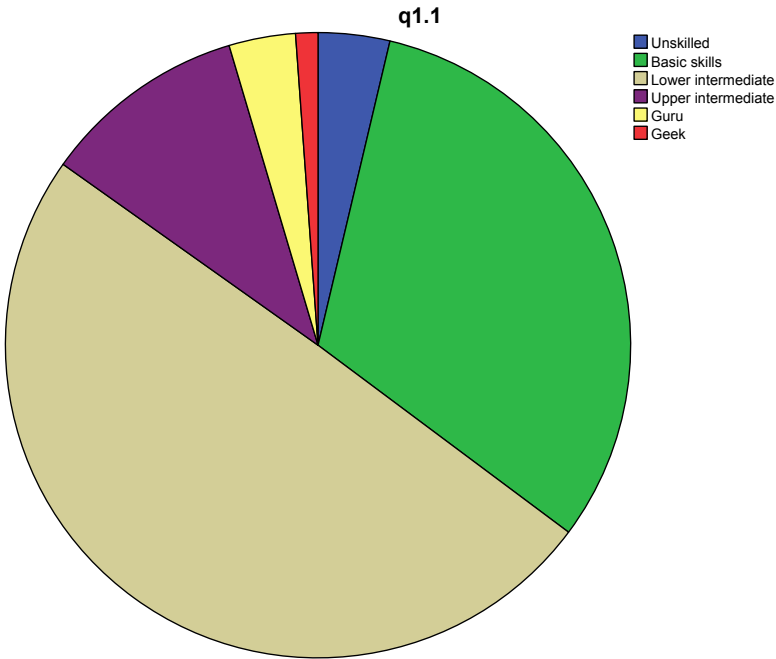


Fig. 33: IT proficiency classification of respondents. The “Valid Percent” column in the frequency table denotes the number of observations without missing values, which were not encountered in this case. The value is therefore identical to the “Percent” column.
 Source: own work.

Internet Explorer’s patch deployment model hinders security because it relies on a fixed window and is not tailored to proactively address emerging threats. Moreover, turning off or disregarding automatic updates entirely is detrimental for withstanding novel exploits, especially when running Microsoft Windows, almost an exclusive choice as documented by Figure 37. Such a result could be expected due to the system’s strong position in the consumer market, popularity, and prevalence in desktop stations and notebooks offered by major hardware vendors. Figure 38 demonstrates the distribution graphically.

Users strongly prefer Microsoft Windows, Mac OS X and Linux are represented marginally. It can be hypothesized the popularity of Mac OS X partially stems from an interplay between mobile devices (iPad, iPhone) and the operating system from the same vendor which some users prefer for the tight integration of hardware and software as well as the design philosophy. By itself, Linux has a negligible share; if at all deployed, users opt for a dual-boot system running Microsoft Windows and Linux in parallel. This may indicate either a conscious choice or a form of vendor lock-in mentioned in chapter 2.2.3, for instance software compatible only with a particular operating system which is run solely for the purpose or accessing it, although virtualization may be a more plausible alternative. Dual-boot systems can be mainly expected on

Statistics

q1.2

N	Valid	778
	Missing	6

q1.2

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Internet Explorer	193	24,6	24,8	24,8
	IE+Mozilla	4	,5	,5	25,3
	IE+Mozilla+Chrome	1	,1	,1	25,4
	IE+Chrome	6	,8	,8	26,2
	IE+Opera	2	,3	,3	26,5
	Mozilla Firefox	218	27,8	28,0	54,5
	Mozilla+Chrome	5	,6	,6	55,1
	Mozilla+Chrome+Opera	2	,3	,3	55,4
	Mozilla+Opera	2	,3	,3	55,7
	Chrome	272	34,7	35,0	90,6
	Chrome+Safari	1	,1	,1	90,7
	Chrome+Opera	3	,4	,4	91,1
	Safari	15	1,9	1,9	93,1
	Opera	49	6,3	6,3	99,4
	Maxthon	5	,6	,6	100,0
	Total	778	99,2	100,0	
	Missing	999	6	,8	
Total		784	100,0		

Fig. 34: Browser selection frequency table. Combining multiple browsers for regular use is marginal in the sample of respondents.
Source: own work.

machines of advanced users.

The correct answer to Q1.5 is “HTTPS says the webpage may be protected” and was supposed to gauge how the respondent understands online security and attacks invalidating HTTPS protection. Table 39 depicts the results.

The difference between answers two and three is that the former disregards any known attacks against HTTPS, and treats it as inherently secure. However, if channels prone to active eavesdropping are utilized to access the Internet, the attacker can hijack the traffic and supplant a fake website front-end. When the user inputs their login credentials, they are intercepted and immediately reused on the genuine page to impersonate the victim. Instead of implicitly trusting HTTPS and accepting the site as secure when the padlock icon is visible, users should be advised to carefully inspect the digital certificate and the site itself, and never use channels such as unprotected Wi-Fi networks for sensitive operations. The answers seem to indicate respondents included in the sample are likely to believe the website is secure when an HTTPS notification and the padlock icon are displayed in the browser window or address bar, both of which can be spoofed when the perpetrator controls the packet flow. Countermeasures exist which Chrome, Mozilla Firefox, and Opera implemented; Internet Explorer and Safari do not support the features as of January 2014 and remain vulnerable. Users are advised to select their browser based on whether security is enforced at default settings which will most likely be used without changes.

Questions Q1.6–Q1.11 form basis for a case study 1 in chapter 5.1. They map password strength, composition, and selection rationale for increased success during reverse engineering, a process which attempts to get human-readable string from hash-obfuscated sequences. Details

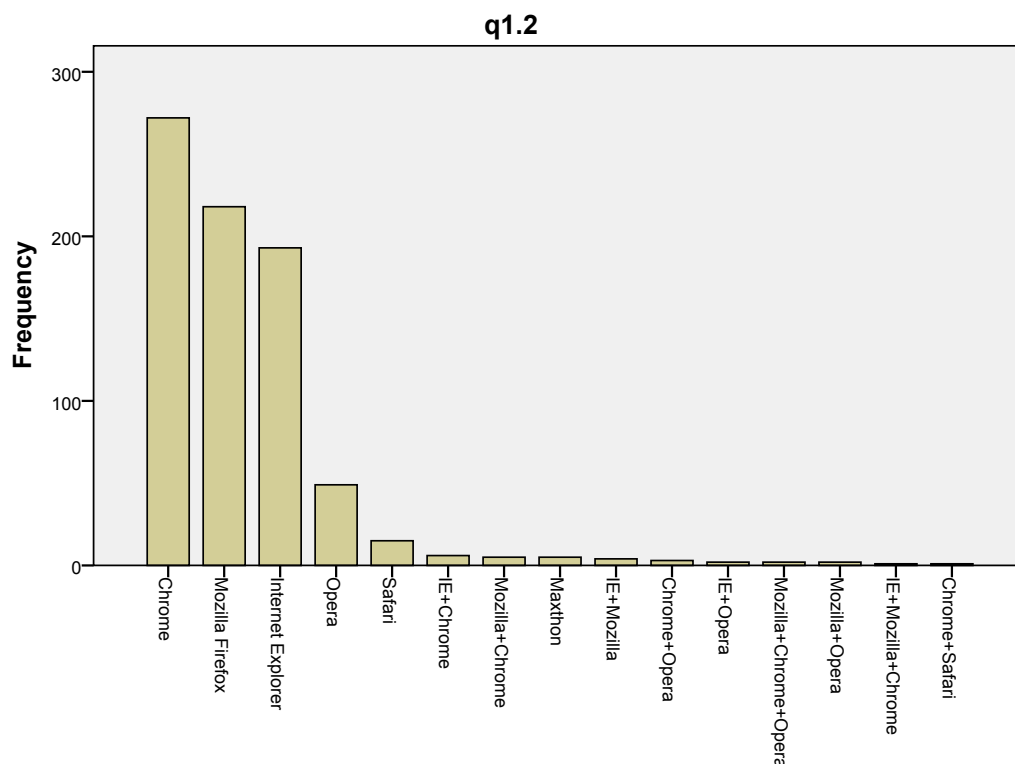


Fig. 35: Browser selection bar chart. Chrome, Mozilla Firefox, and Internet Explorer are preferred for accessing the Internet, with Chrome having the largest user base.
Source: own work.

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
q1.2 * q1.3	777	99,1%	7	0,9%	784	100,0%

q1.2 * q1.3 Crosstabulation

Count		q1.3							Total
		Automatic	Don't know	Last month	Last 3 months	Last 6 months	A year	Longer than a year	
q1.2	Internet Explorer	85	65	17	5	13	2	6	193
	IE+Mozilla	2	1	0	1	0	0	0	4
	IE+Mozilla+Chrome	0	1	0	0	0	0	0	1
	IE+Chrome	4	0	1	0	0	0	1	6
	IE+Opera	2	0	0	0	0	0	0	2
	Mozilla Firefox	115	36	41	10	11	3	2	218
	Mozilla+Chrome	2	0	1	0	1	1	0	5
	Mozilla+Chrome+Opera	1	0	1	0	0	0	0	2
	Mozilla+Opera	0	1	1	0	0	0	0	2
	Chrome	172	36	26	15	18	0	4	271
	Chrome+Safari	1	0	0	0	0	0	0	1
	Chrome+Opera	2	0	0	0	1	0	0	3
	Safari	11	0	2	1	0	1	0	15
	Opera	22	7	14	4	2	0	0	49
	Maxthon	1	1	2	0	1	0	0	5
Total		420	148	106	36	47	7	13	777

Fig. 36: Browser update frequency contingency table. Seven missing values were registered which could be explained by lack of technical knowledge to answer despite the “I don’t know” option.
Source: own work.

will be described later but the answers are expected to supply evidence for choosing viable vectors of approach. Password length breakdown is quantified in Figure 40, Figure 41 depicts

q1.4

N	Valid	776
	Missing	8

q1.4

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Microsoft Windows	727	92,7	93,7	93,7
	Mac OS X	31	4,0	4,0	97,7
	Windows+Linux	10	1,3	1,3	99,0
	Linux	3	,4	,4	99,4
	Combination	2	,3	,3	99,6
	Windows+Mac OS X	1	,1	,1	99,7
	Mac OS X+Linux	1	,1	,1	99,9
	Alternative OS	1	,1	,1	100,0
	Total	776	99,0	100,0	
Missing	999	8	1,0		
Total		784	100,0		

Fig. 37: Operating system selection frequency table. Mac OS X is more popular than both Linux and a combination of Linux with Microsoft Windows.

Source: own work.

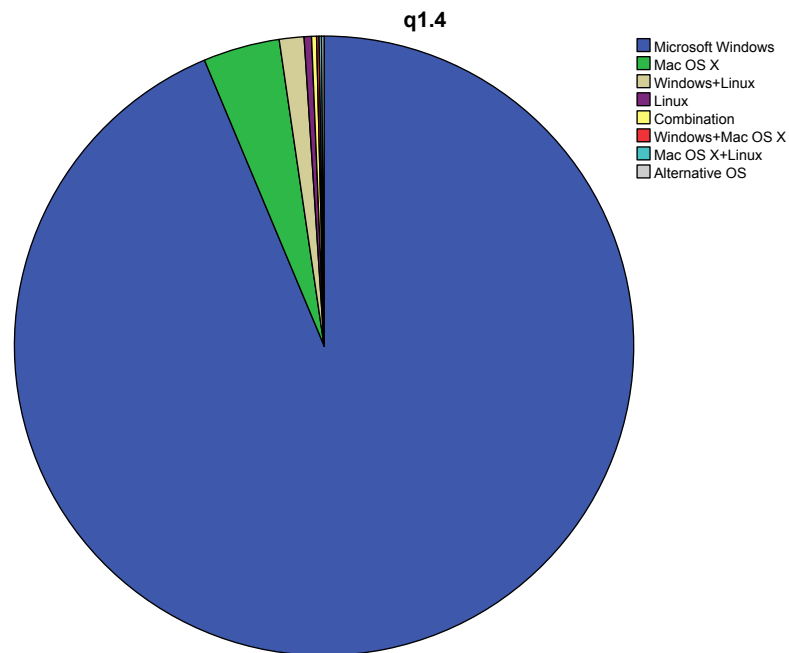


Fig. 38: Operating system selection pie chart. Microsoft Windows dominates with 95 % adoption rate for single installations and dual-boots with Linux.

Source: own work.

the same situation graphically.

Three out of four respondents in the sample have passwords no longer than 11 characters, with 56.4 % having their password between 7 and 11 characters. While the trend of moving away from shorter strings is clear, passwords have several attributes contributing to how resilient they are against reverse engineering: complexity, length, and uniqueness which are discussed in chapter 6.2.4. While emphasis is frequently put on length due to seemingly sound logic behind longer authentication credentials (“The longer the password, the harder it is to guess.”), if the sequence is generic and predictable, length does not provide any security added value.

q1.5

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Website is on the Internet	130	16,6	16,8	16,8
	Website is definitely protected	254	32,4	32,9	49,7
	Website may be protected	122	15,6	15,8	65,5
	Never seen it	107	13,6	13,8	79,3
	Seen but don't know what it is	160	20,4	20,7	100,0
	Total	773	98,6	100,0	
Missing	999	11	1,4		
Total		784	100,0		

Fig. 39: *HTTPS understanding frequency table. The second answer hints at general understanding of protective measures deployed on the Internet.*

Source: own work.

q1.6

N	Valid	771
	Missing	13

q1.6

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	<6 characters	143	18,2	18,5	18,5
	7--11 characters	435	55,5	56,4	75,0
	12--16 characters	155	19,8	20,1	95,1
	17--21 characters	23	2,9	3,0	98,1
	22+ characters	15	1,9	1,9	100,0
	Total	771	98,3	100,0	
Missing	999	13	1,7		
Total		784	100,0		

Fig. 40: *Password length frequency table. More than 18 % of respondents use password shorter than 6 characters.*

Source: own work.

Composition is therefore of importance as well; sadly, lowercase and uppercase characters, numbers, and special symbols do not occur in real-world passwords due to decreased comfort during typing. Password managers are a viable option in the proposed ICT governance model in chapter 6. We hypothesize the respondents favor selection rules which can be summarized by one or multiple mutators in Table 6, integrated in password-cracking software. The following four figures demonstrate composition properties, specifically lowercase (Figure 42) and uppercase (Figure 43) characters, symbols (Figure 44), and spaces (Figure 45). However, it must be assumed answers for special characters may or may not include numbers and thus should be treated as polluted and possibly unreliable. Due to space constraints, frequency tables were placed in Appendix D.

The results confirm users prefer comfort rather than security when composing and changing passwords. Out of 721 valid answers for Q1.7, the following was discovered:

- 83 (11.5 %) changed their password in last 30 days,
- 108 (15 %) did so during last three months,

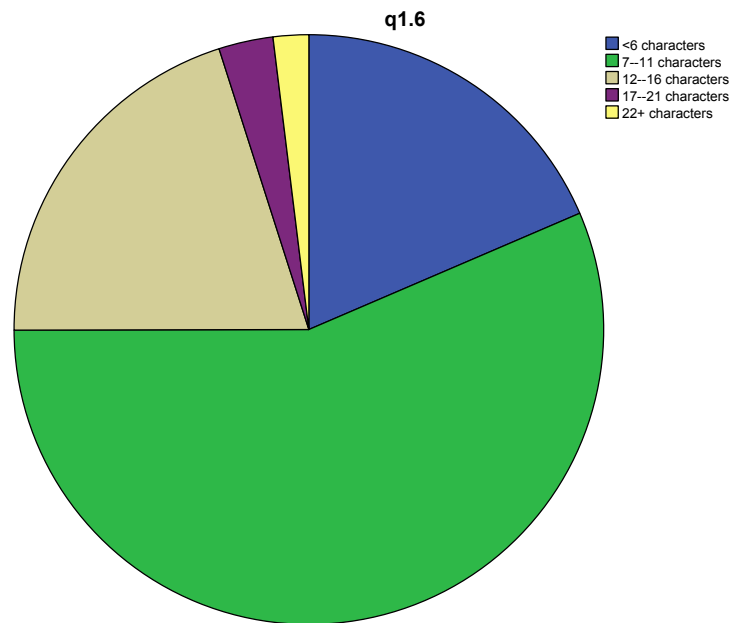


Fig. 41: Password length pie chart. Passwords of length 7–11 are prone to several attacks under realistic assumptions on consumer-grade hardware.
Source: own work.

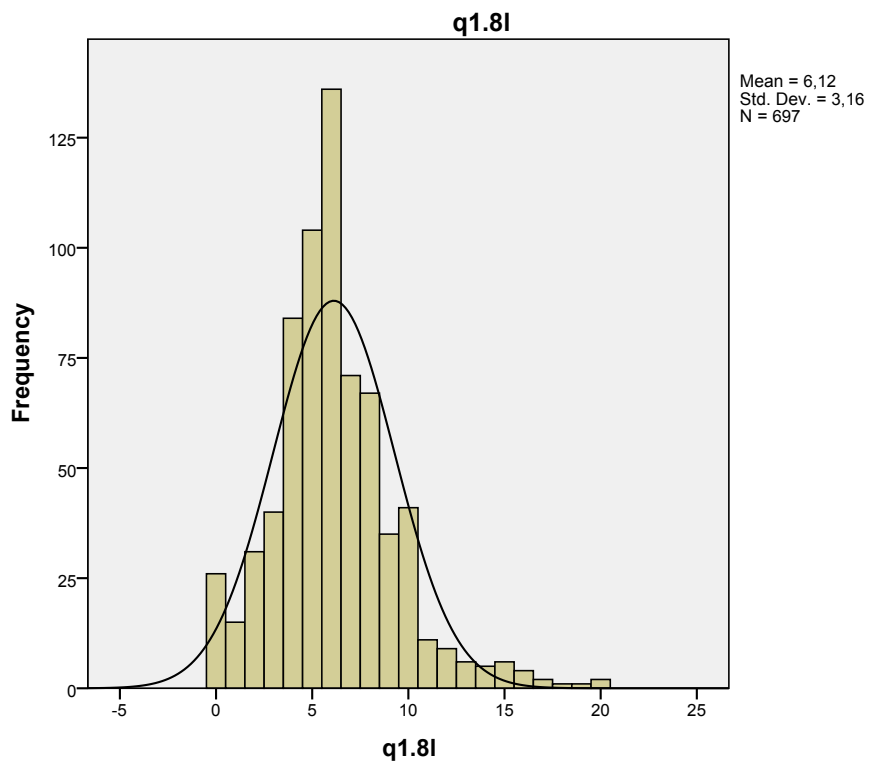


Fig. 42: Password composition: lowercase characters. The histogram can be approximated using Gaussian distribution with $\mu = 5.12$ and $\sigma = 3.16$.
Source: own work.

- 121 (16.8 %) no more than 6 months back,
- 218 (30.2 %) during the last year,
- 186 (25.8 %) never changed their password,
- 5 (0.7 %) provided different answers usually stating they cannot remember exactly.

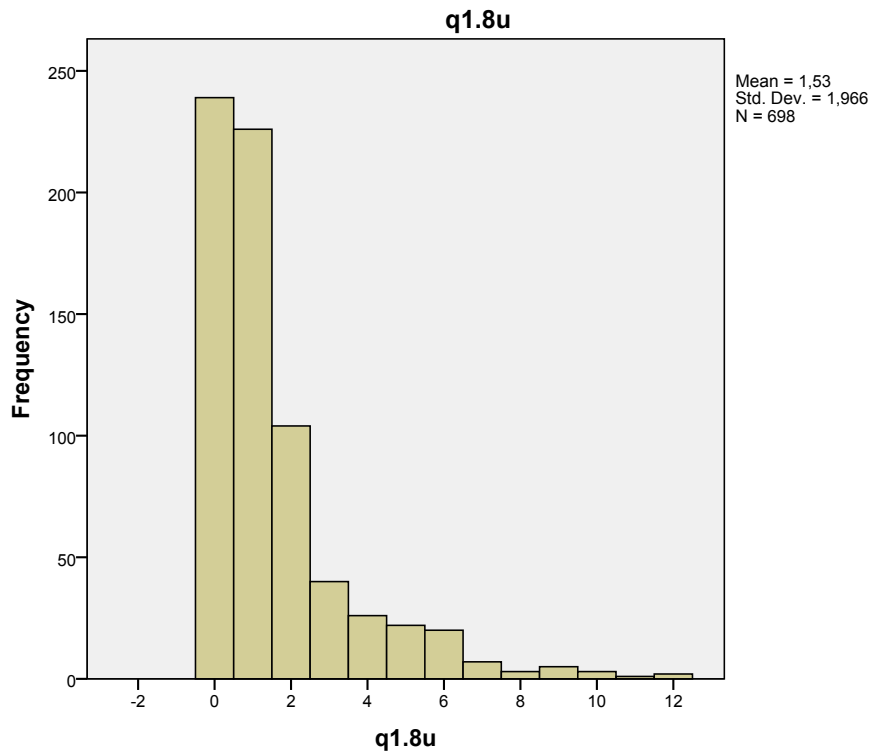


Fig. 43: Password composition: uppercase characters. The distribution is skewed toward lower values, the majority of respondents do not have any uppercase letters in their password.
Source: own work.

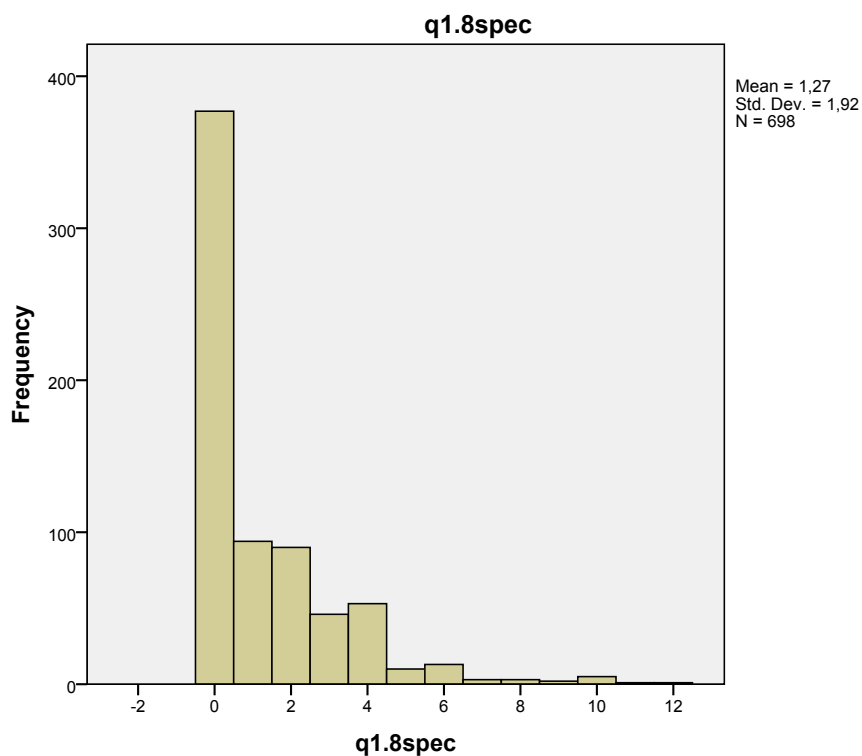


Fig. 44: Password composition: special characters. Symbols are frequently omitted even though they improve security by increasing complexity, uniqueness, and length.
Source: own work.

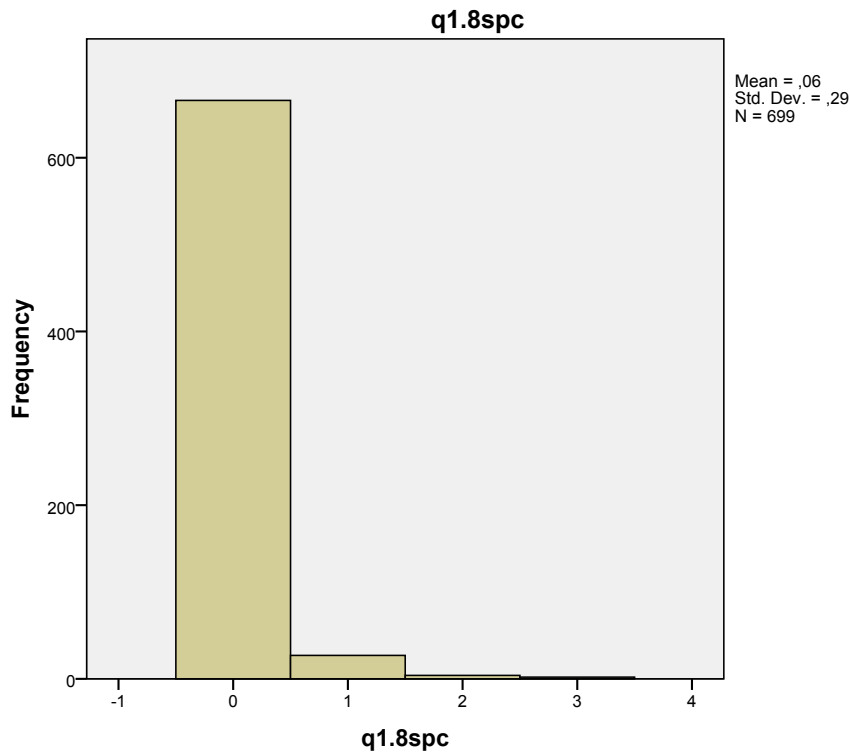


Fig. 45: Password composition: spaces. Spaces are rare either because the service does not allow them inside the authentication string, or as a conscious choice.
Source: own work.

A combination of weak passwords and their slow replacement creates viable attack vectors. Should the adversary get hold of obfuscated strings from a database breach or by other means, 56 % of users from the sample would use the same password for a year or more, and 72.8 % at least for 6 months. This ensures sufficient time to mount sophisticated resource- and time-intensive techniques. Passwords predominantly include lowercase characters, 1–2 uppercase letters and symbols on average, and almost no spaces, limiting the search space considerably; uppercase letters can be with high probability expected on the very first position. These constraints allow to prune the candidate list further, decreasing the time factor involved. Moreover, further assumptions may hold which stems from total length. While it is almost certain shorter (at most six-character) passwords were selected for memorability, the same could likely be said about the 7–11 group, a length well within the capacity of the average user to remember. We may hypothesize even very long strings could be defeated if suitable dictionary is employed and mutators applied. Case study 1 will attempt to reverse engineer a database of passwords using naïve brute-force enumeration and various rules based on a candidate word list. It is expect more than 50 % of strings will be cracked using a single dictionary and successively iterating through the entire search space up to a given bound.

We will calculate Pearson’s chi-squared test to determine whether frequency with which passwords are changed influences their length, i.e., if users deliberately choose shorter passwords, and rotate them regularly and vice versa (longer passwords tend to be used longer). Since the data originates from the same data set, the test prerequisites are assumed to be met. Pearson’s chi-squared test hypotheses:

- null H_0 : No statistically significant association exists between password length and frequency of change, i.e., length is not a factor when changing passwords.

- alternative H_1 : Statistically significant association exists between password length and frequency of change, i.e., length is a factor when changing passwords.

Observed and expected frequencies and p-value for Pearson’s chi-squared test are depicted in Figures 46 and 47, respectively. Exact p-value, obtained by complete enumeration did not differ from the asymptotic version at three decimal positions. We have strong evidence against the null hypothesis, and therefore can conclude length is a factor when changing passwords. It cannot be claimed users would decide based solely on string length, though, other aspects certainly contribute as well but their influence was not conducted due to lack of data. We hypothesize many users do not rotate passwords on schedule but at irregular intervals without necessarily improving security, i.e., authentication credentials are picked for their memorability, not a combination of complexity, length, and uniqueness. Q1.10 and Q1.11 evaluate password selection rationale and storing habits along with reuse, respectively.

q1.7 * q1.6 Crosstabulation

			q1.6					Total
			<6	7--11	12--16	17--21	22+	
q1.7	Last 30 days	Count	12	33	23	12	2	82
		Expected Count	15,1	46,2	16,8	2,3	1,6	82,0
	1--3 months	Count	10	62	28	2	6	108
		Expected Count	19,9	60,9	22,1	3,0	2,1	108,0
	No more than 6 months	Count	10	74	31	4	1	120
		Expected Count	22,1	67,6	24,6	3,3	2,3	120,0
	A year	Count	35	134	43	2	2	216
		Expected Count	39,8	121,7	44,3	6,0	4,2	216,0
	No change yet	Count	65	98	22	0	1	186
		Expected Count	34,2	104,8	38,1	5,2	3,6	186,0
	Something else	Count	0	3	0	0	2	5
		Expected Count	,9	2,8	1,0	,1	,1	5,0
Total		Count	132	404	147	20	14	717
		Expected Count	132,0	404,0	147,0	20,0	14,0	717,0

Fig. 46: Password length and frequency of change table. Some cells are underrepresented but calculating p-value exactly did not change the results.

Source: own work.

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	158,676 ^a	20	,000
Likelihood Ratio	114,306	20	,000
Linear-by-Linear Association	52,331	1	,000
N of Valid Cases	717		

a. 13 cells (43,3%) have expected count less than 5. The minimum expected count is ,10.

Fig. 47: Password length and frequency of change Pearson’s chi-squared test. Even though p-value is low, it should not be assumed zero due to truncated decimal expansion.

Source: own work.

To summarize, it was found passwords of shorter lengths are common in the sample which coupled with slow password rotation generates a window of opportunity for perpetrators. We also presume password reuse across sites is routinely practiced as it increases user comfort

but expands the attack surface. A breach on one site will affect all accounts such as email, banking, corporate domain accounts with access to sensitive electronic assets, health records, social network profiles, and others sharing the credentials. Repercussions range from mild to a complete takeover of individual's online identity.

Out of 771 valid answers, 32 (4.2 %) indicated willingness to provide login credentials for their most important account, and 739 (95.8 %) either declined outright, or asked for higher monetary compensation than the 5 EUR offered in the question text. This suggests users mostly recognize their passwords as sensitive, and attempt to prevent direct third-party disclosure. To what extent do the results reflect real-world situation remains unknown, though, because presenting the same offer in person may generate contrasting response. Considering probable password reuse, such social engineering campaign may be highly successful but has not been pursued and constitutes a suggestion for future research.

Q1.10 focuses on password selection rationale, frequency table and a bar chart are shown in Figures 48 and 49, respectively. Major composition rules identified are dictionary entries without modifications, and prepending/appendixing them with numbers, letters, or symbols; minor rules are typing random characters and substitution. However, as will be discussed in chapter 5.1.3, all are trivial to implement and automate in software and therefore vulnerable. Dictionary entries in particular only necessitate matching the correct word with the password, an algorithm executable in parallel even on consumer-grade setup. Depending on the quality of the prepended/appended sequence, the same could be stated of the rule. Joining multiple common words together increases the time factor but should still be treated as unsafe because dedicated hardware components or cloud computing virtual instances can substantially speed up the computations while not being overly costly for a dedicated attacker. Substitution does little to thwart reverse engineering and due to its popularity among users, the rules have evolved and are highly effective against common alterations. Typing random characters may seem secure, but since the strings need to be memorized and no password manager is used, they likely represent keyboard patterns, e.g., *qwerty*, guaranteed to be included in most word lists freely available on the Internet. The preferred, secure alternative is to deploy credentials managers as detailed in chapter 6.2.4.

Results for Q1.11 are depicted in Figures 50 and 51. The most popular choices for storing and reusing passwords are one master string for multiple accounts, and memorizing several unique credentials without reuse. The other two sparsely-populated categories are writing the password down and keeping it either private, e.g., wallet or public, e.g., office desk for reference purposes. The secure alternative, password manager, was represented by a negligible proportion of respondents, 0.8 %. The risks associated with the four options are obvious and could be summarized as:

- master password: multiple-account hijacking, complete electronic identity takeover, reputation damage,
- several unique credentials: proneness to reverse engineering attempts due to memorability and their tailoring for typing comfort,
- password written down and kept private: direct observation (shoulder surfing), physical acquisition (theft),
- password written down and kept public: stealthy acquisition when the individual is not present at their workplace (insider threat).

Vulnerabilities when using a single string were mentioned previously, but memorizing unique credentials does not improve security as it is highly improbable an individual can recall multiple complex sequences consisting of symbols from all character sets organized so that they are impervious to dictionary and brute-force attacks. A more plausible scenario is several memorable passwords consisting of dictionary words combined together, or a suitable, consistent

q1.10

N	Valid	749
	Missing	35

q1.10

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Pet, husband/wife, child, city, date of birth, movie, song, quote, dictionary word (1)	248	31,6	33,1	33,1
	1+3	1	,1	,1	33,2
	1+5	1	,1	,1	33,4
	Same as 1 but prepended/appended characters (2)	249	31,8	33,2	66,6
	Same as 1 but doubled/substituted characters (3)	82	10,5	10,9	77,6
	3+6	1	,1	,1	77,7
	Typed random characters (4)	117	14,9	15,6	93,3
	4+6	1	,1	,1	93,5
	Web page/program (5)	7	,9	,9	94,4
	Another way (6)	42	5,4	5,6	100,0
	Total	749	95,5	100,0	
Missing	999	35	4,5		
Total		784	100,0		

Fig. 48: Password selection rules frequency table. The question permitted multiple answers, major ones were assigned numbers and their combinations denoted in shorthand, e.g., 4+6.

Source: own work.

substitution scheme applied. While the threat is reduced to one or at most several accounts, credentials management is cumbersome and concessions likely made, e.g., changing just the prepended/appended part and leaving the remainder unchanged when rotating the password. Writing down sensitive information is strongly discouraged as physical acquisition becomes viable. For instance: in a shared office space, the string could be copied by malicious insider when the victim is not present, leading to covert account compromise. The model proposed later in the thesis introduces password managers, software which encrypts login credentials and largely mitigates risks of memorization and other practices undermining security.

4.2.2 Mobile Phones, Additional Questions

Out of 782 valid answers, 518 respondents (66.2 %) acknowledged owning a smartphone or a tablet, 264 (33.8 %) answered negatively. Though the research was a one-time event, conducting the survey at multiple reference points would allow for analyzing whether the adoption rates increase, stagnate, or decrease over time. Assuming the general population gradually migrate toward newer technologies, it can be hypothesized smartphone share will rise, especially as cheaper models are promoted which justify cost for consumers whose needs have so far been satisfied by feature phones. Mobile operating system preferences are depicted in Figures 52 and 53.

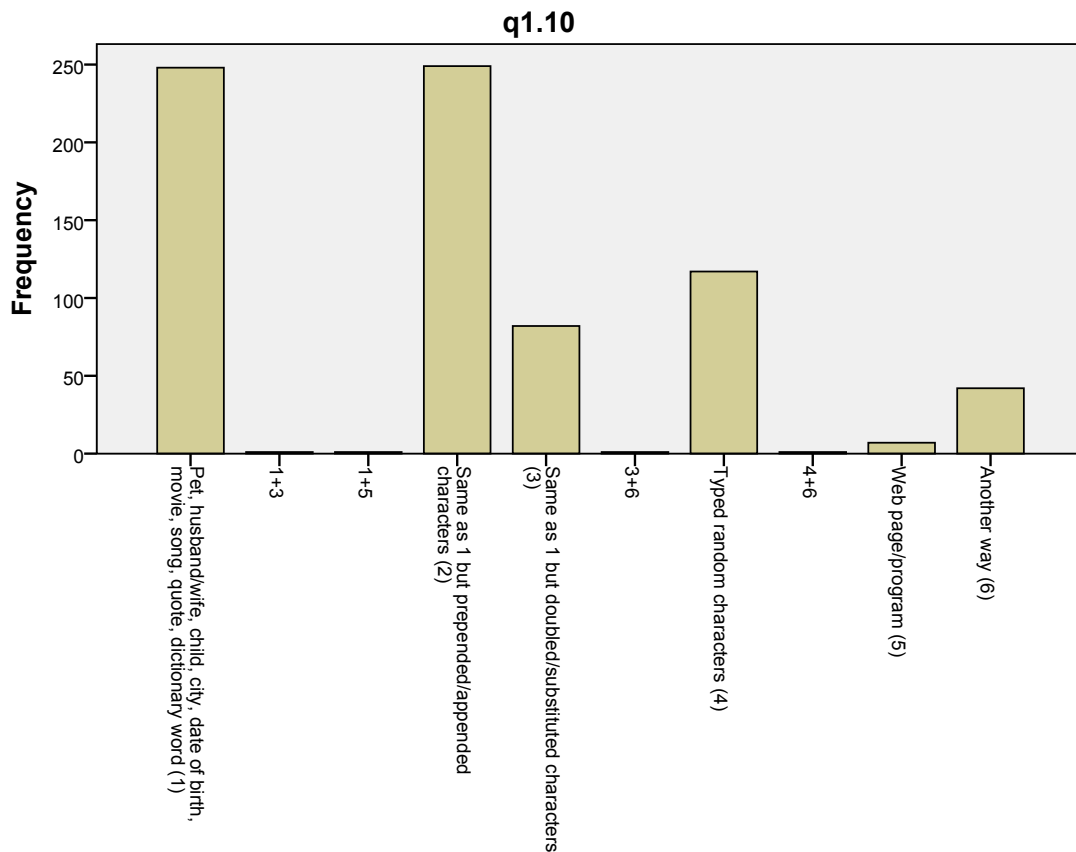


Fig. 49: Password selection rules bar chart. Two major and two minor preferences are apparent which correspond to mutators integrated in password-cracking software. Source: own work.

Android is the most popular mobile operating system followed by iOS and Windows Phone. However, the data might be polluted if participants chose “None” despite not owning a smartphone, inflating the count. Constructing a contingency table for Q2.1 and Q2.2, it was concluded 159 people who answered negatively to the former question selected “None” for the latter. While this shifts the percentages, absolute counts remain identical. Original data was therefore included and the discrepancy marked textually here. Reasons for Android’s popularity are lower price and inclusion in broader portfolio of models compared with iOS. It had been suspected financial incentives are pivotal in deciding about smartphone purchase but Pearson’s chi-squared test for Q2.2 and Q4.4 did not result in rejection of the null hypothesis ($df = 28$, $p\text{-value} = 0.078$) stating no statistically significant association exists between income and mobile operating system preference. This leads us to conclude one does not influence the other. Other factors, e.g., brand, application ecosystem, and previous experience could play a more important role.

Q2.3 pursues the idea by querying for reasons why the particular mobile operating system, and in extension the platform, is favored. From 643 valid answers, the following were listed as the most important:

- applications (117, 18.3 %),
- it was a gift (110, 17.2 %),
- aesthetics (52, 8.1 %),
- previous experience (50, 7.8 %),
- brand (44, 6.9 %),
- recommendations from relatives or friends (44, 6.9 %),

q1.11

N	Valid	774
	Missing	10

q1.11

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Remember and reuse (1)	390	49,7	50,4	50,4
	1+2	7	,9	,9	51,3
	1+2+3	1	,1	,1	51,4
	1+4	3	,4	,4	51,8
	1+5	1	,1	,1	51,9
	Remember but don't reuse (2)	302	38,5	39,0	91,0
	2+4	1	,1	,1	91,1
	2+5	1	,1	,1	91,2
	Written down and with me (3)	30	3,8	3,9	95,1
	Written down where I can see it (4)	26	3,3	3,4	98,4
	Password manager (5)	6	,8	,8	99,2
	Another way (6)	6	,8	,8	100,0
	Total	774	98,7	100,0	
Missing	999	10	1,3		
Total		784	100,0		

Fig. 50: Password storing and reuse frequency table. Out of 774 valid answers, at least 748 (96.7 %) can be exploited to gain system access under various realistic assumptions.

Source: own work.

- combination of aesthetics, applications, and brand (23, 3.6 %),
- switch from another platform (22, 3.4 %),
- other reason (18, 2.8 %).

Application ecosystem proves to be the deciding factor when selecting a new phone. Cross-vendor compatibility allows heterogeneous operating systems to run the same programs which abstract from platform specifics and homogenize the virtual space. Furthermore, migration between platforms is made easier when the functionality is guaranteed not to differ for popular applications (browsers, calendars, email clients, maps, social networks). The second most-cited reason, gift, suggests the respondents do not seek particular brand but are content with whatever smartphone they are given as their fundamental capabilities are largely unchanged. We believe consumers switching from feature phones would belong in this category, or alternatively would be influenced by recommendations from relatives or friends, preferring similar user experience. Aesthetics and previous experience were represented almost identically, brand was also cited as somewhat relevant, hinting at certain socio-economic status tied with owing a particular brand. Design philosophy mentioned in chapter 4.2.1 apparently sees some users inclining toward sleek, tightly-controlled touch interfaces while others value broad customization. A switch from another platform could either suggest dissatisfaction or willingness to experiment. Among other reasons, open-source nature and price/quality ratio dominated. Price itself was not singled out once which strongly suggests it is secondary for many participants.

The scope of smartphone use is demonstrated in Figures 54 and 55. Three groups summing to

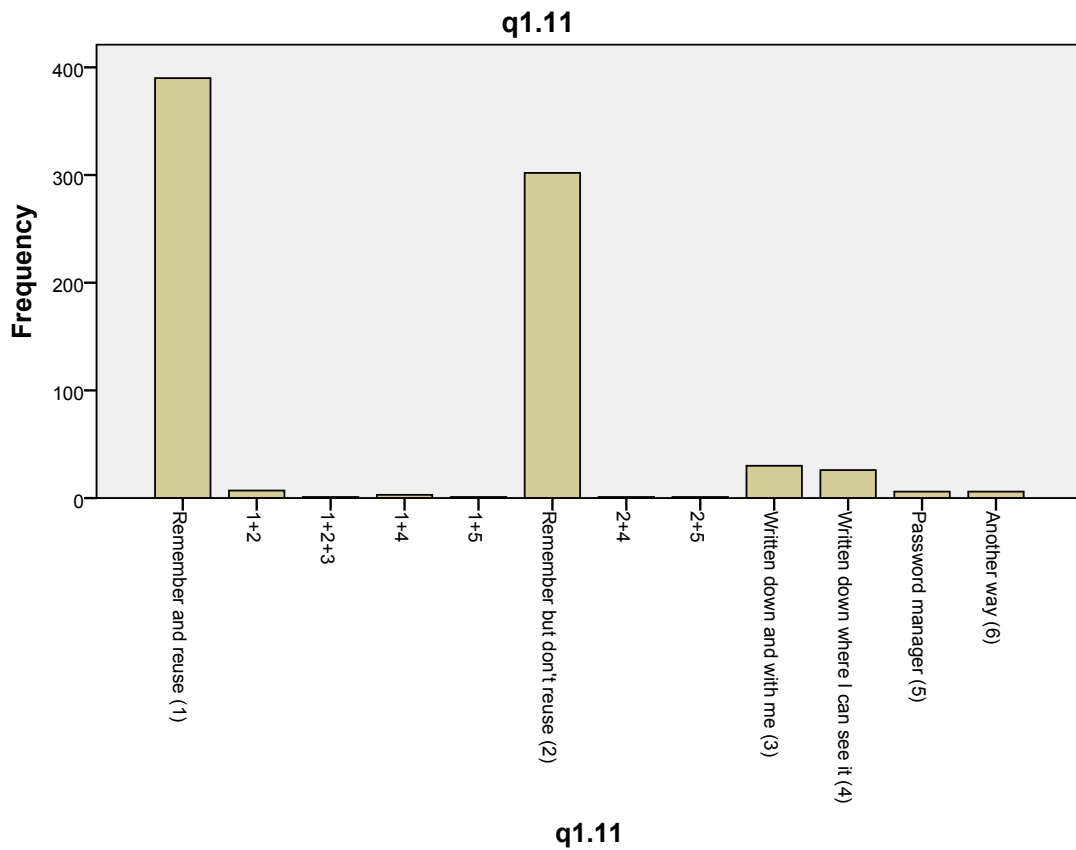


Fig. 51: Password storing and reuse bar chart. The two most populated options were cumulatively singled out by 89.4 % of all participants in the survey.
Source: own work.

q2.2

N	Valid	730
	Missing	54

q2.2

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	None	221	28,2	30,3	30,3
	Android	341	43,5	46,7	77,0
	iOS	98	12,5	13,4	90,4
	Windows Phone	65	8,3	8,9	99,3
	Other	5	,6	,7	100,0
	Total	730	93,1	100,0	
Missing	999	54	6,9		
	Total	784	100,0		

Fig. 52: Mobile operating systems preference frequency table. Missing values represent participants who chose not to answer or do not own a smartphone.
Source: own work.

60.1 % of valid answers are accessing the Internet, basic functions (SMS, calls), and a combination of the Internet, applications, basic functions, and multimedia. The trend of smartphones

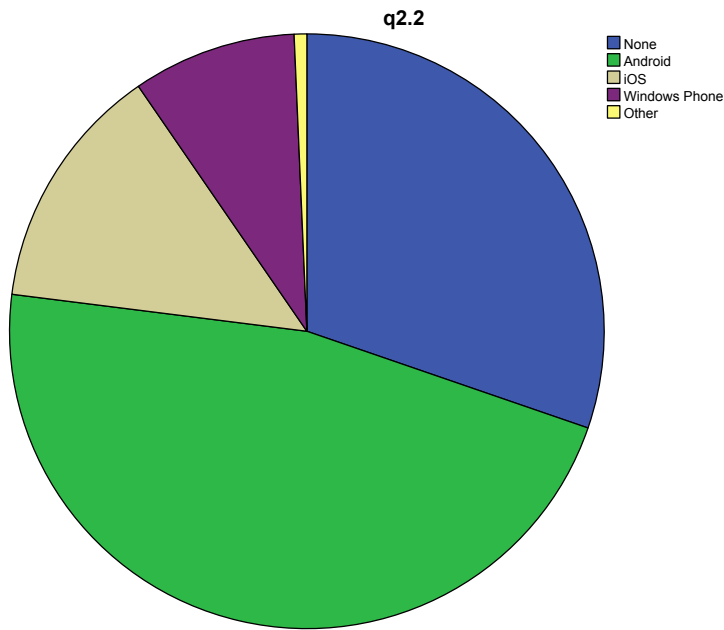


Fig. 53: Mobile operating systems preference pie chart. BlackBerry was not included by default and is included in the “Other” category with 0.7 %.
Source: own work.

complementing portable and stationary computers for connectivity was expected and clearly shows BYOD management in organizations is necessary because security implications of compromised mobile devices will grow as more consumers replace old technology with Internet-enabled endpoints. But it seems trust toward smartphones has not reached a point where they would be considered pertinent for sensitive online operations, e.g., banking, as witnessed in Q2.5: out of 746 valid answers, 469 participants (62.9 %) reported they would rather pick a PC/notebook instead of their mobile device. On the other hand, 243 (32.6 %) do not have issues with using smartphones in such circumstances. Temporal comparison would have provided information about whether the trend is on the rise. If so, we expect the uptake would be much slower especially in new users due to distrust toward and unfamiliarity with the new technology. In time, the attitude may change and accessing sensitive accounts becomes a matter of habit which will lower inhibitions and security awareness in favor of comfort. Therefore, countermeasures such as profiles should be deployed immediately after the device has been integrated into the organizational network.

Q2.6 provides insight into how users compare smartphones and personal computers function-wise. From a pool of 727 valid answers, 346 respondents (47.6 %) believe PCs are superior, while 330 (45.4 %) think the two are comparable. Results are depicted in Figure 56.

Mobile phones so far do not offer computational performance on par with medium- and high-end personal computers, although multi-core hardware which can be considered approximately equal to lower-end stations is becoming commonplace. It can be assumed future advances will improve the specifications further while retaining or decreasing energy draw, a necessity when battery capacity is a limiting factor. Chapter 2.3 provided a brief overview of hardware and software aspects of smartphones. Compared with PCs, smartphones have smaller screen estate, although newer tablets are equipped with larger screens, a trend which will likely proliferate. Apart from their mobility which the respondents were explicitly asked to disregard in the question text, smartphones offer ubiquitous Internet connectivity, GPS, touch interfaces, and power efficiency the PCs/notebooks can reproduce only with difficulty. On the other hand,

q2.4

N	Valid	685
	Missing	99

q2.4

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Email, web (1)	194	24,7	28,3	28,3
	SMS, calls no Internet (3)	137	17,5	20,0	48,3
	1+2+3+4	81	10,3	11,8	60,1
	1+2	46	5,9	6,7	66,9
	1+3	43	5,5	6,3	73,1
	Apps (2)	37	4,7	5,4	78,5
	1+2+3	32	4,1	4,7	83,2
	1+2+4	24	3,1	3,5	86,7
	1+4	24	3,1	3,5	90,2
	1+3+4	17	2,2	2,5	92,7
	Music, movies, photos (4)	12	1,5	1,8	94,5
	3+4	10	1,3	1,5	95,9
	2+3	8	1,0	1,2	97,1
	2+3+4	6	,8	,9	98,0
	2+4	5	,6	,7	98,7
	Other things (5)	4	,5	,6	99,3
	1+2+3+4+5	2	,3	,3	99,6
	1+2+5	1	,1	,1	99,7
	1+2+6	1	,1	,1	99,9
	3+5	1	,1	,1	100,0
Total		685	87,4	100,0	
Missing	999	99	12,6		
Total		784	100,0		

Fig. 54: Smartphone use frequency table. The results were sorted in descending order of observed counts.

Source: own work.

storage capacity is either locked or restricted. Some users also prefer keyboard and mouse peripherals over virtual keyboards, particularly on smaller screens, though this may be subjective. The results indicate the sample is polarized on whether the breadth of functions on PCs (and in extension, their replacement with smaller devices) is comparable with smartphones. An argument can be raised the two setups will be complementary in the future.

Questions 2.7 and 2.8 focus on mobile security, specifically if and how the respondents store passwords on their phones, whether lock screen is enabled and which character classes were included in the string if so. Contingency table and clustered bar chart are shown in Figures 57 and 58, respectively.

Pearson's chi-squared test was calculated with the following hypotheses:

- null H_0 : No statistically significant association exists between storing passwords and lock screen password complexity, i.e., presence or absence of sensitive information does not affect the string composition, and vice versa.

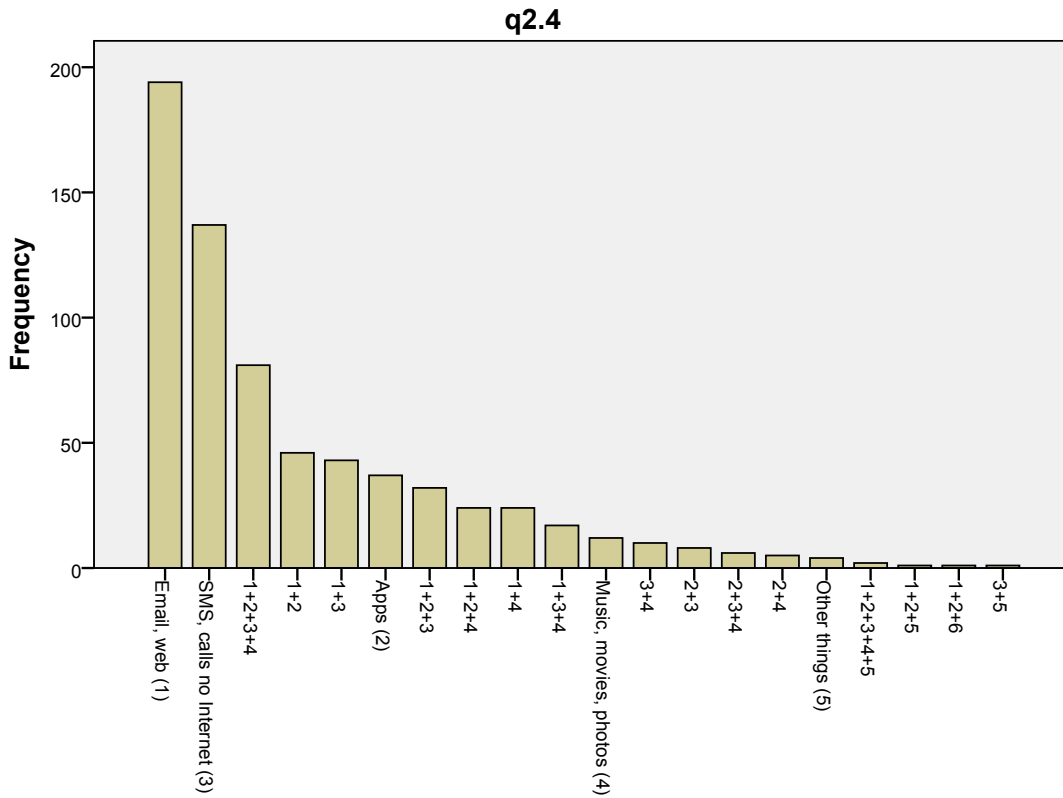


Fig. 55: Smartphone use bar chart. The distribution is positively (right) skewed with a long tail.
Source: own work.

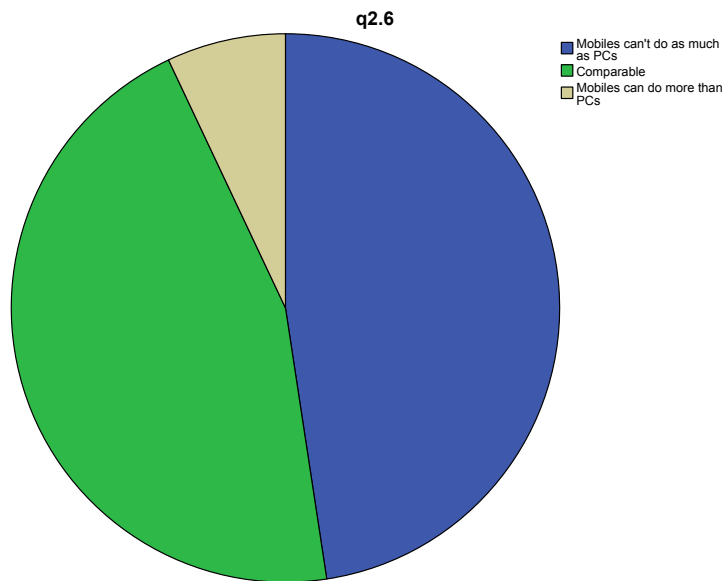


Fig. 56: Smartphones and PCs perceived functions comparison pie chart. The two major categories differed by 2.2 %.
Source: own work.

- alternative H_1 : Statistically significant association exists between storing passwords and lock screen password complexity, i.e., presence or absence of sensitive information does affect the string composition, and vice versa.

For $df = 9$, we obtained p-value of 0.343 which leads us to believe the evidence against the null hypothesis is weak and we are not entitled to reject it. However, because 9 cells (56.3 %) had

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
q2.7 * q2.8	767	97,8%	17	2,2%	784	100,0%

q2.7 * q2.8 Crosstabulation

			q2.8				Total
			No	Numbers	Numbers and letters	Numbers, letters, symbols, spaces	
q2.7	In a reminder	Count	33	11	4	1	49
		Expected Count	34,6	8,7	4,3	1,3	49,0
	Secured/hidden	Count	13	8	0	0	21
		Expected Count	14,8	3,7	1,9	,6	21,0
	No	Count	493	117	64	20	694
		Expected Count	490,4	123,1	61,5	19,0	694,0
	Other way	Count	3	0	0	0	3
		Expected Count	2,1	,5	,3	,1	3,0
Total		Count	542	136	68	21	767
		Expected Count	542,0	136,0	68,0	21,0	767,0

Fig. 57: Smartphone password storing and lock screen contingency table. Counts in nine cells were lower than the threshold value of 5, and exact p-value calculation was used.
Source: own work.

expected counts less than the threshold value, Pearson's chi-squared test likely did not provide correct result. Exact p-value calculation led to p-value of 0.304. Additionally, Fisher's exact test which does not make prior assumptions about the random variable's probability distribution produced p-value of 0.431, i.e., higher than the asymptotic Pearson's chi-squared test. All three support the conclusion presence or absence of sensitive information does not affect the string composition, and vice versa. The fact passwords are saved on the device does not influence the decision to enable lock screen or indeed, how complex the string protecting the phone is. Figure 58 makes it clear storing login credentials is rare as well as having a lock screen enabled. While the former is commended, missing lock screen presents a threat when the phone is misappropriated because data can be trivially siphoned off. Even though passwords would not be affected based on the data, contact list entries, text messages, photos, reminders, and emails would be available for inspection and exploitation. It is hypothesized user comfort again influenced the results: the survey shows users are much more likely to sideline fundamental security procedures (setting a numeric code) so that the device can be accessed quickly when needed. BYOD profiles should reflect this by enforcing lock screens and minimum password complexity. Alternative means of authentication (biometric tokens, graphical passwords) should be evaluated as to the repercussions in case of compromise.

Question 2.9 pertains to BYOD profiles, a central aspect of mobile management in organizations. The results are demonstrated in Figures 59 and 60. Around 43 % of respondents claimed profiles are unacceptable and dismissed them outright, 17.6 % were not against having programmatic safeguards in place as long as they can be turned off should the need arise. Some users (22.2 %) do not see any issues with adding a profile regardless of whether it can be disabled. It is argued profiles are perceived as novel, unproven technology which coupled with lack of information about their underlying technical principles leads to misinformation rectifiable by training and objective evaluation of advantages and disadvantages they pose for mobile devices. This could lessen enmity for profiles and BYOD management generally. Albeit

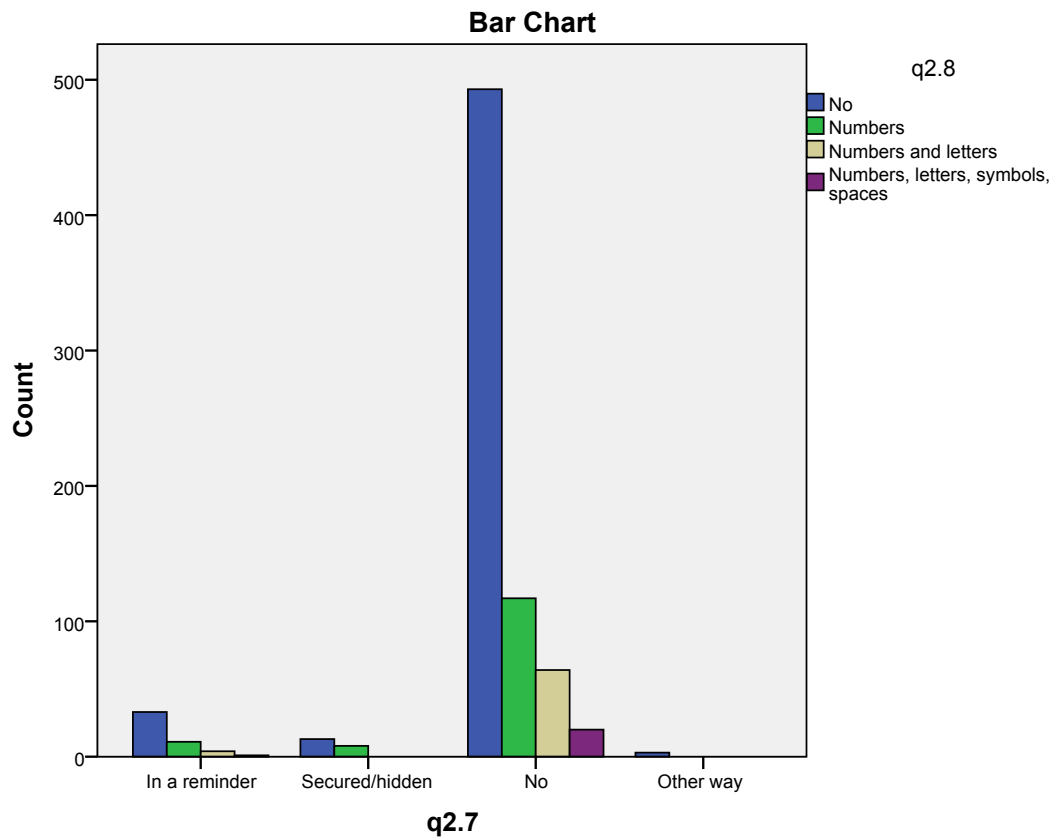


Fig. 58: Smartphone password storing and lock screen clustered bar chart. Participants were consistent in choosing not to store passwords on their phones.
 Source: own work.

q2.9

N	Valid	757
	Missing	27

q2.9

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	I would mind	325	41,5	42,9	42,9
	No, I wouldn't mind	168	21,4	22,2	65,1
	Yes, as long as I can switch it off	133	17,0	17,6	82,7
	Only if the company wanted me to	131	16,7	17,3	100,0
	Total	757	96,6	100,0	
Missing	999	27	3,4		
Total		784	100,0		

Fig. 59: BYOD profiles acceptance frequency table. Reasons for rejecting profiles were not covered in the research but lack of information is hypothesized to be a factor.
 Source: own work.

opposition from some users will remain, the current situation would very likely change and the unacceptability rate decrease in favor of unconditional acceptance, or assurance of administrative

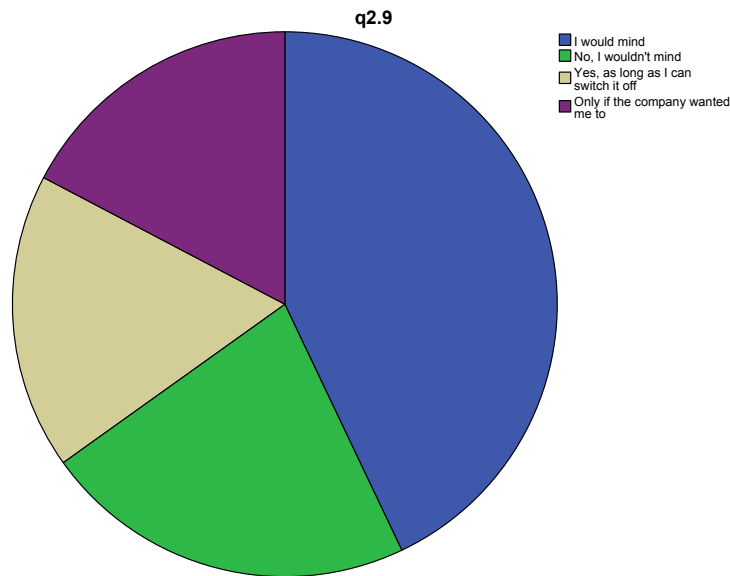


Fig. 60: *BYOD profiles acceptance pie chart. A non-negligible portion of participants (17.6 %) expressed wish to exercise control over their devices but were not against installing a profile on it.*
 Source: own work.

smartphone control retention. On the other hand, profiles are tailored for protecting sensitive organizational electronic assets; users should expect an adjustment period as new policies are gradually implemented in day-to-day operations, e.g., automatic email logout when inactive, enforcing establishment of secure communication channels, and periodic patch deployment cycles.

For the Likert scale in Q2.10, non-parametric Kruskal–Wallis one-way analysis of variance by ranks was selected as the best fit. Unlike parametric analysis of variance, it does not presuppose random variables conforming to particular probability distributions and treats them as unknown, although an implicit assumption is made all observations come from an identical distribution which can be confirmed for the input data. While having lower statistical power, prerequisites such as Gaussian distribution, equality of variances, and independence may not be met and would lead to incorrect results had parametric alternative been used. Given a sufficiently large sample, the test asymptotically approximates the unknown probability distribution using chi-squared distribution with total error decreasing as sample size increases. Hypotheses for Kruskal–Wallis one-way analysis of variance:

- null H_0 : The data comes from distributions where probability that a random sample comes from one distribution is greater than the probability of the other random sample coming from the same distribution, i.e., men and women do not significantly differ in their preferences.
- alternative H_1 : The data does not come from distributions where probability that a random sample comes from one distribution is greater than the probability of the other random sample coming from the same distribution, i.e., men and women significantly differ in their preferences.

The results are depicted in Figure 61. Gender (Q4.1) was selected as a grouping variable, and the results will determine whether men and women differed significantly when ranking preferences for brand, functions/applications, look, price, and security in smart mobile devices.

A total of 5 independent tests were conducted, one for each category with $df = 1$. The second table lists p-values: it can be seen men and women do not deviate significantly when rating security and brand. Conversely, price, functions/applications, and looks were found

Ranks

q4.1		N	Mean Rank
q2.10sec	Female	416	391,06
	Male	357	382,27
	Total	773	
q2.10price	Female	414	371,72
	Male	357	402,56
	Total	771	
q2.10func	Female	415	414,06
	Male	356	353,29
	Total	771	
q2.10looks	Female	415	351,01
	Male	356	426,79
	Total	771	
q2.10brand	Female	415	383,83
	Male	356	388,53
	Total	771	

Test Statistics^{a,b}

	q2.10sec	q2.10price	q2.10func	q2.10looks	q2.10brand
Chi-Square	,323	4,039	15,425	23,963	,091
df	1	1	1	1	1
Asymp. Sig.	,570	,044	,000	,000	,764

a. Kruskal Wallis Test

b. Grouping Variable: q4.1

Fig. 61: Likert scale Kruskal-Wallis test. If mean ranks diverge substantially, the test evaluates the grouping variable as having significant effect on the test variable.

Source: own work.

gender-dependent with high probability. Comparing mean ranks suggests women appreciate security together with functions/applications more than men who put more emphasis on price and looks, corroborating smartphones could be perceived as a sign of social status. Brand was rated similarly which suggests participants from both genders in the sample do not vary in their perceptions. The results should not be expected to hold when purchasing decisions are made, though: users are more interested in brand, functions/applications, and looks than security. Current iterations of popular mobile operating systems (Figure 53) include similar security additions, and smartphones can be postulated adequately protected regardless of vendor. The exception is BlackBerry whose commitment to corporate security currently outstrips its competitors. We predict the situation will change and more operating systems will make inroads into high-security environments including organizations where strict data protection is required.

Questions 3.1–3.3 analyze online behavior. Q3.1 polls the respondents about opening attachments and links in email messages sent to them by third parties. In 775 valid answers, the following breakdown was observed:

- the attachments and links are never opened: 452 (58.3 %),
- they are still opened despite previously causing problems: 63 (8.1 %),
- increased suspicion but occasionally, an attachment or link is opened: 134 (17.3 %),

- attachments are opened only from known and trusted parties: 126 (16.3 %).

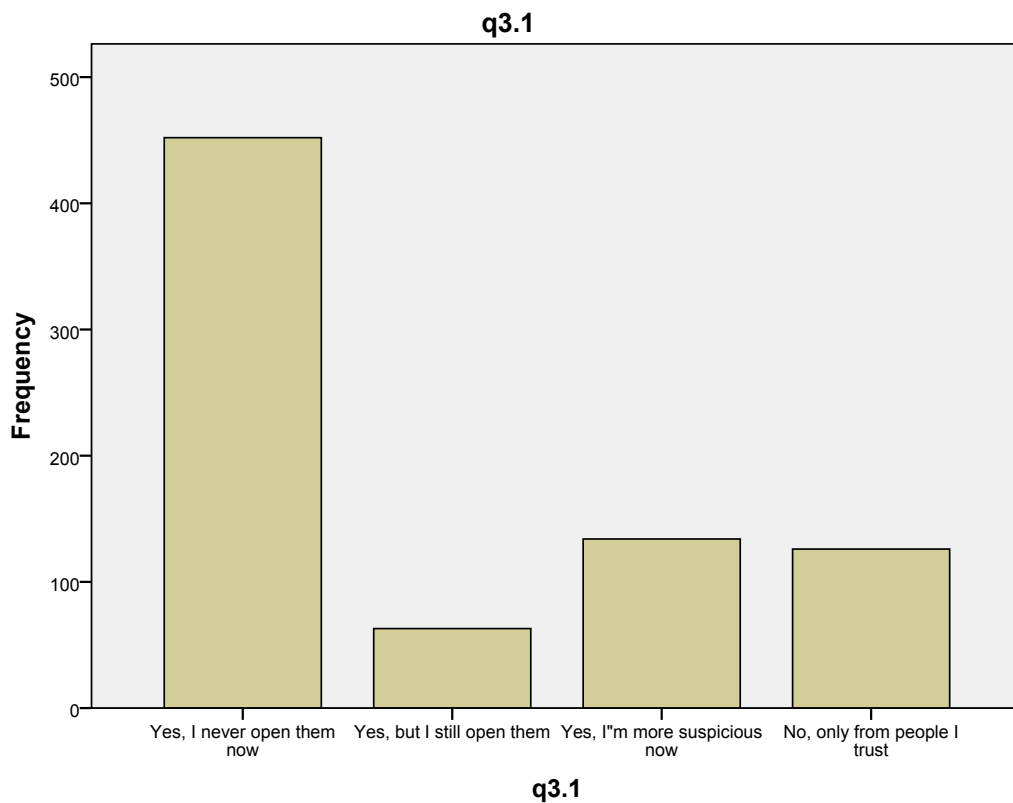


Fig. 62: Spam resilience self-assessment bar chart. Most respondents claimed never to open suspicious attachments and links.

Source: own work.

Results are visualized in Figure 62. More than 50 % of participants do not open suspicious attachments and links, although the wording leaves ambiguous whether only from unknown third parties or anyone regardless of origin. An attacker can devise a scenario in which the email source is spoofed so that it appears to come from a trusted sender; this will be demonstrated later in chapter 5.2.3. Unless spam filter does not classify the message as fraudulent, the recipient must resort to advanced techniques, e.g., header inspection, beyond what can be expected from the average user (Figure 33). The attack thus has high probability of success if the wording in the email body does not give any indication an automated means was used to generate it. Alternatively, the adversary can write the text manually for high-value targets to minimize risk of detection. Hence, the answers represent ideal state where the email is unequivocally categorized as spam by either automatic or human-facilitated inspection; an interesting future research venue would be to analyze how people handle seemingly genuine messages with content tailored to generate interest. It can be hypothesized the proportion of participants disregarding it outright would be substantially lower than in the data collected during the questionnaire research. The concept of trusted parties could be also explored further, e.g., whether the recipient verifies message authenticity via alternative channels (telephone, personal inquiry) when in doubt, or whether the sole decision criteria are sender's address, subject, and text. The findings could provide insight into how trust alters individuals' behavior in online transactions. More than 25 % of respondents admitted opening attachments and clicking on links despite previous problems, which leads us to conclude the email content stirred enough interest so that rational response gave way to an emotional one. Psychology training for resisting such social engineering campaigns will be discussed in the proposed ICT model in chapter 6.1. Ideally, the training should result

in dismissal of the message, preferring elevated false positives, i.e., genuine emails treated as malicious, to false negatives, i.e., fraudulent attachments and links considered harmless. In organizations processing sensitive data, a suspicion-based response should be the first reaction.

Question 3.2 is the only question where the participants were required to write answers in form fields. The author then opted for two codes: don't know/wrong, and correct. The approach necessarily led to loss of information while introducing personal judgment and bias. Even though an effort was made to minimize both, the results are acknowledged to be subjective and should be treated as such. Where phishing was delimited (150 out of 774 cases, 19.4 %), the respondents offered very specific definitions which suggests they were familiar with the term. Conversely, spam had higher percentage of correct replies (367 out of 774, 47.4 %) which almost uniformly cited unsolicited emails as being its main feature. It is argued proliferation of the term in common parlance contributed to the pervasiveness even in basic-skilled users. However, some incorrectly associated spam with a folder in their email accounts.

Pearson's chi-squared test will determine if the participants were likely to delimit both terms if they did the first one (spam), and vice versa. The hypotheses are:

- null H_0 : No statistically significant association exists between delimitation of spam and phishing, i.e., the respondents were not more likely to describe one if they did the other, and vice versa.
- alternative H_1 : Statistically significant association exists between delimitation of spam and phishing, i.e., the respondents were more likely to describe one if they did the other, and vice versa.

Results, contingency table and a bar chart are depicted in Figures 63 and 64, respectively. We hypothesized an association between the respondents' knowledge of basic and advanced social engineering techniques may not be statistically significant.

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
q3.2sp * q3.2ph	774	98,7%	10	1,3%	784	100,0%

q3.2sp * q3.2ph Crosstabulation

			q3.2ph		Total
			Wrong/don't know	Correct	
q3.2sp	Wrong/don't know	Count	401	6	407
		Expected Count	328,1	78,9	407,0
	Correct	Count	223	144	367
		Expected Count	295,9	71,1	367,0
Total		Count	624	150	774
		Expected Count	624,0	150,0	774,0

Fig. 63: Spam and phishing delimitation contingency table. No cell count was below the recommended threshold but Fisher's exact test was used for verification purposes.

Source: own work.

For $df = 1$, p-value was listed as 0 at three decimal positions for both Pearson's chi-squared and Fisher's exact test. The data does not provide enough evidence to support the null hypothesis which leads us to conclude a statistically significant association exists between delimitation of

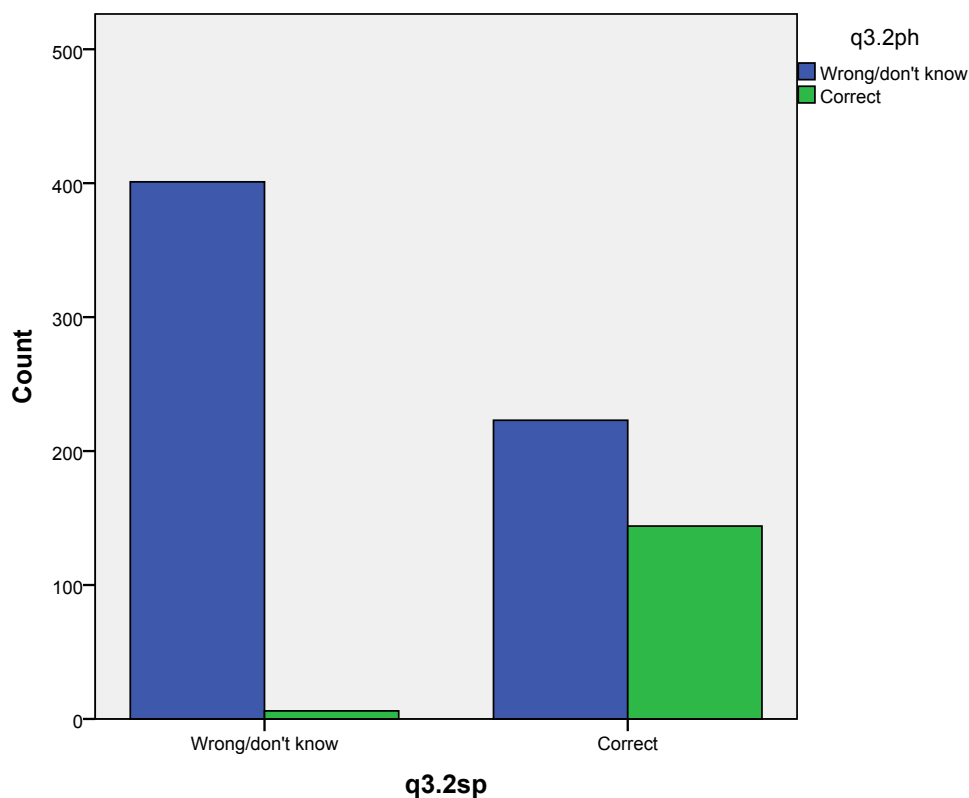


Fig. 64: Spam and phishing delimitations bar chart. The graph may hint at some association between the two variables.
 Source: own work.

spam and phishing. If the participant answered the first part correctly, it can be assumed they will do the same for the second part, and vice versa. Inspecting the contingency table, 6 people correctly delimited phishing while being wrong or not knowing what spam is; conversely, 223 were aware of spam but not phishing. Spam is thus considerably more widely-known than phishing which opens a window of opportunity the attacker can exploit because phishing is bound to be more effective. The bulk of unsolicited emails with links to sites hosting malicious software are trivial to filter by automated filters, and users are somewhat aware of the risks. Conversely, customized messages using the victim’s name, presented in their native language, and grammatically correct are much more likely to illicit a response, particularly when the recipient is not aware of common phishing techniques. Long-term employee education is strongly recommended for organizations to raise their security profile and to protect data in information systems from unauthorized access. The results show improvements are needed, especially as the attack vector will become more sophisticated.

Question 3.3 presents a hypothetical scenario in which the respondents were asked if they would seriously consider engaging in online criminal activities should the tools be freely available. Results are depicted in Figures 65 and 66. The question aimed to discern whether relative anonymity on the Internet is a factor when deciding about legally-dubious conduct. The data suggests most people do not contemplate first-hand participation in computer crime seriously, probably due to limited knowledge and lack of interest. It can be further argued respondents are aware that restrictions are put in place for critical electronic assets, and deep technical expertise beyond skills of the average user is required to circumvent them. Computing devices may also be perceived as tools expected to simply function (ICT as a service) and perform routine tasks rather than machines exploitable for nefarious purposes. The question text states the scenario is

q3.3

N	Valid	775
	Missing	9

q3.3

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	579	73,9	74,7	74,7
	Yes, as long as not caught	121	15,4	15,6	90,3
	Yes, for money	57	7,3	7,4	97,7
	Own opinion	18	2,3	2,3	100,0
	Total	775	98,9	100,0	
Missing	999	9	1,1		
Total		784	100,0		

Fig. 65: Willingness to engage in computer crime frequency table. Opinions in the form field mostly reiterated one of the offered options in participants' own words.
Source: own work.

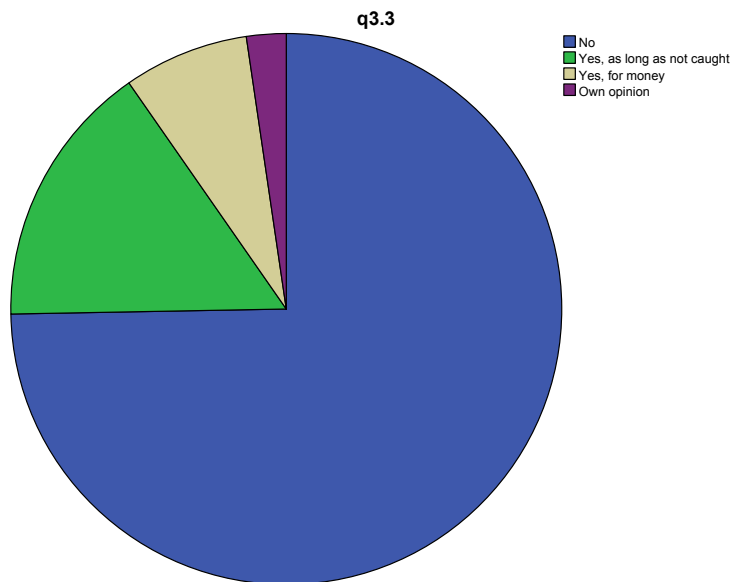


Fig. 66: Willingness to engage in computer crime pie chart. Almost 75 % of the answers rejected the idea of illegal activities on the Internet.
Source: own work.

speculative but chapter 5 will demonstrate the tools are available for download free of charge, and an attack of choice can be launched using a single command. Tutorials can be trivially found, empowering anyone with potent set of programs which can be used for legitimate (penetration testing, chapter 2.4.8) and malicious purposes. Knowledge and ethics are therefore the main limiting factors of computer crime. A total of 178 (23 %) participants would be willing to engage in criminal activities on the Internet either for monetary compensation, or promise of remaining undetected. The result could be considered troubling but we hypothesize the proportion would be different and likely lower should the individuals be presented with the same opportunity in the real world.

4.3 Conclusion

The questionnaire research mapped habits and viewpoints pertaining to ICT and mobile security in a representative sample of 784 respondents from the Czech Republic. The findings will form basis for for case study 1 in chapter 5.1 and user-side solutions presented in chapter 6. Statistical tests were performed to the best of author's abilities, and alternatives with fewer assumptions were preferred. Sample size was considered sufficient for acceptable approximation errors.

Shortcomings of the research can be stipulated to be twofold: subjectivity and lack of advanced statistical methods for data analysis. Subjectivity is pertinent to Q3.3 in which the participants were asked to delimit electronic spam and phishing with the answers reduced to codes: 0 for incorrect/missing, and 1 for correct specification. This may have led to loss of information and subjective interpretation. Other questions where ambiguity may have resulted in skewed results were Q1.8, specifically concatenation of numeric and special characters in passwords, Q2.2 which inquired about a preferred mobile operating system without specifying only smartphone/tablet users should answer, thus inflating the count, and Q2.11 not included in the analysis. Where possible, the problem was addressed by exact calculation (Q2.2), or explicit acknowledgment the interpretation may be incorrect (Q1.8, Q3.3). However, all findings ultimately rely on accuracy and reliability of the source data, and a coordinated effort on part of the participants to deliberately include erroneous information could produce substantial shifts in the output. This disadvantage is inherent to any qualitative research method and can be mitigated by applying experience and third-party evaluation. As the participants were assured the questionnaires would not be scrutinized by external parties, the latter was not utilized.

The second shortcoming, lack of advanced statistical methods, was a conscious decision due to interpretation difficulties, nature of the questions, and space constraints. Interpretation difficulties stem from multitude of prerequisites for advanced tests whose violation would make the results valid only for a limited number of cases. It is the author's opinion statistical methods where assumptions are met and the output applicable with fewer restrictions constitute sounder research methodology than purposefully preferring complex alternatives. Additionally, nature of the questions did not permit much variety because categorical variables lend themselves to a limited subset of analytical methods. The questionnaire was designed for accessibility and textual rather than numerical input. Space constraints were also an issue: advanced tests require at least fundamental background to be presented and explained, prerequisites need to be verified, and impacts discussed in case of deviations. This reduces the number of tests which can be performed. Conversely, the approach taken in the thesis uses fewer methods repeatedly which allows to present broader range of results.

Case study 1 will be based on the findings from Q1.8, Q1.10, and Q1.11 which dealt with real-world password composition, selection, and reuse policies in the representative sample. A data set of test passwords will be analyzed, hypothesizing the QR conclusions can be generalized to the whole population, i.e., statistical inference. We can reasonably assume that if they exist, the differences will be marginal with the majority of the authentication strings vulnerable to exhaustive enumeration and dictionary attacks. The ICT governance model will then address the issue by introducing ways to make reverse engineering prohibitively time-consuming so that rational attackers conforming to the model mentioned in chapter 2.4.1 are forced to seek alternative vectors of approach. Further benefits of the research include a snapshot of real-world practices in ICT and BYOD with actionable intelligence for any organization whose management wishes to increase security and proactively improve policies. The sample size is sufficient for statistically-significant deductions applicable to marketing (Likert scale preferences for smartphone sales), general knowledge (declared ICT competencies), and identification of

behavior patterns when faced with spam and social engineering campaigns. The author opines the research not only contributed to understanding of important phenomena currently taking place, but was necessary to update current state of knowledge about how users keep pace with rapid technological advances we have been witnessing for some time.

Future directions include repeating the research and comparing temporal shifts in preferences, trends, and opinions, particularly regarding the use of mobile devices for financial and other high-risk operations, password practices, acceptance of BYOD profiles and their prevalence in organizations. Managers could also be polled about how businesses perceive growing number of mobile devices accessing internal electronic resources, what measures are taken to protect them, scope of ICT policies as well as what returns are expected from security investments. Finally, the sample size could be expanded by polling respondents on the Internet and globally; in that case, data reliability and validity would have to be assured.

5 CASE STUDIES

In the following chapters, two case studies will be presented which will demonstrate how adversary with moderate knowledge of security can mount sophisticated attacks using freely-available tools, harnessing performance of consumer-grade hardware. Termed “script kiddy,” this type of perpetrator is defined as “[t]he more immature but unfortunately just as dangerous exploiter of security lapses on the Internet. The typical script kiddy uses existing and frequently well known and easy-to-find techniques and programs or scripts to search for and exploit weaknesses in other computers on the Internet—often randomly and with little regard or perhaps even understanding of the potentially harmful consequences” (Mead, Hough, & Stehney II, 2005, p. 72). While unsophisticated, script kiddies are capable to cause harm on a large-scale due to sophistication and breadth of functions the tools contain, making the potency and the features directly proportional. Chapter 2.4 demonstrated multiple exploitable vectors in ICT infrastructures which require little interaction apart from running programs automating the tasks. Script kiddies should be considered a serious threat, perhaps even more so than highly-skilled individuals who prefer targeted actions, leaving basic vulnerabilities able to destabilize firms’ operations dormant, to be picked by others.

Script kiddies represent the lower bound on the threat potential generated. While some industries, e.g., financial, research and development, and technology attract more malicious attempts at asset misappropriation, appropriate security policies should be set along with countermeasures and employee participation programs. The case studies will focus on entities dealing with and retaining sensitive data regardless of origin but in need of protection. Examples are provided in Table 8.

Tab. 8: *Types of sensitive data. The table is an upper bound on what types of electronic information are processed. Electronic shops and social networks require user verification and the amount of data required to prove their identities is thus extensive. Universities store student-related and employee-related records, may provide electronic mail, forums, and social networking, all of which expand the attack surface. ADD: Addresses, DOB: Dates of birth, IP: IP addresses, EM: Email addresses, FIN: Financial data, TEL: Telephone numbers, PASS: Passwords, MISC: Miscellaneous.*
Source: own work.

Entity	ADD	DOB	IP	EM	FIN	TEL	PASS	MISC
Corporate website	X		X	X		X	X	
Electronic shop	X		X	X	X	X	X	Payment and purchase history
Email provider			X	X			X	
Forum			X	X			X	
Hospital	X		X			X		Medical history and records
Search engine			X					Search and location history
Social network	X	X	X	X	X	X	X	Location history, photographs
University	X	X	X	X		X	X	ID numbers, marks

Electronic shops and social networks are based on unique client identification, the span of information requested is very broad, and everything is collated into a single profile. Strict laws

pertaining to personally-identifiable data, e.g., Payment Card Industry Data Security Standard (PCI DSS), require compliance through implementation of baseline precautions for thwarting attacks on asset confidentiality, integrity, and availability. The laws will not be discussed further because they deviate from the doctoral thesis topic. Universities constitute viable targets: they deal with data pertaining to students and staff, have high-speed network connections, and multiple exploitable social engineering avenues. Securing desktop stations, notebooks, server infrastructure, and mobile devices connected by software prone to bugs requires cooperation and diligence on part of stakeholders, prerequisites not always met with understanding. ICT gradually shifts toward being perceived as a service; systems are expected to work autonomously and protect users, not the other way around.

The case studies will practically demonstrate steps taken to reverse engineer original strings from one-way hash functions (chapter 5.1), and security auditing process (chapter 5.2). A case study is useful when "... a 'how' or 'why' question is being asked about a contemporary set of events, over which the investigator has little or no control" (Yin, 2008, p. 13). They were selected as a suitable research method to fuse theoretical background and real-world experiments using a methodology which assures objectivity and reproducibility.

Flyvbjerg (2006, p. 3) argues that "... the problems with the conventional wisdom about case-study research can be summarized in five misunderstandings or oversimplifications about the nature of such research. . .," and lists the following:

- "... [g]eneral, theoretical (context-independent) knowledge is more valuable than concrete, practical (context-dependent) knowledge[.]"
- "... [o]ne cannot generalize on the basis of an individual case; therefore, the case study cannot contribute to scientific development[.]"
- "... [t]he case study is most useful for generating hypotheses; that is, in the first stage of a total research progress, while other methods are more suitable for hypotheses testing and theory building[.]"
- "... [t]he case study contains a bias towards verification, that is, a tendency to confirm the researcher's preconceived notions[.]"
- "... [i]t is often difficult to summarize and develop general propositions and theories on the basis of specific case studies."

While some concerns about case studies may be valid, "[t]he issue of generalization has appeared in the literature with regularity. It is a frequent criticism of case study research that the results are not widely applicable in real life" (Tellis, 1997). Darke, Shanks, and Broadbent (1998, p. 15) mention that "[w]eaknesses of case study research include difficulties in generalizing research results and the subjectivity of the data collection and analysis processes," Rowley (2002, p. 5) concedes that "[g]eneralisation can only be performed if the case study design been appropriately informed by theory, and can therefore be seen to add to the established theory. The method of generalisation for case studies is not statistical generalisation, but analytical generalisation in which a previously developed theory is used as a template with which to compare the empirical results of the case study." Four types of generalization are development of concepts, generation of theory, drawing of specific implications, and contribution of rich insight (Walsham, 1995). In defense of case studies, Runyan (1982, p. 6) admits that "[a]lthough the case study method can be, and too frequently is abused by being employed in arbitrary and indefensible ways, a quasijudicial or adversarial procedure can be used in which the evidence and arguments in case studies are subject to critical examination and reformulation." Despite the claims on the contrary, the author considers case studies a suitable method because they not only allow to analyze real-world scenarios but also provide background and logical sequence of steps, both of which establish research context.

Three research philosophies exist: critical, interpretive, and positivist. The research can be classified as critical “. . . if the main task is seen as being one of social critique, whereby the restrictive and alienating conditions of the status quo are brought to light. Critical research seeks to be emancipatory in that it aims to help eliminate the causes of unwarranted alienation and domination and thereby enhance the opportunities for realizing human potential. . .” (H. K. Klein & Myers, 1999, p. 3). The study is interpretive if “. . . if it is assumed that our knowledge of reality is gained only through social constructions such as a language, consciousness, shared meanings, documents, tools, and other artifacts. Interpretive research does not predefine dependent and independent variables, but focuses on complexity of human sense making as the situation emerges. . . it attempts to understand phenomena through the meanings that people assign to them. . .” (H. K. Klein & Myers, 1999, p. 3). It is positivist “. . . if there is evidence of formal propositions, quantifiable measures of variables, hypotheses testing, and the drawing of inferences about a phenomenon from a representative sample to a stated population. . .” (H. K. Klein & Myers, 1999, p. 3).

The case studies combine critical and positivist approaches: while hypotheses testing will not be extensively utilized, some variables are measurable and generalized inferences can be drawn to a population (case study 1). However, the case studies also analyze shortcomings in organizational ICT infrastructure which should be rectified to close the exploitable attack vectors and in so doing, resilience its increased. Both studies describe realistic conditions which bring about their own challenges: “Conducting research on real world issues implies a trade-off between level of control and degree of realism. The realistic situation is often complex and non-deterministic, which hinders the understanding of what is happening, especially for studies with explanatory purposes. On the the other hand, increasing the control reduces the degree of realism. . . Case studies are by definition conducted in real world settings, and thus have a high degree of realism, mostly at the expense of the level of control” (Runeson & Höst, 2009, pp. 5–6). Not all variables in the test models can be controlled which may open new avenues to pursue.

Various software will be used throughout and while care was taken to ensure the up-to-date versions were employed at the time of writing, future releases may render some features obsolete or introduce others which will optimize performance, reducing the work factor involved. The reader is advised that while some operations will remain unchanged, those tied to particular version could become archaic. The case studies thus serve as a historic snapshot which itself could constitute a future research direction: a comparative study on how the tools and procedures have evolved.

The author is convinced case studies are adequate for communicating how basic and advanced security measures can be circumvented by adversary lacking theoretical knowledge to carry out the attacks without automated tools. While the damage may be marginal, when ICT design and policies are incapable to mitigate low-priority threats, they cannot be expected to do so in case a sophisticated incursion attempt is mounted. Script kiddies present an opportunity to establish a lower bound on infrastructural capabilities with respect to security engineering. For example: when methods exploiting known vulnerabilities have been found ineffective, unskilled attackers are likely to deprioritize further malicious actions due to number of other device classes (personal computers, embedded and mobile devices, server front-ends) accessible over the Internet. Determined perpetrators who target the organization will, however, necessitate active deterrents in place as they do not aim to penetrate systems indiscriminately, but rather focus on specific victims. This establishes upper bound on security considerations. Malware infections, insider threats, and script kiddies constitute more realistic scenarios than being targeted by skilled adversaries. Therefore, such scenarios will not be analyzed and priority instead given to threats commonly encountered in the Internet-connected settings.

Note: throughout the case studies, plural addressing “we” instead of singular “I” is preferred.

5.1 Case Study 1: Reverse Password Engineering

Reverse password engineering (password cracking) refers to “...the process of getting the plaintext passwords from the stored secrets, or at least an equivalent one, which collides with respect to the hash function used, with the user’s one... In order to maximize cracking ability and efficiency, the cracking process should be sped up as much as possible, and the candidate password selection should be as smart as possible” (Marechal, 2008, p. 1). In an offline cracking scenario presented here, “...the attacker can make a number of guesses bounded only by the time and resources she is willing to invest” (Dürmuth, Chaabane, Perito, & Castelluccia, 2013, p. 2).

The case study will demonstrate a complete pipeline of how an attacker can reverse engineer misappropriated data stored in back-end databases connected to vulnerable front-ends accessible on the Internet, and available to any user. While administrators implicitly presuppose the majority of interactions are harmless, precautions are taken to prevent unauthorized access. Rapid release schedules on one hand offset by the need to thoroughly test each version before deployment into production environment on the other, incompatibilities introduced in non-incremental revisions which prevent timely adoption, and strong preference for maintaining stability instead of ensuring up-to-date state all contribute to software with multiple exploitable attack vectors. These vulnerabilities are usually closed in subsequent iterations but present in older ones. Breaking compatibility with third-party modules is especially aggravating. Akin to bottlenecks presented in Figure 2, lack of backward support can negatively affect business performance in case dependencies exist with other parts of the software base, a situation demonstrated in Figure 67.

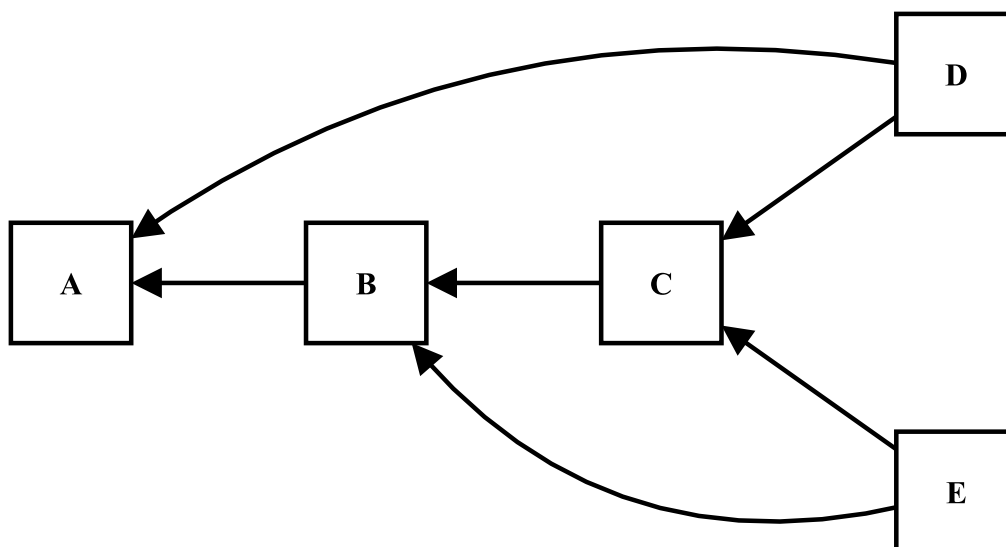


Fig. 67: *Third-party software module dependency violation. Patch management process should map all ties which may cause stability or performance issues. Here, the organization controls software (A) and third-party module (C), but dependencies (D, E) require particular version of B outside the scope of patch management policy to be present.*

Source: own work.

To show the dependency tree for the simple structure, a truth table can be constructed to show all possible combinations of states. In logic, a truth table “... shows the way in which the truth value of a sentence is built up using truth-functional connectives depends on the truth values of the sentence’s components” (S. M. Cohen, 2004, p. 10). Let 0 denote a scenario where

the software is not updated (status quo), and 1 a scenario where a newer version is introduced. A total of $2^5 = 32$ combinations will have to be considered to cover the complete probability space, i.e., every situation which may occur. Two edge cases may arise: preserving status quo (first line in the table in Appendix B) and full software update (last line), both of which do not break compatibility assuming the system was stable initially and the upgrade did not generate additional errors.

The table indicates that apart from the two edge cases, 5 situations lead to unconditional system stability. Let us present a sample case to determine the yes/no output for line 4 (A: 0, B: 0, C: 0, D: 1, E: 1) which evaluates to “Yes.” Checking the dependency tree in Figure 67, the following sequence is valid:

- A, B, and C are left unchanged, A–B and B–C pairs do not flip their states and therefore preserve compatibility,
- D is updated; it depends on A and C with neither of them changed, A–D and C–D are stable as long as new D is stable,
- E is updated; it depends on B and C with neither of them changed, B–E and C–E are stable as long as new E is stable,
- the system is considered stable.

With n dedicated software components, the combinations are calculated using 2^n which grows exponentially even for small numbers. This was previously demonstrated in Table 5 in chapter 2.4.3. Note that even for $n = 10$, a value not uncommon in practice, the patch management policy must cover the 1 024 possible situations based on deployment priorities to ensure stability while maintaining security. A reasonable criterion is linking the priorities to a vulnerability class (critical, high, medium, low) and number of discovered and disclosed exploits. That way, programs exhibiting the highest proportion or severity of known vulnerabilities are prioritized.

The probability of system stability decreases with higher n subject to exponential decay, i.e., if no coordinated plan has been devised, at least some instabilities can be expected when updating. Hypothesizing further, many organizations will strongly prefer stability over security, especially for mission-critical ICT systems. In case study 2, this will be leveraged during the security audit to exploit vulnerabilities in older and out-of-support software, if it is found.

5.1.1 Phase 1: Background, Methodology

Before commencing, a plan which contains a sequence of reproducible steps the case study will follow has to be formulated, and implications of the research assessed due to ethical and legal considerations involved.

The question the case study aims to answer is: **How effectively can attacker possessing low to moderate ICT security knowledge reverse engineer password digests using trivial methods with pre-defined rules?** It will do so by establishing causal links between operations which will lead to a list of real-world plaintext passwords many of which are hypothesized (as per results in chapter 4.2.1) to be reused across multiple sites. The question above states “How effectively. . .” which inherently represents a biased construction; interest is put not on if there is a chance the adversary (modeled in chapter 2.4.1) is able to get the information, but how effective she is in procuring plaintext, human-readable strings from hashes. The case study will use only freely-available software, a condition not indicative of real-world situations where the attacker can obtain any tool she wishes by downloading it illegally. Violating intellectual property laws is of little concern to a party who decided to conduct a criminal offense by accessing information

systems in an unauthorized manner with the intent to misappropriate sensitive data and use them to perform unsanctioned actions. Restricting the case study to such programs will not, however, influence results as the breadth of functions is satisfactory even in the selected software pool.

Case study stakeholders are IT managers, users, and system administrators. A stakeholder is a concept derived from organizational theory and business ethics and involves “[a]ny group or individual who can affect or is affected by the achievement of the firm’s objectives... Each of these groups plays a vital role in the success and of the business enterprise... Each of these groups has a stake in the modern corporation, hence, the term...” (Freeman, 2010, p. 25). IT managers in conjunction with system administrators devise policies which prioritize ICT infrastructure components to be upgraded. Case study 2 will show that security auditing and penetration testing are crucial for assessing weaknesses exploitable by anyone with limited theoretical background, but with access to the right tools. Despite being relegated as low-risk, systems are frequently vulnerable to basic exploitation techniques and pose a threat to data confidentiality, integrity, and availability (chapter 2.2). Users are at risk as all information entered in good faith and assumed adequately protected are within easy reach: when stolen, the asset is dependent solely on the encryption scheme employed to withstand offline attack scenarios. In case of substandard configuration or shortcuts taken during deployment, the data (Table 8) will be reverted in reasonable time. The outcome is augmented as users habitually select weak passwords which do not pose any challenge to parallel computations within reach of any perpetrator. Strong and unique strings partially alleviate the weak-hash problem: even with massive computational resources available and an easy-to-reverse hash function, the attacker still needs to correctly identify the original string whose length and complexity are significant obstacles to overcome. With schemes such as bcrypt, PBKDF2, and scrypt (chapter 6), an arbitrary work factor can be set to force multiple iterations in one pass to produce a single valid attempt. This considerably reduces the effectiveness of reverse password engineering.

The topic is of interest because it demonstrates how low the barriers to entry are with automated procedures taking over many technical aspects, and how high the success rates can be. No custom hardware is necessary; in fact, any computer capable of running a modern operating system and a browser (including portable mobile devices, i.e., smartphones and tables as detailed in chapter 2.3) is fit to carry out all the attacks described below, substantially increasing the pool of potential adversaries. When neither hardware, software, nor lacking technical background constitute a limiting factor, protecting sensitive data needs to become a priority.

Microsoft Windows operating system will be used. The choice may be controversial because alternative, highly-specialized Linux distributions designed for security-related research are freely available. The rationale is based on adversary profiling: considering her low skills, maximum returns on minimum time investment mindset, identical tools for both platforms, little assumed difference in performance as well as propensity toward Graphical User Interface (GUI) rather than Command Line Interface (CLI) makes Microsoft Windows an obvious contender. Despite its closed-source nature, limited modifiability and flexibility together with a commercial licensing model, it consistently maintains top market share in the desktop segment (chapter 4.2.1).

Reverse engineering of the hashes will be carried out in Hashcat-GUI, a graphical front-end to three utilities: hashcat, oclHashcat-plus, and oclHashcat-lite. Compared to John the Ripper, another popular password cracker, Hashcat-GUI simplifies the process of loading source files, setting up rules and word lists for dictionary attacks, and running the computations by providing visual cues and help. John the Ripper is purely CLI-based and requires familiarity with switches which the perpetrator has to have for penetration testing (chapter 5.2), but which are not necessary here due to intuitive GUI.

In case of multiple software versions, stable snapshots were preferred over beta and development ones which may be unstable or exhibit irreproducible behavior. This conservative approach is in line with building a dependable program base with which to carry out the attacks. To summarize, the software used will be:

- OS: Windows 7 Professional Service Pack 1 64-bit,
- password cracker: Hashcat-GUI v0.301, hashcat v0.46, oclHashcat-plus v0.15, oclHashcat-lite v0.15,
- statistical software: R 3.0.2 32-bit,
- text editors: Vim 7.4, Notepad++ 6.5,
- additional: .NET Framework 4.

The case study will use commercial off-the-shelf (COTS) hardware setup to conduct the experiments. Reverse password engineering benefits from modern components, in particular GPUs (graphics processing units) capable to execute basic mathematical operations in parallel comparatively faster than CPUs. However, the adversary is assumed to have access only to integrated hardware, i.e., portable computers, no desktop stations or cloud instances which can be scaled to meet her demands. Dedicated parties leverage stations with GPU arrays which distribute workload units dynamically and ensure high utilization factors. As per Moore's law (discussed in chapter 2.2.3), transistor-equivalent integration in electronic circuits is expected to increase over time, and with associated cost savings the prices of hardware are expected to decrease. This makes building small-scale hardware farms financially justifiable which necessitates system administrators to implement data encryption schemes reflecting gradual increases in performance. The test setup is a full-size laptop with the following specifications (some not relevant to the case study, included for completeness):

- CPU: Intel Pentium T 4400 Penryn Dual-Core 2.20GHz,
- cache: L1 data 2×32 kB, L1 instruction 2×32 kB, L2 32kB,
- instruction set: MMX, SSE, SSE2, SSE3, SSSE3, EM64T,
- RAM: 4GB DDR3 SDRAM Dual-channel 399MHz,
- GPU: ATI/AMD Mobility Radeon HD 4570 Series 512MB PCI-Express v1.1 x16,
- HDD: TOSHIBA MK5055GSXN 500GB SATA 5400RPM,
- HDD cache buffer size: 8192kB.

All components perform at stock parameters, no overclocking was applied to maintain stability, prevent overheating and irreproducible behavior. The setup represents inexpensive, affordable option with CPU, HDD, and GPU all bottlenecks to higher password-cracking performance. Of note is the hard disk drive: mechanical parts introduce non-negligible delay during read/write operations which increases with the amount data to be accessed. The solution is SSD (solid-state drive), a component replacing moving components with memory chips that allow uniform access times to every location. Replacing CPU and GPU with newer models boosts speed due to higher frequencies, tighter integration, optimized instruction sets for mathematical operations, and efficient multi-core and multi-threaded design.

Legal disclaimer in chapter 3 applies to the case study but ethical considerations must be also mentioned. The Association for Computing Machinery (ACM), a non-profit organization for scientific and educational advanced in computing, published Code of Ethics and Professional Conduct which states that it is prohibited to use "... computing technology in ways that result in harm to any of the following: users, the general public, employees, employers... [P]ersonal information gathered for a specific purpose not be used for other purposes without consent of the individual(s). These principles apply to electronic communications... and prohibit procedures that capture or monitor electronic user data, including messages, without the permission of

users or bona fide authorization related to system operation and maintenance” (ACM, 2013). The Institute of Electrical and Electronic Engineers (IEEE) released a similar document which prohibits “to avoid injuring others, their property, reputation, or employment by false or malicious action” (IEEE, 2013).

In the study, personally-identifiable data (login credentials, passwords) will be analyzed which may raise concerns and reasonable objections. However, they will not be used in ways which would negatively affect users. The data will serve as input to analyses, and at no point will login be attempted with the passwords extracted. Furthermore, screen shots will be sanitized to mitigate malicious actions on part of the reader, as specified in chapter 3. After the research is concluded, all electronic assets will be wiped from the hard drive, cloud storage, and removable media which will effectively render them unavailable. No identifiable information pertaining to individuals will be released. Aggregated statistics will be preferred to honor confidentiality, and in case of examples, context will be omitted. This will turn the passwords into strings, not actionable intelligence.

5.1.2 Phase 2: Analyzing the Data Set and the Tools

Regardless of how the password hashes were obtained, the adversary has them in her possession and the target lost exclusive control over the data. We will now focus on ways to procure cleartext passwords by means of reverse engineering. Chapter 2.2.2 dealt with cryptographic hash functions in more detail, a short overview will be provided here without references.

A cryptographic hash function is a mathematical construct which compresses input bits in a lossy manner, not preserving information content of the original but providing a computationally-efficient procedure to generate output which then serves as a comparison baseline. Supplying a product of identical function, we can conclude whether the original data were the same or different with overwhelming probability. A situation may occur, though, which sees two or several unrelated inputs generating equivalent hashes, known as a collision. Collisions cannot be ruled out in any function with serious negative results: the adversary can use distinct data, e.g., passwords to impersonate legitimate user and instigate identity theft based on hash indistinguishability. As opposed to “normal” hash functions where the probability is much higher, cryptographic alternatives expand the output to include more bits which increases the computational time to produce a collision. For a probability of getting k unique values in a full set of N , the following expression holds:

$$W = \frac{N-1}{N} \times \frac{N-2}{N} \times \frac{N-3}{N} \times \dots \times \frac{N-(k-2)}{N} \times \frac{N-(k-1)}{N} \quad (5.1.1)$$

We simply reduce the space of all values by subtracting each unique one until we are left with the last one which is unique as well. A complementary event, $1 - W$, denotes the probability of at least one hash being identical. N is the total number of hash products, and is directly related to number of bits: 128 for MD5 and 160 for SHA-1, the two popular hash functions in production environments. Despite neither suitable for storing passwords due to their speed which allows generating and comparing millions of candidates per second, they are nevertheless still used. Number-wise, SHA-1 is considered a better alternative than MD5 but still fares worse than mathematical functions specifically designed to perform slowly by requiring multiple thousand iterations, fixed or variable amount of RAM.

Advances in CPU and GPU design results in performance of at least 10^9 hashes per second on a single unit. Parallelizing the workload across multiple physical or logical instances also reduces

the time factor involved. By employing dictionaries, the search space is pruned to a subset of the most probable strings. The trade-off is acceptable when word lists incorporate strings from database breaches which can be mutated for higher efficiency. A comprehensive mutator list was presented in Table 6, and the majority of rules mentioned when creating secure passwords can be defeated by simply invoking a command line switch or a checkbox in GUI.

A source of hashed credentials will have to be supplied for analyses, a real-world database with sufficient amount of data from which reliable conclusions can be drawn. After weighing multiple options, a database of hashed passwords was picked representing a June 26, 2011 usernames and passwords leak from an action video game titled Battlefield Heroes, developed and published by Electronic Arts. A single file was made available for download and testing with the following properties:

- name: Battlefield Heroes Beta (550k users),
- format: Comma-separated values (.csv),
- size: 25,869,892 bytes,
- columns: 2,
- rows: 548,773.

A .csv is a file structure for storing database records in textual form. Each data element is separated from others by a delimiter, e.g., colon, semicolon, space, or tabulator while records are separated by a newline character. Many tools support .csv importing even though no standard has been enforced and implementations differ, but compatibility is not an issue and the format recommended for portability. Unlike proprietary alternatives, .csv is vendor-agnostic. Passwords in the file were hashed using MD5 without salt (chapter 2.4.2), which will simplify the reversing process considerably. Usernames in the first column were stored in human-readable form, turning the hashes into a single line of defense. Had the operators encrypted them along with the passwords, the adversary would have to crack both to get actionable data. More than half a million records in total with each representing a user who created an account were stolen during the attack. This suggests either SQL injection (chapter 2.4.7) or similar technique which allows batch access to credentials, although this information is speculative and not based on known facts. The first 20 lines are shown in Figure 68. Usernames were blacked out, a precaution explained in chapter 3.

The data set will be fed into Hashcat-GUI and various reverse engineering techniques run to determine the time to obtain plaintext information which can be exploited to surreptitiously gain access to tens of thousands user accounts, or integrate the passwords into word lists which could then be plugged in for increased efficiency during future cracking attempts. The window of opportunity is wide because some users do not change their passwords regularly (chapter 4.2.1), some decided to abandon their accounts without purging them of personal details, and others forgot their login credentials. In the last two scenarios, the database stores the data after they became irrelevant, leaving a non-obvious, dormant attack vector open.

We should examine the data set for erroneous entries and remove them. Errors can be introduced in many ways: improper batch processing, database corruptions, accidental overwriting, deliberate attempts to hamper integrity (chapter 2.2.2 mentions checksums to mitigate the risk), unfinished or conflicting database transactions which left it in a state of flux and DBMS could not resolve the situation, etc. While the reasons vary, we will focus on how to automatically detect and delete the malformed strings using regular expressions.

Regular expressions are understood as "... the key to powerful, flexible, and efficient text processing. Regular expressions themselves, with a general pattern notation almost like a mini programming language, allow you to describe and parse text. With additional support provided

1		cbae07efa0c6ed330a283e80a9c02e8d
2		252b4927a2811f5bd1c38b2e270cc95c
3		7230a073e5a694275e7906a470d5bba2
4		1d285891d61c59b9b108db77f93d4707
5		e34f2d044c7fc5414ab8fd337bae66e2
6		2562047199bab3e82000e69e6b9a3068
7		33247d3fb6efcf8cc3c906236bb15db2
8		0b541cf54395a1f94383a3cc34ec4ac4
9		275334c41e8d33a28d90c83e05037f3b
10		1ee8c585b91199f3e7fd8f5f367d7616
11		96e0fcd205940c2fbc242a1977cee005
12		c779a945f6b0c752703ced0fd98fd131
13		52ef8a66894a50ddd14efa45c52a2114
14		cdd64570cc852ff155dfdb467a495f88
15		592db47d5ccc62700d6dfc9f16c1f862
16		0358083f16138f6379a7dedd7501476b
17		b352c30a4f4b77b59ec89e5d0a1c71a6
18		0733601d2e5dd5937a0ac3053b8200c0
19		686a38ae699a2a0fc4f176206996fab
20		3d236013281409ad180c41652697487c

Fig. 68: Data set structure. Each line represents a user and their password as a 128-bit MD5 hash value with no salt added for security.

Source: Own work.

by the particular tool being used, regular expressions can add, remove, isolate, and generally fold, spindle, and mutilate all kinds of text and data” (Friedl, 2002, p. 1). A comprehensive tutorial on regular expressions is not within the case study’s scope. For MD5, the number of alphanumeric hexadecimal characters comprising the hash is 32, and all strings should have equal length. Any sequence with less or more than 32 characters has to be discarded because Hashcat-GUI will otherwise throw an exception. Notepad++, an open-source text editor which supports searching and replacing by regular expressions, was selected as a suitable tool. Afterwards, the transformed file consists only of properly-formatted hashes exactly 32 characters in length, and usernames. Hashcat-GUI will identify and skip them because they do not represent any known cryptographic function.

Before we delve into Hashcat-GUI, word lists for dictionary attacks will be mentioned. Even though it would be possible to just run the program, load the hashes, choose a mode, and launch the process, options would be limited to variations of brute-force attack which belongs to a category whose output grows exponentially even for small input sizes, demonstrated in Table 5 in chapter 2.4.3. In practice, reverse engineering longer passwords takes prohibitive amount of time which threatens to close attacker’s window of opportunity which may range from hours to days or weeks. In either case, brute-force approach cannot be relied on for longer strings. It will be used for the 1–6 character space so that the pool of shorter passwords is tested exhaustively. While slow, it guarantees all character sets will be included in all possible combinations up to and including the upper bound. This pertains to CPU-bound cracking: physical or virtualized GPU hardware can enumerate passwords up to length 8 and further advances are inevitable.

Word lists can be sourced from passwords uncovered from previous database leaks, social networks, encyclopedias, movie scripts, lyrics, books, titles, names, and predictable combinations. Many websites offer free downloads of vast collections organized into categories, and even specific to geographic locations: word lists for Chinese and American users will reflect cultural, political, and social realities of the particular country. We will use a collection of real-world passwords to test hypothesis that users’ choice is governed not by security but ease of use and comfort. Of all the options, the following word list was selected:

- name: realhuman_phill,
- format: text file (.txt),
- size: 716441107 bytes,
- columns: 1,
- rows: 63941069.

The dictionary was downloaded from Crackstation¹ and is explicitly distributed under Creative Commons Attribution-ShareAlike 3.0 license which allows to share and adapt the work with attribution. When instructed to open it, Notepad++ threw an error about the file being too big and halted the operation. An alternative text editor, Vim, had to be run to determine the row count. Hashcat-GUI will manipulate the word list in parts depending on the amount of RAM at its disposal, an adjustable parameter. With almost 64 million unique passwords and many more constructed by intra- and inter-word mutations, the collection is suitable for inferring representative conclusions about the data set. Higher success rate compared to naïve brute-force attack is a reasonable expectation because dictionary attack focuses on picking the most probable candidates as opposed to testing for all, even highly unlikely sequences. Nevertheless, we will use both and juxtapose the results.

Hashcat-GUI will be our reverse engineering tool of choice. After downloading and extracting the archive, user must provide paths to command-line versions of Hashcat, oclHashcat-plus, and oclHashcat-lite. The first harnesses CPU, the other two require compatible GPUs from NVIDIA and AMD technology companies. User interface is depicted in Figure 69.

Starting from left to right, the screen shot shows three tabs, each with settings and the ability to execute the test scenario using the respective tool. Until all prerequisites are met, the launch button is grayed out, indicating that additional input is necessary. The fourth tab, “Wordlists & Markov,” loads up dictionary files and Markov chaining rules, “Commands” exports commands from the current session to a portable text format; pasting the text back, user can run an identical test scenario. “Help” and “About” list basic syntactic rules for password masks and credits, respectively. The last tab, “Distributed,” incorporates the client identified by username and password into a cluster for distributed processing.

Markov chains mentioned in the previous paragraph seem to be a promising research avenue in linguistics and reverse password engineering. In its simplest form, A Markov chain denotes a process whose property is “. . . that it retains *no memory* of where it has been in the past. This means that only the current state of the process can influence where it goes next. . . . [When] concerned exclusively with the case where the process can assume only finite or countable number of states, . . . it is usual to refer it as a *Markov chain*. . . . What makes them important is that not only do Markov chains model many phenomena of interest, but also the lack of memory property makes it possible to predict how a Markov chain may behave, and to compute probabilities and expected values which quantify that behaviour” (Norris, 1998, p. 1). Probabilistic Context-Free Grammars mentioned in chapter 2.4.3 are based on Markov chains. Hashcat-GUI offers facilities to automate Markov chain rule mapping: a word corpus is analyzed and probability tables for each n -gram where $n = 0, 1$ created, i.e., rank-ordering the characters according to position while taking into account which character came before. The rules do not rely on (possibly skewed) collection of known strings as a discriminator (dictionary attack) and are not as resource-demanding as brute-force enumeration despite possibly covering the whole search space. Mathematical background for Markov chains is left out since it does not directly relate to the thesis’ focus, and is hypothesized to be beyond the capabilities of a low-skilled adversary.

¹<https://crackstation.net/buy-crackstation-wordlist-password-cracking-dictionary.htm>

5.1.3 Phase 3: Brute-Force and Dictionary Attacks

Scientific experiments must follow certain rules for the results to be objective, reproducible, and verifiable. Special attention will therefore be paid to describing each link in the process of obtaining plaintext passwords via two main attack modes: brute-force and dictionary. Even though some variables can be controlled and fixed, computer systems are complex structures, and some variables, namely performance fluctuations and factors influencing peak throughput values, will vary. Just gathering performance metrics affects the system, much like the role of observer in second-order cybernetics mentioned in chapter 2.1.1. Eliminating the influence would mean forfeiting objective, factual results for subjective, perceptual ones. A simplified sequence which the computer goes through when Hashcat-GUI is started is as follows:

- a 64-bit CLI is invoked from the GUI and all the parameters passed as switches,
- resources (CPU, RAM) are requested and granted from the scheduler,
- reverse engineering starts and continues until the search space is exhausted or the process is aborted by the user.

The first two steps are performed in less than a second while the third one may last for days in case of brute-force attack, depending on hardware specifications. We opted to let Hashcat-GUI run until all combinations of character sets were tested to assure passwords of lengths 1–6 were recovered. Various dictionary attack modes will then be employed for the remaining hashes. A 64-bit Hashcat build, `hashcat-cli64.exe` will be linked to the GUI but no speedups are expected as the advantage of running 64-bit applications is access to more than 4 GB of RAM per process. The hardware setup has exactly this memory capacity but the operating system reserves a part of it for active processes. We assume the difference in performance between 32-bit and 64-bit software to be negligible for the testing purposes. The setup step is depicted in Figure 69.

“Remove found hashes” deletes the strings for which plaintext counterparts have been found, leaving only those which were not identified. The modified file will then be inputted to the dictionary attack mode. Executing the task by clicking “Hash me, I’m a digest” brings up a terminal window with the CLI version of Hashcat whose path was specified in the GUI. The program has several switches to specify character sets to be iterated through during the brute-force attack; those can be combined or used separately:

- ?l: lowercase letters,
- ?u: uppercase letters,
- ?d: numbers,
- ?s: special characters,
- ?a: ?l?u?d?s,
- ?h: 8-bit characters from 0xC0 – 0xFF,
- ?D: 8-bit characters from German alphabet,
- ?F: 8-bit characters from French alphabet,
- ?R: 8-bit characters from Russian alphabet.

In the study, we restrict ourselves to the first 5 switches. This may favor users who selected German, French, or Russian glyphs but we do not aim to achieve 100% success rate. Rather, we will demonstrate how effective even basic password cracking techniques still are in real-world situations. In Figure 69, the “Mix Alpha Numeric Symbol” option is active which corresponds to the ?a switch. A combinatorial problem, the total number of unordered sets is calculated as $K(k, n) = \frac{n!}{k! \times (n-k)!}$ for $k = 1, 2, 3, 4; n = 4$. There are 15 combinations of switches. Note that this is a symbolic notation and during reversing, set order is important. Figure 70 shows the terminal window after the task was started.

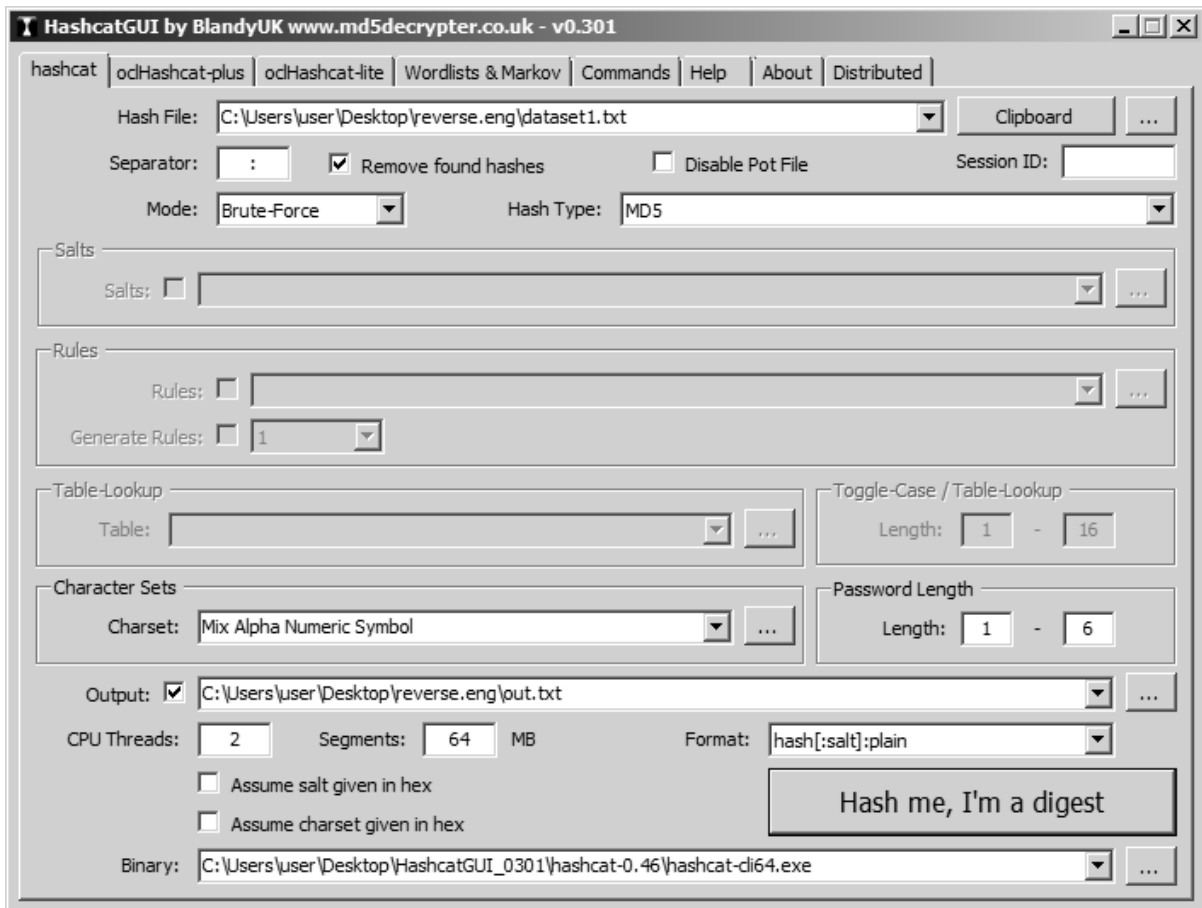


Fig. 69: Hashcat-GUI test setup. Data set source and brute-force attack mode are selected along with a mix of alphanumeric characters and symbols in all combinations from 1 to 6. Output plaintext passwords will be piped to a text file.
Source: Own work.

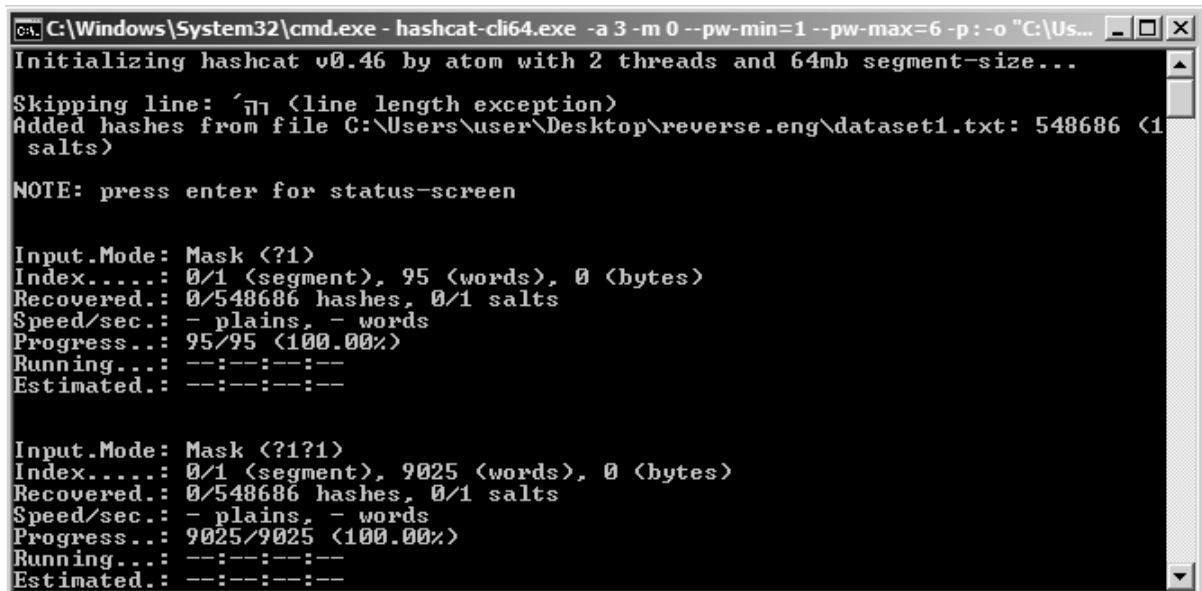


Fig. 70: Hashcat-GUI command line interface. The window title lists switches and parameters set in the GUI which can also be invoked from the CLI.
Source: Own work.

Hashcat was started on October 18, 2013 at 12:00:00 on both physical cores. At this point, it became self-sufficient and did not require any user input. A terminal window displays character sets completely iterated through, and current progress could be queried at any time by pressing Enter. Figure 70 further shows the following statistics:

- Input.Mode: a mask specified by the switches mentioned above,
- Index: segment shows how much RAM Hashcat reserves and uses to store word list data (0 for brute-force attack); words denote the candidate strings pool size for the current mask,
- Recovered: number of hashes and salts successfully reverse engineered in the particular step,
- Speed/sec: strings tested per second from the mask (plains) and from the dictionary (words),
- Progress: absolute and relative measures of what proportion of the search space has been covered,
- Running: elapsed time,
- Estimated: time until the candidate pool is exhausted, calculated dynamically based on available resources.

Several variations of dictionary attack were implemented which increase the candidate pool size while keeping it smaller than for brute-force enumeration. The modes are selected from a drop-down menu titled “Mode” and require at least one word list loaded on the “Wordlists & Markov” tab. Figure 71 depicts the setup before the first test was commenced.

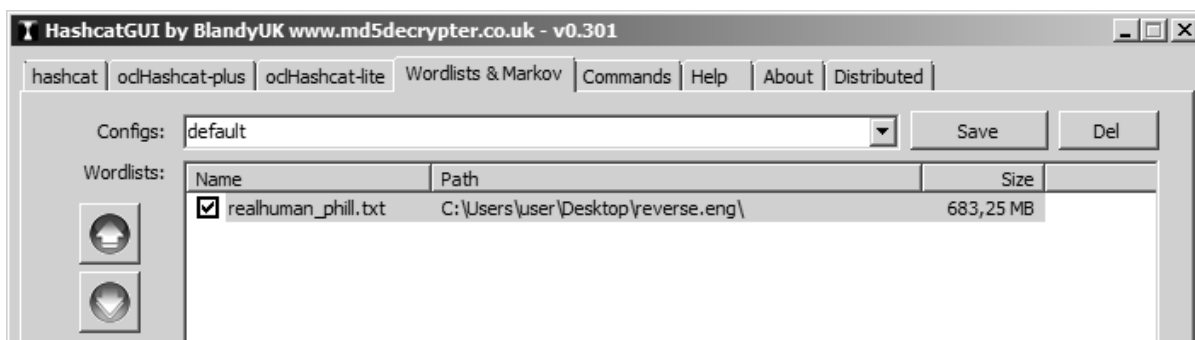


Fig. 71: Hashcat-GUI word list management. The file is active, indicated by the check box, and if bigger than the segment size specified on the “hashcat” tab, will be split into chunks each of which will be processed separately. Segment size is a trade-off between resource utilization and speed.

Source: Own work.

Straight instructs the program to take each string from the word list, hash it into a digest of user’s choice, and compare the value with the target fingerprint. If a match is made, plaintext versions of both are equal with overwhelming probability; otherwise, a collision has been found which sees two distinct inputs produce identical output. Optionally, a .rule file can be supplied with mutators for the word list entries, or randomly-generated rules (1, 10, 100, 1 k, 10 k, 100 k, 1 m) can be employed. The higher the parameter, the longer it takes to iterate through the whole set, greatly increasing the time factor. Even though the case study was not limited by time, simulating a real-world scenario imposes a constraint on how long the reverse engineering process can last before the window of opportunity is closed. To balance search space coverage and speed, 100 k was deemed optimal.

Combination allows two separate word lists to be loaded and all possible combinations generated from both. Even though we have a single source, we can still use the mode by simply pointing the program to the same file, indicating we want to create combinations between entries

in the same collection. However, it is reasonable to expect performance will exhibit strong negative correlation with sample size because $n!$ combinations will need to be addressed to exhaustively map all 2-tuples. On the other hand, users may repeat the same word to artificially boost password length under the assumption length achieved in this way increases security, but complexity and uniqueness are both low. Combination mode was ruled out as an option for the case study due to extensive word list: it would take more than two months of computational time to cycle through all candidates.

Toggle-case operates with one list, and for each entry generates all combinations of lowercase and uppercase letters. The first method in Table 6 (case mutation) follows the same logic. Optional configuration involves fine-tuning two threshold parameters: minimum and maximum length of an entry to be included and modified. Toggling cases from lowercase to uppercase and vice versa even for short strings expands the search space considerably. A total of $\frac{n!}{k! \times (n-k)!}$ mutations will have to be created, where n denotes the number of letters in the original string, and k the number of toggles in each iteration. It follows that longer passwords containing 30 letters, not unheard of in high-privileged login credentials, will require hundreds of passes for a single template. The thresholds are cut-off lengths: any word whose length is above or below the set value will be disregarded and only tested using straight mode. This prevents longer strings from slowing the reverse engineering process down. Therefore, 6+ character passwords with some letters switched to uppercase pose a challenge for all the methods presented so far: brute-force cannot enumerate longer strings, straight mode will not match the string unless an identical one is found in the word list, combination works with two words only, and toggle-case needs to have thresholds set which trade length for speed. One way to reverse such a hash would be for the password to combine two basic strings so that combination mode would generate identical output, but toggling case on non-obvious positions (i.e., middle, not the beginning) decreases the probability of success.

Permutation mingles the characters inside the word by switching their positions. Here, permutation is understood as “[t]he act of changing the order of elements arranged in a particular order, as abc into acb, bac, etc., or of arranging a number of elements in groups made up of equal numbers of the elements in different orders, as a and b in ab and ba; a one-to-one transformation of a set with a finite number of elements” (Fuchs, 2009). For longer passwords, the mode is expected to perform on par with toggle-case mode, necessitating threshold parameters to limit focus on passwords of fixed lengths because $n!$ permutations are outputted, where n is the string size.

Other modes (Table-Lookup and Mask) will not be considered as they require understanding of Markov chains, pattern matching, rule extraction, and Linux operating system commands which provide facilities to clean and optimize lexical structures in word lists. These make cracking more efficient with less hardware resources spent on storage and pre-processing. The model adversary does not possess such knowledge and is motivated to maximize her gain as quickly as possible after breaching the database.

5.1.4 Phase 4: Results, Conclusion

After all modes described in the previous section were cycled through, the hashes remaining in the data set represented passwords with the following properties:

- approximately balanced length, complexity, and uniqueness,
- longer than 6 characters, or shorter but with special characters included,
- longer than any threshold parameter set,

- not included in the word list,
- not conforming to any rule Hashcat-GUI applied,

To uncover the original strings, more sophisticated rules would have to be generated, additional word lists loaded, Markov chains generated, or hardware employed which would increase performance. Combining the suggestions, we would likely see even more hashes cracked, though the low-skilled attacker would gain access to thousands of accounts using the strings uncovered in the case study. It is important to note again that a retrieved password could be exploited not just to impersonate legitimate users on the site whose database was breached, but on other popular sites as well, e.g., email providers, electronic retail outlets, social networks, and payment processors. The net effect is thus broader and goes beyond a single victim. Assuming a success rate of 10 %, i.e., 10 % of affected individuals reused their credentials, the lower bound on the number of unique compromise user accounts is conservatively estimated at 10000. In the questionnaire research (chapter 4.2.1), 50.4 % of users admitted reusing a single passwords on multiple sites.

Resource utilization has not been tracked. The process was allocated around 90% of CPU cycles and around 385 MB RAM almost constantly and no significant difference was observed for brute-force and dictionary attacks. Memory consumption was managed by segment size: Hashcat-GUI loaded strings into memory for faster access which minimized HDD swapping and delays caused by high seek times.

We start by analyzing **brute-force attack results**. An overview is provided in Table 9. The output text file size was 2633440 bytes, a sample of 20 MD5 hashes with corresponding passwords is depicted in Figure 72.

Tab. 9: *Brute-force attack summary. The process was stopped after two hours and resumed afterwards due to power management issues. Total time excludes the inactivity and includes only the computation period. CEST: Central European Summer Time.*

Source: Own work.

Started on	Friday October 18, 2013 12:00:00 CEST
Finished on	Friday October 18, 2013 14:45:00 CEST
Started on	Monday October 21, 2013 07:45:00 CEST
Finished on	Tuesday October 22, 2013 11:45:16 CEST
Total time	00:31:01:16 (dd:hh:mm:ss)
Hashes total	548686
Hashes recovered	65836 (11.9988 %)
Hashes remaining	482850

Thorough password analysis will not be undertaken as the purpose of the case study was to ascertain whether user password selection process in a representative sample deviated significantly from previous research efforts, and the questionnaire research discussed in chapter 4. Proportion tests will be used for all modes in the data set: we are interested whether the individuals were more careful and diligent when choosing login credentials for entertainment purposes than is generally believed. Findings included in chapter 2.4.2 and many password policies suggest minimum length of 8 characters to thwart brute-force enumeration; assuming users favor comfort over security, the threshold is probably understood as a maximum rather than a lower bound. Even a single consumer GPU can cycle through the 1–8 pool in reasonable time, and parallel computation decreases the period substantially. Recommended password length is therefore

```

1 0b4e7a0e5fe84ad35fb5f95b9ceeac79:aaaaaa
2 a6b3ab582986e8c9b29c79987b803108:laaaaa
3 1c1fbdd037fc5fda9db95dc6b62b3ecd:paaaaa
4 9b79c437f85f135945d15e310f87381d:olaaaa
5 cb6761494116df84b3debc67b0f3784c:btaaaa
6 1d2a121f7d455767c86ff906e698eadc:mwaaaa
7 5b842885be8780c70f7f67ca11421031:bbbaaa
8 672c9d1f8f936068a72ea78f11aabae7:owcaaa
9 264cbc754c86b0907a26e779d6e31617:2010VV
10 a04616e4c2ec48529baeff1478db0ff8:lodaaa
11 f5bb3f813595fa93b315aeb2df5c58d3:53gaaa
12 c7cb03a2a218de731e528ca0826cecf9:jahaaa
13 1c0a569d7d5777527c0510083b837284:ajajaa
14 ca25bad051675f466f5ab0d77feba81a:mikaaa
15 51b92d410f3fa70044f7ad9a31de4371:lolaaa
16 d47fc4942da4b7c842f67c13be92a03f:dimaaa
17 4086e264837b7b1528f168bf4f07c1ce:kanaaa
18 09b47acfc0a5cc52c54d3cc556afddfd:penaaa
19 a10451ac3df20b4313bdf23d064421e2:konaaa
20 99ebb8954fd6f24e869f0336afae5218:akoaaa

```

Fig. 72: MD5 brute-force attack plaintext passwords sample. Output does not contain duplicates because cryptographic hash functions are deterministic: outputs of two equal inputs will not differ, i.e., a cracked hash results in compromise of all identical hashes.
Source: Own work.

not a panacea and will likely constitute modal value found in the data set. For the data set, the parameter was set at 0.3: we suppose 30 % of users chose passwords which can be trivially retrieved using any attack mode presented in chapter 5.1.3.

The proportion test is of the following form: we will statistically validate if observed sample proportion obtained in the case study is significantly higher or lower than the expected proportion of 0.3. To do that, several prerequisites will have to met: calculating the sample proportion, stating the null (H_0) and the alternative (H_1) hypotheses, significance level, and a sample size for a fixed test power. First, we must determine which method to use: z-test, chi-squared test without Yates' correction for continuity, or Fisher's exact test. All three allow to compare proportions but only the second one does so without taking absolute values into account. Z-test is based on the assumption both random variables (password lengths) are continuous and sampled from Gaussian distribution, which is not met since they are discrete. Chi-squared test seems an adequate match for larger samples: a non-parametric method along with Fisher's exact test, it does not rely on any particular probability distribution. In our case, it is reasonable to assume Fisher's exact test for smaller samples would give almost identical results to chi-squared test but absolute values are required instead of proportions. The limitation can be overcome by a control group. Prerequisites for the chi-squared test are:

- null $H_0 : p_0 = p_1$, the two proportions do not differ significantly,
- alternative $H_1 : p_0 \neq p_1$, the two proportions differ significantly, i.e, one is higher or lower than the other,
- significance level: $\alpha = 0.01$,
- power: $1 - \beta = 0.99$.

A two-tailed alternative hypothesis was tested. Null and alternative hypotheses are defined as "... a statement of no effect or no difference. Since the statement of the research hypothesis generally predicts the presence of an effect or a difference with respect to whatever it is that is being studied, the null hypothesis will generally be a hypothesis that the researcher expects

to be rejected. The alternative hypothesis, on the other hand, represents a statistical statement indicating a presence of an effect or a difference. Since the research hypothesis typically predicts an effect or difference, the researcher generally expects the alternative hypothesis to be supported” (Sheskin, 2004, p. 54). Statistical significance (alternatively Type I error) denotes a probability of the test rejecting a valid null hypothesis, Type II error is a probability of failing to reject a false null hypothesis. Statistical power $1 - \beta$ denotes a probability of correctly rejecting a false null hypothesis; unlike α and β , we aim to maximize it. Type I and II error rates are inversely proportional, and decreasing one will increase the other unless a third variable is introduced: sample size. The relation is depicted in Figure 73. For the control group, a 2011 Sony Pictures password database leak analysis with 37,608 unique accounts was picked (T. Hunt, 2011a).

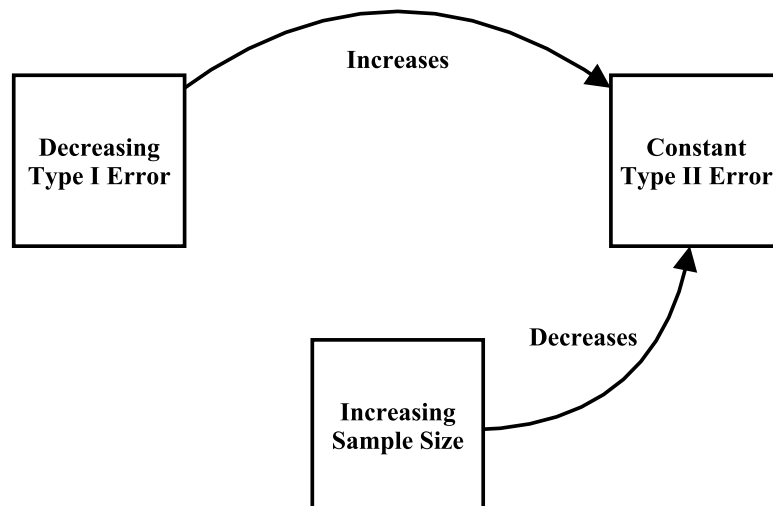


Fig. 73: Type I and Type II error dependence. To retain fixed β and thus $1 - \beta$, another variable must be tweaked which does not influence α but decreases β . Sample size exhibits such a feature and should be calculated before starting the experiment.

Source: Own work.

To see how large a sample is necessary to get $\beta = 0.01$ and $1 - \beta = 0.99$, the following command was entered to the R console:

```
power.prop.test(n=NULL, p1=0.3, p2=0.1199, sig.level=0.01, power=0.99,
  alternative = "two.sided", strict=TRUE)
```

Two-sample comparison of proportions power calculation

```

n = 240.0429
p1 = 0.3
p2 = 0.1199
sig.level = 0.01
power = 0.99
alternative = two.sided
  
```

NOTE: n is number in *each* group

The NULL parameter is a placeholder for determining the required sample size for the proportion test, p1 and p2 are proportions from both groups, $\alpha = 0.01$, and an alternative two-tailed non-directional hypothesis was selected. The last parameter, `strict=TRUE` signifies that “... the power will include the probability of rejection in the opposite direction of the true effect, in the

two-sided case. Without this the power will be half the significance level if the true difference is zero” (Dalgaard, 2012). There should be at least 240 observations per sample size for the parameters to hold. The condition was fulfilled. To determine the p-value, the lowest probability for which we can reject a null hypothesis, we entered:

```
x1<-c(11000)
n1<-c(37608)
prop.test(x=x1,n=n1,p=0.3,alternative="two.sided",correct=FALSE,conf.
  level=0.99)
```

1-sample proportions test without continuity correction

```
data: x1 out of n1, null probability 0.3
X-squared = 10.0979, df = 1, p-value = 0.001484
alternative hypothesis: true p is not equal to 0.3
99 percent confidence interval:
 0.2864857 0.2985694
sample estimates:
      p
0.292491
```

The first two inputs denote the number of passwords of length 1–6 and total number of strings, respectively; the former was inferred from a graphical bar plot and may be skewed. The proportion test for the Sony Pictures sample resulted in p-value of 0.001484, and not enough evidence was collected to reject the alternative hypothesis stating the proportion of insecure, trivially-retrievable passwords the users had selected significantly differ from the 30 % baseline. The claim holds with 1% probability of error. For our case study, the input was:

```
x2<-c(65836)
n2<-c(548686)
prop.test(x=x2,n=n2,p=0.3,alternative="two.sided",correct=FALSE,conf.
  level=0.99)
```

1-sample proportions test without continuity correction

```
data: x2 out of n2, null probability 0.3
X-squared = 84665.25, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.3
99 percent confidence interval:
 0.1188631 0.1211231
sample estimates:
      p
0.1199885
```

As evident, p-value is extremely low which suggests we are not entitled to reject the alternative hypothesis. In the data set, the proportion of passwords which can be reverse engineered in about 31 hours on commercial hardware utilizing brute-force approach significantly differed from 30 %. The conclusion supports the evidence some users prefer longer passwords.

Proportion tests rely on a continuous probability distribution, namely χ^2 , which may have introduced a non-negligible error. To verify, Fisher’s exact test was employed and despite it being usual for contingency tables with small sample sizes where χ^2 approximation without

continuity correction leads to elevated α , its use is not limited to this particular situation. The source table is depicted in Figure 10 under the assumption the two data sets are disjunctive, i.e., no user had an account in both databases. H_0 states no difference in proportions between the two variables exist, while H_1 asserts the contrary.

Tab. 10: Contingency table for Fisher's exact test. Based on exhaustive enumeration of all tables whose marginal frequencies are equal to those in the original one, it calculates p-value directly rather than by asymptotic approximation.

Source: Own work.

	Password length ≤ 6	Password length > 7	Sum
Sony Pictures	11000	26608	37608
Data set	65 836	482 850	548 686
Sum	76 836	509 458	586 294

```
table<-matrix(c(11000,26608,65836,482850), nrow=2)
fisher.test(table,or=1,alternative="two.sided",conf.int=TRUE,conf.level
=0.99)
```

Fisher's Exact Test for Count Data

```
data: table
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
99 percent confidence interval:
 2.938586 3.127969
sample estimates:
odds ratio
 3.032004
```

After entering the contingency table and specifying the number of rows, the test was run with explicit request for a non-directional alternative hypothesis. Of interest is the p-value which seems to confirm the claim passwords differ significantly across sites with regards to minimum length. The root cause can be tracked to inefficiency of brute-force attack because it was CPU-bound which required limiting the search space to 1–6 characters. Employing a GPU and expanding the candidate pool, particularly to the 8-character threshold, would have changed the results considerably. Source code for graphs can be found in Appendix C.

Employing a slightly modified method by Amico, Michiardi, and Roudier (2010), we will now launch a set of regular expressions to approximately determine composition of the uncovered passwords. Figure 69 depicted Hashcat-GUI before the brute-force attack had been started: note in particular the specification of the output format to be `hash[:salt]:plain`. The salt is optional as is the case here. Table 11 lists the patterns with percentages, Figure 74 depicts pattern counts graphically.

Several observations can be made regarding the composition rules. The histogram shows four distinct groups: 1+2, 3+4, 5+6+7, and 8+9+10 which describe the password selection rationale. Users in the first category (1+2) selected their credentials to contain at least some lowercase letters or more than one number, but only 40 % opted to include both. Less than 20 % started with a number; on the other hand, 12 % finalized the string with “1.” Also of note is strong

Tab. 11: *Brute-force output pattern matching. For the purposes of the case study, regular expressions use the PCRE (Perl Compatible Regular Expression) engine whose lower bound on execution time is proportional to the input size.*

Source: Own work.

Pattern	Expression	Password contains at least one	Relative (%)	Absolute
1	[a-z]+	Lowercase	80.61	53069
2	[0-9]+	Number	67.24	44271
3	[a-zA-Z]+[0-9]+	Lowercase or uppercase followed by a number	40.50	26665
4	[a-z]+[0-9]+	Lowercase and a number	38.81	25551
5	[0-9]+[a-zA-Z]+	Number followed by lowercase or uppercase	16.02	10548
6	[0-9]+[a-z]+	Number followed by lowercase	15.12	9952
7	[a-z]+1	Lowercase followed by number 1	12.03	7919
8	[A-Z]+	Uppercase	5.73	3774
9	[a-zA-Z]+	Lowercase and uppercase	1.19	784
10	[a-z]+[A-Z]+[0-9]+	Lowercase, uppercase, and a number	0.32	209

disinclination for uppercase passwords, presumably because of the added effort to hold a modifier key. More complex patterns combining uppercase, lowercase, and numerical characters were virtually non-existent in the brute-force output. Several explanations are plausible:

- Users mentally separate accounts into “primary” and “secondary”: for the former (financial, emails, medical records), care is taken to ensure the passwords have sufficient length and complexity according to some implicit or explicit guidelines; for secondary accounts, any string complying with the minimum password-selection requirements the service enforces is sufficient as long as it is easily memorable (groups 1+2).
- Users believe prepending/appending numbers and alternating lowercase and uppercase letters substantially challenges reverse engineering and disadvantages the adversary while keeping password memorability on an acceptable level (groups 3–7).
- Users consider complexity to be the only measure of password security and prefer shorter strings with uppercase letters to thwart lowercase-only alphabet enumeration, or combine character sets to increase the search space and the time factor (groups 8+9+10).

None of the arguments takes hardware capabilities into account: even low-end CPU is capable to enumerate all passwords regardless of composition in the 1–6 pool in reasonable time, and adding a high-end GPU extends the search space further. Most importantly, advances in circuit design will see attackers migrate to brute-force attacks if user habits do not reflect the development. A single GPU can currently iterate through the entire 1–8 range; in fact, oclHashcat-lite offers 8 as the default parameter for exhaustive enumeration. The bound is not fixed, though, because distributing the workload among several GPU instances (physical or virtual) speeds up the search. Future innovations will make extracting long passwords trivial; single-set passwords which consist solely of lowercase, uppercase, or numerical characters are susceptible even today. Best practices for password selection and management will be discussed in chapter 6.1.

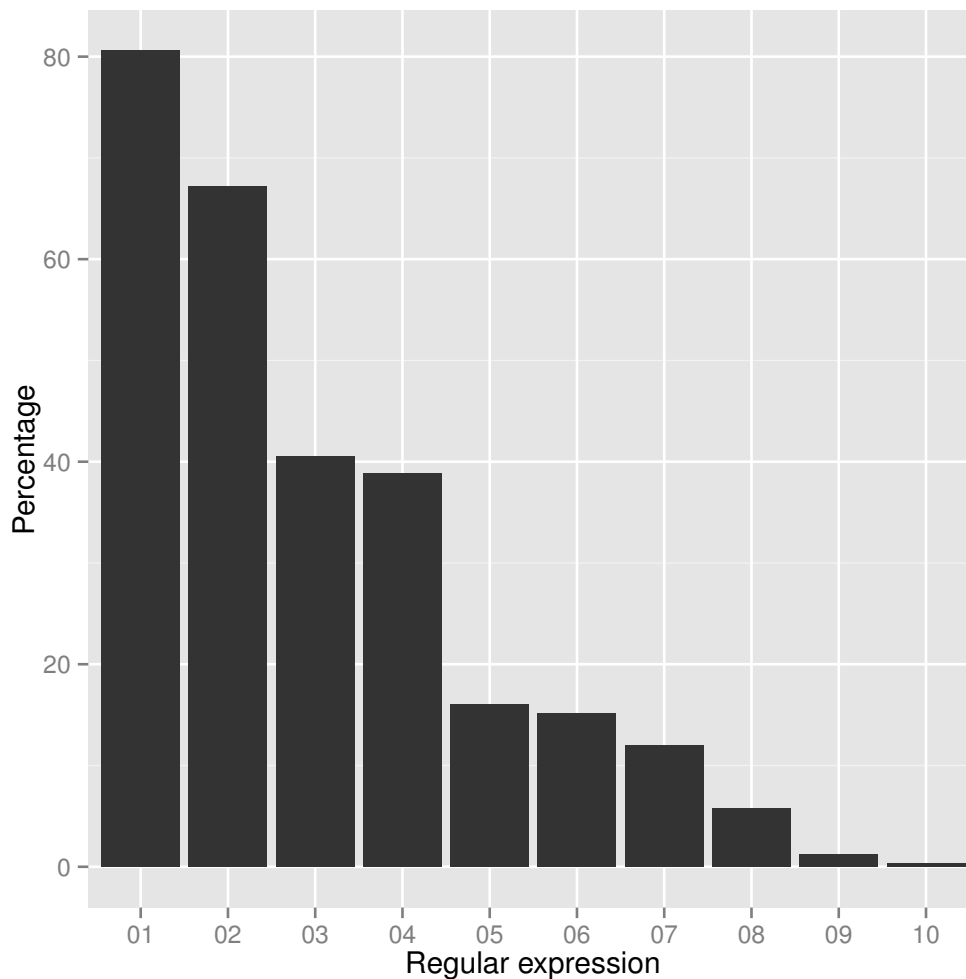


Fig. 74: Password composition patterns from brute-force attack. The groups are conjunctive, i.e., one string may belong to several categories and sums therefore cannot be used.
 Source: Own work.

Next, **straight mode dictionary attack results** will be analyzed. In this case, no limit on how long the passwords can be as imposed, a more realistic scenario because many services prohibit use of short strings and set an arbitrary lower bound, usually 8. An overview is provided in Table 12. The output text file size is 5066125 bytes, a sample of 20 MD5 hashes with corresponding passwords is depicted in Figure 75.

Inspecting the structure of the passwords, it is apparent both complexity and length increased. Indeed, because word lists contain strings of varying composition with credentials uncovered in previous database leaks, the attack’s potency should be higher if users choose their passwords in a non-uniform, biased manner and prefer memorability, not security. The results indicate dictionary attack is more efficient in comparison with naïve brute-force implementation as it fares better in all metrics.

For situations where time is a constraint, straight mode dictionary attack is clearly a preferred solution. Brute-force attack is usable when hardware and time do not limit the adversary, or source data was encountered where the passwords disfavor dictionary vector by explicitly precluding use of known weak sequences such as “password,” “123456,” etc. If the perpetrator gets hold of the blacklist, though, she can easily tweak her dictionaries to omit those strings and focus on custom-tailoring the rules under the assumption users will strive to meet the baseline security

Tab. 12: *Straight mode dictionary attack summary. Requiring no input apart from a suitable dictionary which contains previously-leaked passwords, the results demonstrate credentials are frequently reused across sites without modifications.*

Source: Own work.

Started on: Tue Oct 21, 2013, 12:00:00 CEST
Finished on: Tue Oct 21, 2013, 12:07:46 CEST
Total time: 00:00:07:46 (dd:hh:mm:ss)
Hashes total: 482 850
Hashes recovered: 119 996 (24.85 %)
Hashed remaining: 362 854
Word list segments: 11

```

1 18012668ae6d42fc9cbd3b1169f32edb:36900963
2 c8370ad37d0828c82a46f623b5ea225e:3691215
3 d82497f732f093d9a1b6a0bc3e53c4d1:369121518
4 70743626e51623d7462eedecb6c64f1a:36913691
5 b722942bd9ddaabb60db3234b11deeee80:369147
6 3a715498b51a468e87e07820b3e72734:369258
7 860c6d6d6c82a25147a6bb7312fbd3a7:369258147
8 7061b2706fb85e0c344c104a167c27d7:36933693
9 5f382d8ba25b8118f41f687563194758:#9506668a#
10 b4ab3004d2c12de1f5c8724fa6f000b1:3693525
11 117a8cc3e4c3009db3db322010e06b68:#az12X#
12 c2610f901735286e712ef0a11df1535b:369369
13 f67c11f5e2c07e24a4bc7b02add1226f:#badboy#
14 435bfa0c3415de27a3e9d9e7ec0c6f9d:369369369
15 2f1d094674b7ac56d40c9ff1cbcabbed:#EDCvfr4
16 315dc8da2c83689f3e716484b64098e6:36939500
17 72f47ff100c03e968703c66a186566ff:$01asuka
18 bce8e893e55c3adf9f60e6e2319f27f1:36952058
19 1c3da72d04c8bffc07068327290f5900:$kad4Life
20 059a0b9fe422a914cada21f976152195:36953695

```

Fig. 75: *Straight mode dictionary attack plaintext passwords sample. Repetitive patterns longer than 6 characters are commonplace in many word lists, as well as syntactically-correct dictionary entries.*

Source: Own work.

requirements by padding (appending, prepending) or otherwise mingling favored passwords to extend their length up to the minimum allowed count. Such policies help reverse engineers to tweak the attack and comply with the same rules users have to obey.

We can further conclude brute-force is still viable as many users select short passwords and underestimate computational capabilities of modern hardware. This renders the attack an option for low-security accounts. However, its efficiency is hampered by a gradual shift toward longer login credentials. Conversely, dictionary attack is susceptible to elevated complexity as the rules to generate such strings are non-trivial and may not be immediately obvious. Depicted in Figure 76 is a histogram of password lengths obtained from straight mode dictionary attack plaintexts.

An obvious tendency toward strings longer than 6 characters is corroborated by bin frequencies. Summing up the two most populated categories, it can be concluded 85.710% users chose passwords of lengths (6, 10]. Also of interest are 19 passwords shorter than six characters which should have been uncovered by brute-force attack. The ?a (alphanumeric characters and symbols) switch was selected in Hashcat-GUI as a trade-off between computational time and search space

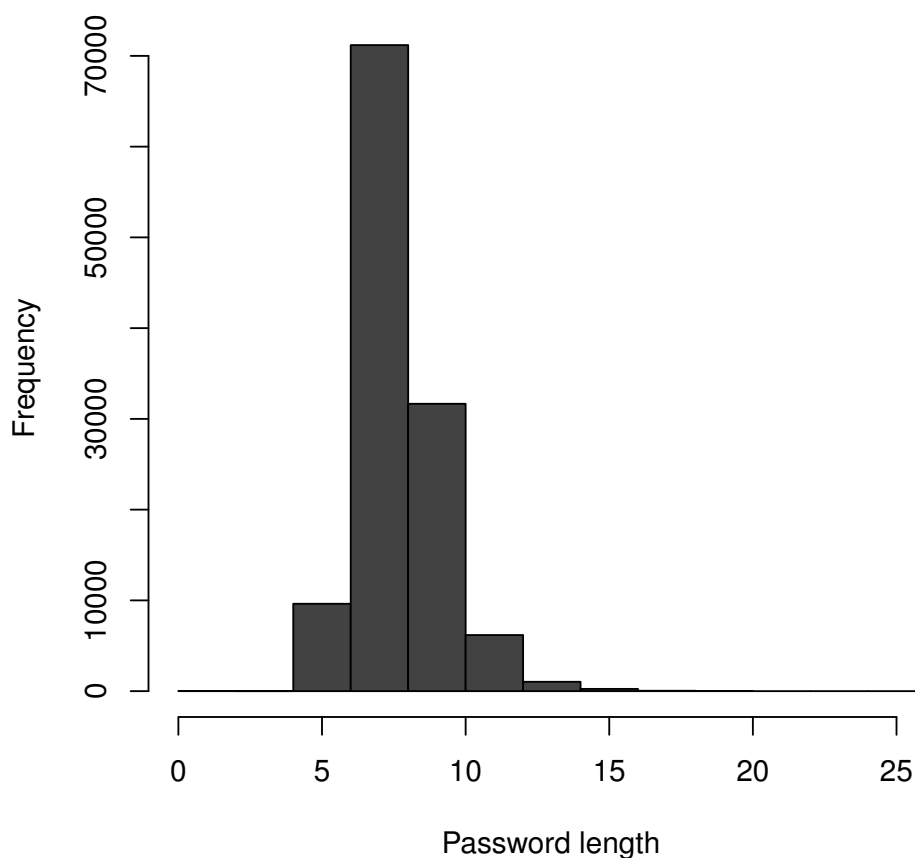


Fig. 76: Straight mode dictionary attack password length histogram. Each bin corresponds to a sum of all passwords of lengths up to and including the cutoff value.
 Source: Own work.

size. Others were left out due to presumed scarcity of the particular character set. We can now conclude only 0.0158% of passwords contained either glyphs from the 0x0C0–0xFF range, German, French, or Russian alphabet, making the amount negligible compared to resources necessary to cycle through all possible combinations.

```
$breaks
[1] 0 2 4 6 8 10 12 14 16 18 20 22 24 26

$countss
[1] 16 3 9626 71187 31662 6180 1029 245 41 18 0 0 1
```

Only a single user in the data set chose their password to be (24, 26] characters long, and it was still uncovered without employing additional mutators, strongly suggesting the same credentials were reused across accounts one of which leaked, reversed, and the string incorporated to the source word list. Complexity, length, and uniqueness are all contributing factors for determining whether the string can withstand dictionary attacks: if the sequence is both sufficiently complex and long but has been recycled several times, it cannot be considered a secure alternative to per-site password.

Straight mode with rules dictionary attack results extends straight mode by adding 100000 rules which modify the candidate strings to account for varieties in how users alter their credentials. No arbitrary length limit was imposed. An overview is provided in Table 13. The output file size is 6 165 943 bytes, a sample of 20 MD5 hashes with corresponding passwords is depicted in Figure 77.

Tab. 13: *Straight mode with rules dictionary attack summary. To adjust for transfer from Central European Summer Time (UTC+02:00) to Central European Time (UTC+01:00), one hour was added to the execution time.*

Source: Own work.

Started on	Tue Oct 21, 2013, 12:29:59 CEST
Finished on	Fri Nov 01, 2013, 12:36:31 CET
Total time	10:24:56:32 (dd:hh:mm:ss)
Hashes total	363 321
Hashes recovered	144 347 (39.7299%)
Hashes remaining	218 974

```

1 96e79218965eb72c92a549dd5a330112:111111
2 97f164b4c1ad3bbf876b4b322ea1a68b:363636
3 1a100d2c0dab19c4430e7d73762b3423:333333
4 980ac217c6b51e7dc41040bec1edfec8:dddddd
5 52c69e3a57331081823331c4e69d3f2e:999999
6 0ec36f7a0fccb746dcf95eaa7ccad2c7:369867
7 70e4648106a799ad31a69adb6ce05ade:33663366
8 13bbf54a6850c393fb8d1b2b3bba997b:*****
9 d5ee2eedfcf7adc285db4967bd86910d:6666666
10 f14029217ff5e7a50cdc7e70f686cf29:hhhhhh
11 f379eaf3c831b04de153469d1bec345e:666666
12 73b197105b5366d300bcab1aba35fb9b:303030
13 21218cca77804d2ba1922c33e0151105:888888
14 2f3a7be98cb9144008736227ffe2951b:360360
15 b1039b8cf7a2976222c7814944143dd6:0000007
16 670b14728ad9902aecba32e22fa4f6bd:000000
17 c1a41159a94ed9bf45e035f6a2a2ca79:300300
18 343b1c4a3ea721b2d640fc8700db0f36:qqqqqq
19 875f26fdb1cecf20ceb4ca028263dec6:bbbbbb
20 a73f86ae408af70b67141843e7130723:111111

```

Fig. 77: *Rule-based mode dictionary attack plaintext passwords sample. The repeating patterns are again present which can be covered by mutators.*

Source: Own work.

Inspection yields similar conclusions to those for straight mode dictionary attack in Figure 75: more complex and longer passwords than for brute-force attack. We can even infer some mutators which were used: repeating a single alphanumeric character or symbol, number 2-tuples and 3-tuples, keyboard combinations in close proximity (zzxx, zcvc, 3366). Once these patterns are uncovered in a large enough sample, the only measure protecting the information from being exploited is the cryptographic hash function at the database level over which the user does not have any control, and must rely on best practices the system administrators purportedly uphold. From the adversary's point of view, if attacks fail to produce the original sequence, the hash may lose its value because it resists the most common mingling rules and would require alternative word lists or advanced reverse engineering techniques.

In Table 13, note in particular the total time which was 8.541 higher than for brute-force attack. To quantify the difference, Table 16 on page 154 provides key metrics and demonstrates how inefficient rule-based mode becomes when the parameter (number of rules) is set excessively high and hardware bottlenecks, primarily CPU speed, are ignored to balance resources and results.

Rule-based mode is the least-effective of the three attacks, its advantage being a dictionary-based search space. Each password took more than 6.5 seconds to uncover on average, which was caused by the inflated search space size directly proportional to the word list line count. By loading a smaller corpus, the pool would shrink and depending on the quality, the hit rate could even increase if the dictionary more faithfully represented strings found in real-world credentials. Histogram of password lengths is depicted in Figure 78.

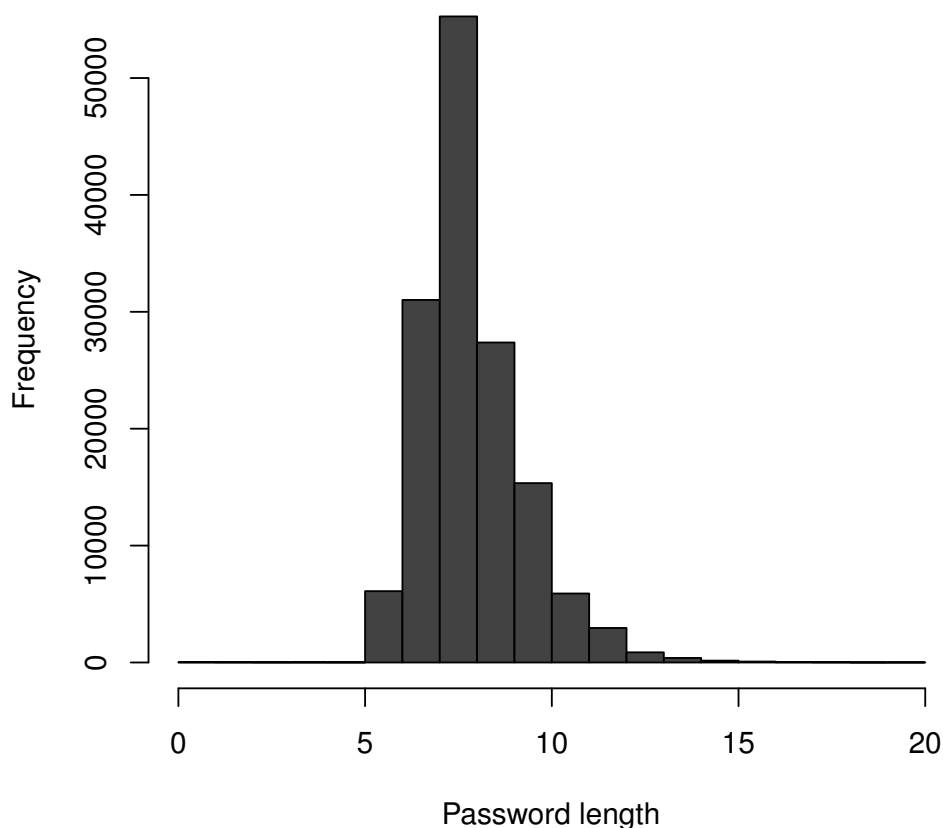


Fig. 78: Rule-based mode dictionary attack password length histogram. Most of the uncovered passwords are aggregated in the $(5, 10]$ interval, actionable intelligence for constructing custom rules. Source: Own work.

Bin frequencies show results do not differ from the straight mode. The proportion of passwords drawing from the character sets not covered by the brute-force enumeration is again low, specifically 0.0527%. This supports the conclusion an adversary modeled in chapters 2.4.1 and 5 would not see any additional benefit in including them in her search due to marginal increase in success rate alleviated by disproportional increase in running time. The benefit-cost ratios for alternative approach vectors promise higher return on resources utilized.

```

[1] 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
$counts
[1] 24    18    11    13    10   6103 31015 55268 27375 15345  5901  2953
[14] 868   387   156   78    25    17    4     6

```

Passwords uncovered with rule-based dictionary attack are more diverse length-wise. Their probability distribution is positively skewed and more heavy-tailed which points to more users selecting longer passwords in the [10,20] interval. However, their modes are both located between [6,8]: for the rule-based mode, the most populous bin is [6,8], for straight mode [7,8]. This suggests 8 characters to be a practical maximum which may correlate with the inability of users to remember complex long sequences. The finding will be addressed later in the thesis when devising password management best practices which should optimally shift the probability distribution’s modes to the right, skewing it negatively toward higher absolute lengths. Preempting inertial mindset of many users with an easy-to-use solution is necessary for broader adoption.

Toggle-case mode dictionary attack results are presented next and capture string case variations. For longer sequences, the toggling extends the search space considerably and Hashcat-GUI was thus instructed to throttle the maximum length to 16 characters. All passwords over the threshold were skipped in the corpus and omitted from computations. An overview is provided in Table 14. The output file size is 143078 bytes, a sample of 20 MD5 hashes is provided in Figure 79.

Tab. 14: *Toggle-case mode dictionary attack summary. All modes which load the corpus compartmentalize it into segments of fixed size which are processed sequentially.*
Source: Own work.

Started on	Fri Nov 01, 2013, 13:40:08 CET
Finished on	Sat Nov 02, 2013, 00:18:40 CET
Total time	00:10:30:32
Hashes total	218974
Hashes recovered	12957 (5.9171%)
Hashes remaining	206017

Compared with straight and rule-based modes, toggle-case managed to recover only more than 5 percent of hashes, indicating users prefer to choose words conforming to predictable structures compounded with basic rules but no case mingling. We postulate the reason might be twofold: either the data set aggregated credentials whose length or complexity were out of bounds and not covered by the test, or individuals do not use key modifiers for their passwords as much. This may be evidence that typing speed, i.e., entering the string quickly, is a desirable property expected in passwords.

Any sequence not uncovered previously which contains at least one uppercase character and its stem is included in the word list, is present in the output file. Because Hashcat-GUI removes reversed hashes and the target pool is progressively shrinking, multiple passes of any mode with identical parameters and corpus would not result in new discoveries. Indeed, a surrogate word list is the sole way to get different results *ceteris paribus*. Inspecting the sample, line 5 is an example of “leet” speak, a lexical substitution scheme where characters are replaced by similar-looking equivalents which should increase complexity and break straight and toggle-case attacks.

```

1 2928e2d1311a9175a4db41666884ff86:!2Qwaszx
2 e294e93611334c35452c9425962ef630:!Bollox1
3 97378f150df2de47495f060d28b27c3c:!Takeoff1
4 7c283f9e49073bdd52794453a123dc5:#1Kentucky
5 8360f73fbdbfc8f926ad8fe46133a6fc:#PasswOrd
6 b51e2529ad4ec311b3f94aae2512b689:$kyW4lk3r
7 953784941c333184f7d0bd3263e25e28:%TGB6yhn
8 55d682a39d4063ad979614166a439160:%Th0mas
9 a558bb26a06c48035598a47c95127130:(Shorty
10 6888c8a6917ace337a2538545fb5e07d:(V)atrix
11 3fd3dc3ebd1f4a16a30852c5b8b7f118:**Lisa**
12 ecb88954c876f66d0b2c5405169c409d:*IFI2412
13 95949f411954b863e386dc06502850ee:*Imagine1
14 c330de430aab82c5beb091c760eec5f5:*Insignia
15 c8efb2266e1186cd61253d6080888629:.Daniel26
16 a36384031931d26dbc25e2da0e4d2a65:00000000Mm
17 07a6a779b44b863d1f0f9c8df8bfff0e7:000056ST
18 960b041ddc9c48b264e4f6f4c78f0c64:0013KILLER
19 08b67237722ef559b9bf474385e84794:001leiR
20 2dc949717bc973e450c6371997fb6ff3:007007Felix

```

Fig. 79: Toggle-case mode dictionary attack plaintext passwords sample. The list was case-insensitively sorted with special characters coming before letters.
Source: Own work.

Because of its popularity and limited number of rules, e.g., E-3, A-4, S-5, reverse engineering software and dictionaries incorporate them by default, negating any perceived advantage it offers. This is apparent from strings on lines 6 and 8. Histogram in Figure 80 depicts absolute length distributions.

Bin frequencies confirm no apparent deviation from the two previous modes. The three almost identical probability distributions lead us to believe a mathematical formula may exist which denotes the most likely selected password length which could used to draw conclusions about user preferences, e.g., are entertainment-related accounts underrepresented in relation to email or banking length-wise? Is 6–8 character password the most popular option due to limits of human brain, perpetuating public awareness campaigns, imposed policies, or strictly related to comfort? While the thesis will not focus on answering these questions, they constitute a viable future research avenue drawing from ergonomics, ICT, and psychology. Better understanding of human-machine interaction, devising more secure and at the same time comfortable authentication schemes together with ways to shift general population toward protecting their online identities rank among the benefits.

```

$breaks
[1] 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44

$countss
[1] 3  157 8283 4997 1138  186  40  4  1  0  0  0  0  0
[16] 0  0  0  0  0  0  0  0

```

Table 16 lists metrics for the toggle-case mode. It ranks higher than rule-based in hashes per seconds but below both straight mode and brute-force attack. It is second in the seconds per hash metric due to short running time. Search space size and hit rate are not included because manipulating, parsing, and executing regular expressions on very large files is computationally challenging. The process would first require executing an expression which would count the number of alphabetic characters, omitting numbers and special symbols as they are immutable,

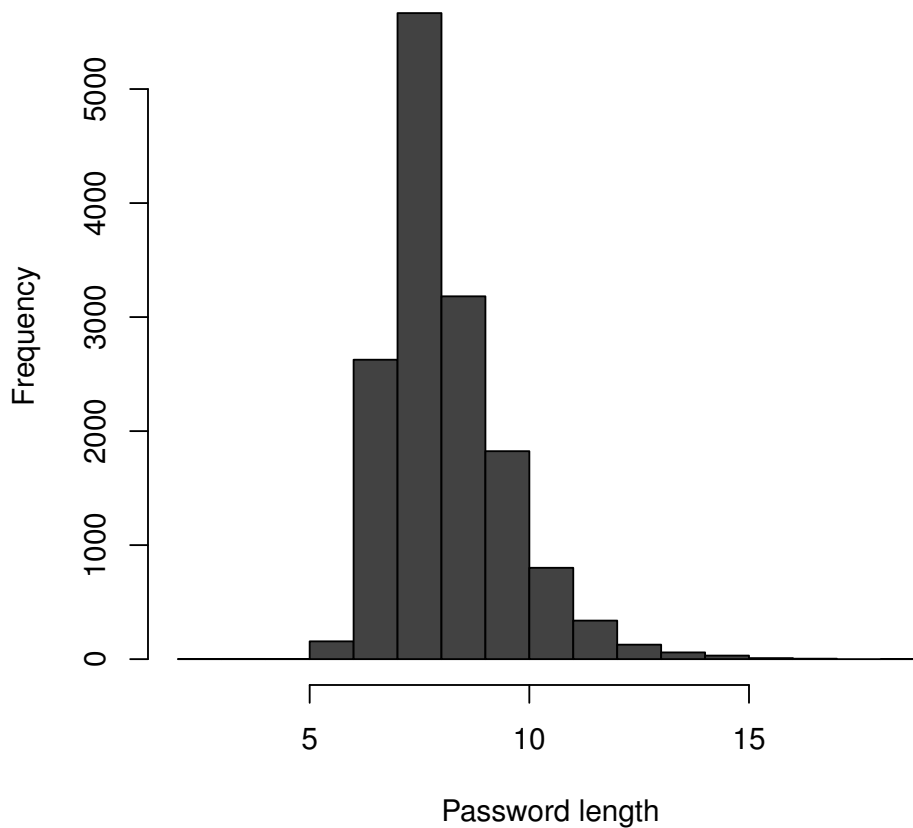


Fig. 80: Toggle-case dictionary attack password length histogram. Mode length was found to be 7 characters which hampers brute-force enumeration on CPU but is within capabilities of a GPU. Source: Own work.

and absolute word length. Both k and n are needed in equation $\frac{n!}{k! \times (n-k)!}$, mentioned in chapter 5.1.3. However, the formula is only applicable to a single string. The equation taking into account the dictionary size is

$$\sum_{j=1}^{n-x} \sum_{i=1}^y \frac{n!}{k_j! \times (n_i - k_j)!}, \tag{5.1.2}$$

where n is the current string's length, k the number of mutable characters inside it, x denotes the number of remaining characters (numbers and special symbols including spaces), and y all strings from the word list to be tested. As many as 63 941 069 (dictionary size) entries can be expected to enter the equation via y which makes the calculation cumbersome to execute, especially the sum of factorials. We expect the search space size to be above brute-force and straight modes but lower than rule-based dictionary attack as per the difference in execution times between the two. If we abstract from variations in CPU performance and assume a uniform portion of the search space was sifted through each second, the toggle-case candidate pool would be approximately 2.3562×10^{11} , with rule-based mode a baseline. Having no way to validate the result, it is included only for reference.

Permutation mode dictionary attack results are the last ones to be analyzed. As case toggling was not a popular option, it had been predicted the would be lower than straight mode while requiring comparable execution time. This was not substantiated as evident in Table 15. The minimum and maximum string lengths above and beyond which no action was taken with regards to permuting the entry were set to 5 and 9, respectively.

Tab. 15: *Permutation mode dictionary attack summary. The number of remaining hashes represent passwords which have withstood the 5 Hashcat modes picked in the case study. Source: Own work.*

Started on	Sat Nov 02, 2013, 13:34:37 CET
Finished on	Wed Nov 06, 2013, 00:11:27 CET
Total time	02:10:36:50 (dd:hh:mm:ss)
Hashes total	206017
Hashes recovered	15261 (7.4076%)
Hashes remaining	190756

The results superseded toggle-case mode which suggests intra-word permutations are preferable. The argument presented earlier, hinting at users unwilling to press a modifier key when entering passwords may confirm the discrepancy between what had been expected and reality: switched characters do not produce discomfort from additional key presses, and provide some protection against straight and rule-based modes. If the string is sufficiently long, even the permutation vector is not viable due to CPU throughput limitations. Moreover, since neither oclHashcat-plus nor oclHashcat-lite directly support it, the adversary is bound to Hashcat or alternative tools, e.g., John the Ripper. Knowledge of reverse engineering strategies may thus help to designate password policies which thwart basic modes and force the perpetrator to utilize advanced, more time-consuming procedures. In addition to strong hash functions, the rules can help mitigate damage substantially. Histogram of password lengths is depicted in Figure 81.

Absolute bin frequencies seem to confirm tendencies observed when analyzing results in the previous modes. Breakpoints were hard coded because the two respective parameters, minimum and maximum lengths (inclusive) of word list entries to be included, were set manually. Other categories are empty and do not provide any relevant information.

`$breaks`

```
[1] 5 6 7 8 9
```

`$counts`

```
[1] 2203 2965 6284 3808
```

Directing efforts at the most populous categories results in the highest impact, an approach consistent with the behavior of a model adversary described in chapter 2.4.1. Passwords of length $(10, x]$ where x is any positive integer denoting the maximum number of characters should be considered outliers producing inferior returns on resources and time. Note that x is not set uniformly and no arbitrary upper bound exists. It is a subjective measure: some users may set $x = 10$, i.e., none of their authorization tokens exceed length 10, whereas some may set it to $x = 40$. Complexity, length, and uniqueness are factors which determine password security, and maximizing one without regard to others leads to vulnerabilities. Some online services disallow

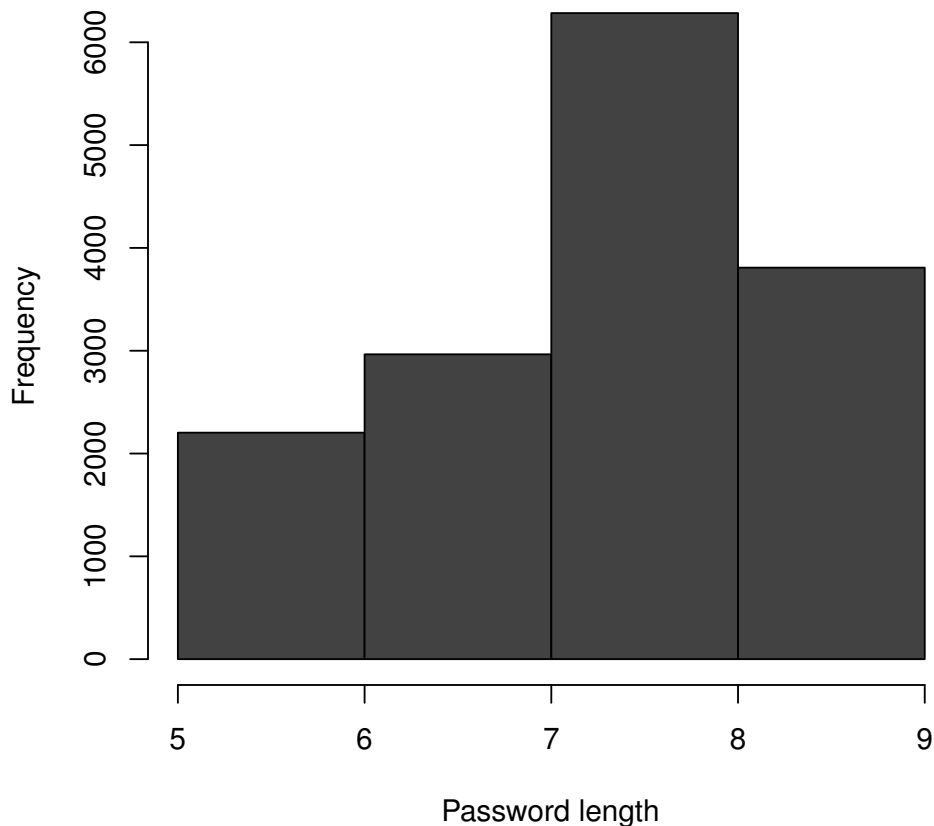


Fig. 81: *Permutation dictionary attack password length histogram. Limited to the [5,9] interval, the distribution is close to those uncovered in previous modes, with 7-character passwords the most popular choice.*

Source: Own work.

strings above certain lengths, bounding x to a fixed value which leaks information on how to generate custom rules.

In chapter 5.1.3, the permutation mode search space size was established to be $n!$, where n is the string length. Expanding to the entire corpus, the formula is modified to $\sum_{i=1}^y n_i!$, where y denotes the total number of entries between 5–9 characters. Getting exact figure using regular expression would take prohibitive amount of time due to word list size. However, as it contains strings from existing database leaks, we assume the proportion of such passwords is higher than any other group, expanding the search space. This is advantageous for the adversary because the probability the stem was incorporated is higher if she suspects the plaintext sequences were of shorter length.

Table 16 quantitatively overviews performance metrics for brute-force and all dictionary attack modes. Search space sizes and success rates were omitted for toggle-case and permutation as explained above. Hashes per second denote the number of strings discovered per second, represented by arithmetic mean, i.e., averaging the figures by supposing a fixed amount of passwords uncovered during each interval. The higher the value, the lower the time to exhaust the search space and thus, the more effective the attack. Seconds per hash quantifies average

Tab. 16: *Brute-force and dictionary attack comparison metrics. Rule-based attack was strongly disadvantaged by a long execution time caused by search space extension, toggle-case was found to be the worst performer even when the cutoff parameter was set to 16. Permutation gave better results even when restricted to 5–9.*

Source: Own work.

Metric	Preferred direction	Brute-force	Straight mode	Rule-based	Toggle-case	Permutation
Hashes per second	High	0.5895	257.502	0.1513	0.3425	0.0723
Seconds per hash	Low	1.6963	0.0039	6.6076	2.9198	13.8267
Search space size	High/Low	95^6	63941069	6.3941×10^{12}	–	–
Hit rate (percent)	High	8.9562×10^{-6}	0.1877	2.2575×10^{-8}	–	–

time it took to uncover a single hash. The lower the value, the faster two successive hashes were found, indicating the candidate pool was selected reasonably to encompass the most likely-occurring strings. Both metrics are directly related to search space size with hardware performance fixed during testing and therefore not factored into the results. Search space size indicates all combinations the attack must go through to terminate successfully. The preferred direction is both high and low as the intruder would like to cover the highest proportion of strings, but at the same time prune them only to those most likely procuring plaintext passwords. Enumerating the space exhaustively with no regard to probability of each sequence being picked diminishes the method’s effectiveness and forces to spend scarce resources on unproductive avenues of approach. The only metric where brute-force attack dominates over straight mode dictionary attack is the search space size which, however, does not take into account probabilities mentioned earlier. This negatively affects performance as indicated by the hashes per second and seconds per hash. Graphical representation will not be plotted because the values differ by orders of magnitude. Hit rate is calculated as the number of found hashes divided by search space size, and denotes a proportion of fingerprints contained in the pool. A relative measure with a maximum equal to 1, higher values indicate greater overlap of the candidate and target sets, 1 would signify the search space comprises only the strings found in the data set, a purely theoretical scenario.

The last two metrics, search space size and hit rate, depend on x in the $1 \dots x$ range, and dictionary size for brute-force and straight mode dictionary attacks. For example: if the attacker wishes to enumerate the 1–8 pool ($x = 8$), the total combination count equals 95^8 with all entries sampled from discrete uniform probability distribution. Conversely, dictionary attack can be said to utilize binomial distribution as each string has certain non-zero probability of being present in the word list with a yes/no decision determined by whether it was encountered previously, i.e.,

a Bernoulli trial of the following form:

$$P(s) = \begin{cases} 1 & \text{if string } s \text{ was encountered} \\ 0 & \text{otherwise} \end{cases} \quad (5.1.3)$$

Dictionary attack is more flexible than brute-force except for cases where the parameters are set excessively high, e.g., large word list combined with high number of mutations applied to each entry. Chapter 5.1.3 mentioned that Hashcat-GUI supports as many as 1 000 000 randomly-generated rules in a single pass. A word list of several hundred million lines would force the program to iterate for prohibitively long. Brute-force attack does not require any tweaking apart from selecting x as the upper bound. Even small changes result in vastly expanded search space: an increase from $x = 6$ to $x = 7$ leads to search space expansion of about 6.9099×10^{13} . The delta between $x = 6$, a currently recommended CPU-based maximum, and $x = 8$, usual in GPU-powered cracking, is 6.6635×10^{15} . The adversary should therefore strive to balance performance, hardware cost, and time as well as consider alternative vectors of approach before committing to any particular attack.

Plotting the first two metrics in Figure 82 depicts differences between the modes and shows their time comparison, a primary criterion when the adversary faces a finite window of opportunity she strives to exploit as efficiently as possible. For a final overview, Table 17 depicts results obtained using various attack scenarios in case study 1, Figure 83 visualizes the absolute and relative frequencies graphically.

Tab. 17: *Case study 1 final results. Decimal truncation prevented listing relative and relative cumulative figures fully, although they were calculated with to reach 100%. Source: Own work.*

Attack	Hashes reversed			
	Absolute	Absolute cumulative	Relative	Relative cumulative
Brute-force	65 836	65 836	11.9988%	11.9988%
Straight mode dictionary	119 996	185 832	21.8697%	33.8686%
Straight mode rule-based dictionary	144 347	330 179	26.3078%	60.1763%
Toggle-case dictionary	12 957	343 136	2.3611%	62.5279%
Permutation dictionary	15 261	358 397	2.7814%	65.3191%
Remaining	190 289	548 686	34.6809%	100%
Total	548 686	548 686	100%	100%

The results should be a warning sign for both website administrators and users to re-evaluate their habits: the former should strive to implement technical safeguards to make the reverse

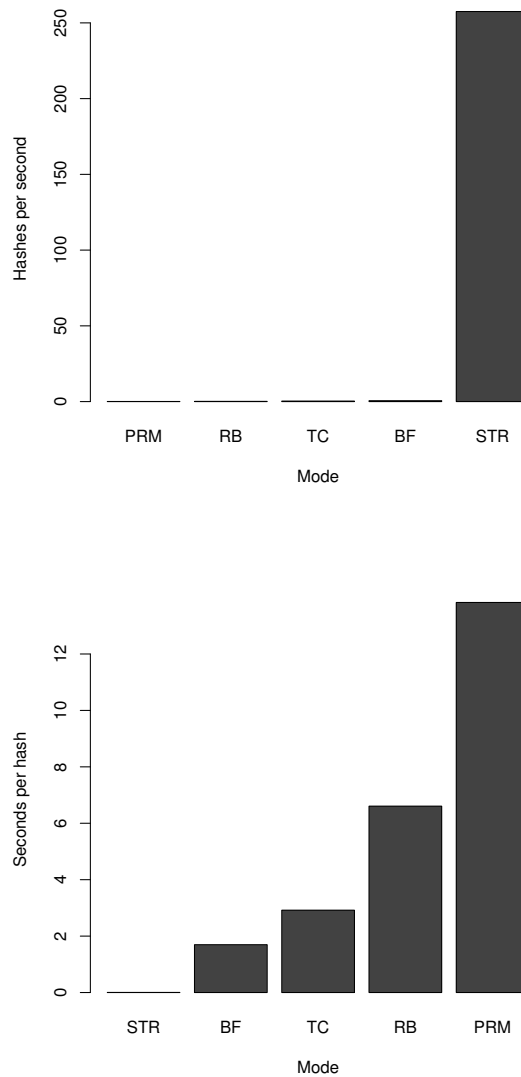


Fig. 82: Case study 1 selected metrics graph. Straight mode dictionary attack exhibits orders of magnitude higher hashes per second and seconds per hash ratios compared to others. TC: toggle-case, PRM: permutation, BF: brute-force, STR: straight, RB: rule-based mode. Source: Own work.

engineering process challenging which partially compensates lacking password habits with high computational demands. Users should strictly obey requirements imposed on them in face of increasing hardware performance and algorithmic optimization. Periodic rotation of credentials with high-security alternatives is an electronic identity protection baseline.

The case study did not aim to reverse engineer all hashes in the data set: the remaining fingerprints likely contain strong passwords, measured by length and complexity. Efficiency of dictionary attack is strongly correlated with the quality of the word list: if a password consists of a string not present in the dictionary and not conforming even to complex rules, it will withstand attempts at cracking from an unskilled adversary. One way is to employ alternate means, e.g., Markov chains or PCFG either separately or in combination, but intruders modeled in chapters 2.4.1 and 5 usually do not possess knowledge to apply advanced techniques as they necessitate additional inputs apart from running automated tools. Programs which offer means to apply such advanced methods are freely available, though. By building, maintaining, and

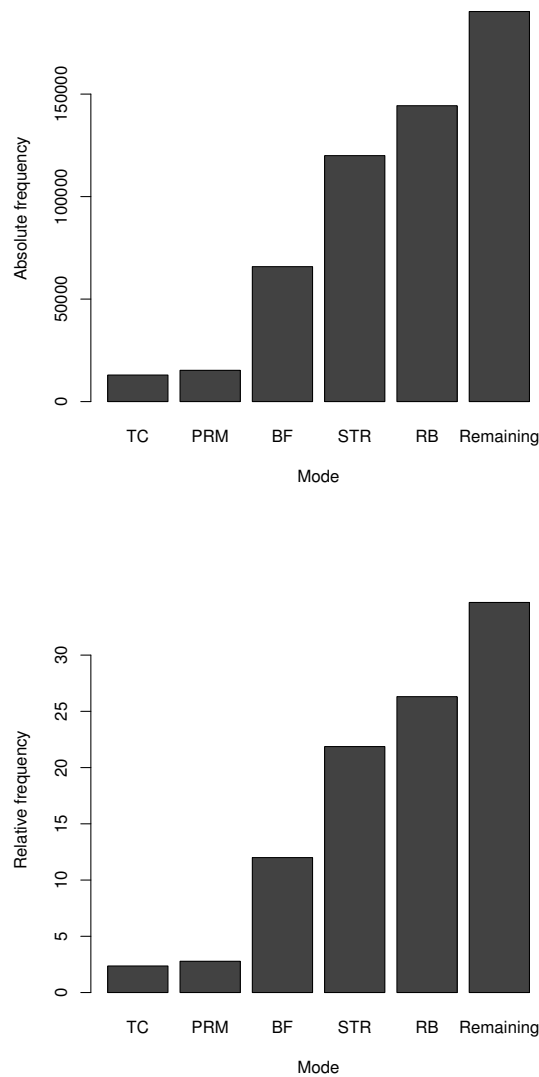


Fig. 83: Case study 1 final results graph. Absolute and relative frequencies are plotted separately for clarity, categories are sorted in ascending order. TC: toggle-case, PRM: permutation, BF: brute-force, STR: straight, RB: rule-based mode. Source: Own work.

updating software and know-how, programmatic and logical procedures can be deployed quickly and with high success rates.

Using a single corpus and sticking to basic modes, we were able to uncover close to 70 % of all passwords in the data set; in practice, custom-tailored rules and multiple word lists are employed to increase success rate over 90 % and more. Due to widespread password reuse, the consequences span compromised email and banking services, assuming victims' electronic identities with unrestricted access to financial operations and personal histories, and internal network breaches which may compromise sensitive electronic assets. The context of a single database leak must therefore not be underestimated.

The question posed in chapter 5.1.1 was: **How effectively can attacker possessing low to moderate ICT security knowledge reverse engineer password digests using trivial methods with pre-defined rules?** We can with high degree of certainty conclude the process is not only

trivial for anyone to execute, but the results favor adversaries at the expense of users. Clearly-stated password composition rules, despite giving away actionable intelligence upon which custom rules for password cracking can be constructed, are needed. High-level, understandable explanations are also necessary so that individuals can internalize and apply them whenever they are presented with a choice between comfort, ease of use, and security in passwords. Best practices and tools should be designed to make login credentials management as bearable and streamlined as possible, a prerequisite the ICT governance model in chapter 6 fully adheres to.

5.2 Case Study 2: Penetration Testing

Penetration testing (PT) was discussed in chapter 2.4.8 and is closely tied to vulnerability assessment (VA) to the point where the two are separated by a single additional step, proof of concept. VA is understood as "... necessary for discovering potential vulnerabilities throughout the environment. There are many tools available that automate this process so that even an inexperienced security professional or administrator can effectively determine the security posture of their environment... Full exploitation of systems and services is not generally in scope of a normal vulnerability assessment engagement. Systems are typically enumerated and evaluated for vulnerabilities, and testing can be done with or without authentication. Most vulnerability management and scanning solutions provide actionable reports that detail mitigation strategies such as applying missing patches, or correcting insecure system configurations" (Allen, 2012, p. 8). Before juxtaposing the delimitation with penetration testing, two comments will be presented.

Inexperienced security professionals mentioned in the preceding paragraph are considered equal to the script kiddies defined in chapter 5. The nature of VA and PT tools results in a situation where both groups will employ the same programmatic means and techniques to test ICT security. As evidenced by news of high-profile database breaches, even corporations whose online presence and revenue streams depend heavily on customer trust and reputation fall victims to simple attack vectors, e.g., SQL injection. Hardening infrastructures should be a priority which VA and PT help to facilitate by launching controlled scenarios which analyze how systems behave in standard and edge cases. Computers are not their only application area, however: risk analysis (chapter 2.1.4) and mitigation strategies are crucial for any host providing services to users who are critically or non-critically dependent on its stability: hospitals, energy and water supply facilities, transportation grids, communication media, government and local authorities, military forces. Business continuity management discussed later in the thesis is a framework for organizations to minimize negative effects of the threat factors they face. Mission-critical policies must clearly delineate steps to take if and when such situations occur.

Another point of interest in the VA description above is that testing can be done with or without authentication. This is a controversial point because any interaction outside of what is considered "acceptable," e.g., sending repeated probing requests, can be classified as harmful. System logs, often in conjunction with dedicated IDS, monitor any network activity of suspicious origin and trigger alerts if the violation is determined serious and threatening system stability or security. Even when no warnings are raised, though, intensive VA pollutes logs with entries which consume storage capacities and require additional CPU cycles to process. A gray zone, the author is of the opinion a permission from the system operator should always be sought where the test can be interpreted as a precursor to breach, or raise reasonable doubts about whether the probing is a first step of a future intrusive action. Chapter 3 mentioned a strict code of conduct will be followed. While a threshold on how many requests the source machine is allowed to send per second to preclude logs from registering too many events can be set, rules

which are not to be violated during the VA or PT were. It is easy to underestimate consequences of thorough testing, and establishing a communication channel between the analyst and the IT personnel beforehand can limit the risks.

Penetration testing is understood as allowing "... business to understand if the mitigation strategies employed are actually working as expected; it essentially takes the guesswork out of the equation. The penetration tester will be expected to emulate the actions that an attacker would attempt and will be challenged with proving that they were able to compromise the critical systems targeted. The most successful penetration tests result in the penetration tester being able to prove without a doubt that the vulnerabilities that are found will lead to a significant loss of revenue unless properly addressed... Penetration testing requires a higher skill level than is needed for vulnerability analysis" (Allen, 2012, p. 8). Again, a few comments are warranted.

Because PT goes a step beyond VA in that the vulnerabilities identified are exploited to gain system access and prove viability of the test scenario, legal obligations are much more strict because with a fully-privileged account, the analyst can trivially compromise all the CIA triad elements (chapter 2.2) and cause resources to become unavailable for legitimate internal (employees) and external (customers) users, incurring productivity and financial losses in the process. Causing a breach is made even easier when focusing on the human element via social engineering vectors as hardware and software are improved to eliminate obvious avenues for unauthorized entry. By ordering a PT, the client effectively permits controlled infrastructure compromise under contractual terms which explicitly specify test scope and depth. Nevertheless, some assumptions are hinted at only implicitly, namely related to code of ethics for which no unified set of criteria exists.

A proposal has been made (Schreiber, 2010) which expands more on the characteristics an analyst should comply with: independence; prohibition of commission fees; care; professionalism and quality assurance; liability; impartiality, neutrality and transparency; conflicts of interests; strict obedience of laws; respect of human beings (social engineering was specifically mentioned); and quoting correctly. The code "... should lay the basis for testers to get out of their original position on the verge of crime and the circle of socially responsible professions" (Schreiber, 2010, p. 3). The contract should state rights and responsibilities of both parties, test scope, depth, and time frame as well as prohibited, off-limits conduct, e.g., threatening CIA of critical resources. Two professional bodies, ACM and the IEEE, defined their own ethical codices, mentioned in chapter 5.1.1. The former states that "[t]heft or destruction of tangible and electronic property is prohibited. . . Trespassing includes accessing communication networks and computer systems, or accounts and/or files associated with those systems, without explicit authorization to do so. Individuals and organizations have the right to restrict access to their systems so long as they do not violate the discrimination principle. . . No one should enter or use another's computer system, software, or data files without permission. One must always appropriate approval before using system resources, including communication ports, file space, other system peripherals, and computer time" (ACM, 2013).

Assembling a report is the final phase of VA and PT but is sometimes treated as secondary to a proof of concept, i.e., demonstration the vulnerability leads to exploit and system breach. The client should be continuously informed about each and every action taken by the analyst, optimally with a dedicated IT personnel available for fast response. The final document should be prepared with attention to detail, and maintain objective tone without subjective assessments. It should collect actionable intelligence and constitute basis for corrective and preventive measures, detailed in chapter 2.2.3. The output can be used in the following ways:

- "as a reference point for corrective actions,"

- “in defining mitigation activities to address identified vulnerabilities,”
- “as a benchmark for tracking an organization’s progress in meeting security requirements,”
- “to assess the implementation status of system security requirements,”
- “to conduct cost/benefit analysis for improvements to system security,”
- “to enhance other life cycle activities, such as risk assessments, [certification and accreditation], and process improvement efforts,”
- “to meet reporting requirements. . .” (Scarfone et al., 2008, p. 62).

Examples of evidence are logs, screen shots, textual information, videos, and links to vulnerability database entries. The report should focus on low-level technical details to clearly demonstrate which flaw enabled to exploit the vulnerability, but also summarize the findings generally for managers with limited ICT knowledge. Documents pertaining to VA and PT must be kept strictly confidential: they thoroughly describe steps taken to provably compromise the target and are trivially reproducible. The analyst may withhold the report from anyone but the entity who ordered the test, and take steps to secure storage media from unauthorized access. Such precautions are needed particularly when critical shortcomings are uncovered in the ICT infrastructure. A non-disclosure agreement is strongly recommended.

To summarize, both VA and PT comprise a series of phases whose purpose is to thoroughly test ICT infrastructure by systematic enumeration, description, and presentation of exploitable vulnerabilities and open attack vectors. The difference between the two is that VA does not act on the findings but rather aims to provide concise and actionable evidence, whereas PT utilizes adversarial mindset and tools to compromise the system, reporting back the results as proven in real-world settings. Vulnerability assessment can be heavily automated while penetration testing presupposes technical background ranging from basic to extensive, depending on the level of sophistication needed to successfully deploy and execute malicious payloads. A detailed step-by-step report is written afterwards which also suggests corrective measures. The tests constitute security snapshots valid up until the time they were carried out. Neither offers predictive capabilities: performing VA or PT at distinct intervals could serve as an assessment tool for organizational security over time.

5.2.1 Phase 1: Background, Methodology

Before the first phase can be started, methodological background needs to be established. The case study will demonstrate system enumeration using public information sources. It will attempt to provide answer to the following question: **What relevant and usable data about the target’s ICT infrastructure can an attacker who possesses low to moderate ICT security gain using freely-available software?** We will do so by establishing a chain of logically-grouped actions with output of each serving as input for the following ones. The phases each test should have are (EC-Council, 2010, pp. 1-4):

- defining the scope: “Before performing a penetration test, it is necessary to define the range of the testing. For different types of penetration testing, different types of network devices exist. The testing criteria can target the entire network and systems, or it can simply target devices such as Web servers, routers, firewalls, DNS servers, mail [servers], and FTP servers,”
- performing the penetration test: “Each company ensures that the processes they are implementing for a penetration test are appropriate. This involves gathering all the information significant to security vulnerabilities. It is the responsibility of the tester to make sure the applications, networks, and systems are not vulnerable to a security risk that could allow unauthorized access,”

- reporting and delivering results: “Once the penetration testing is completed, security testers examine all information derived from the testing procedure... Testers make recommendations for repairing found vulnerabilities and provide technical information on how to fix vulnerabilities found in the system.”

The importance of the third phase in particular could be understated but preparing a concise report usable for both technical personnel and management may persuade both groups to dedicate limited resources toward quantitatively increasing security metrics associated with hardware, software, processes, and human element. Both VA and PT can be executed from the inside and outside of the target environment; upsides to each exist and they should be thought of as complementary to fully cover the attack surface. Some assumptions will be made about the information provided to the analyst and the extent of knowledge she possesses before experiments are commenced. We will restrict the scope of VA and PT as exhaustive enumeration would inflate the results with low-level technical details whose impact would not be obvious for the business-oriented audience. This will not decrease value or importance of the findings but rather demonstrate how freely-available tools enable the perpetrator to fingerprint target infrastructure.

The three types of VA and PT are black-box, gray-box, and white-box (chapter 2.4.8) testing, each modeling the intruder as respectively having no, partial, or extensive information about the business process structure (discussed in chapter 2.1.3) exploitable for malicious purposes. Black-box scenarios faithfully simulate real-world settings where the perpetrator engages the target as an unknown entity and builds her knowledge solely from data obtained from artifacts, documents, electronic assets, organizational culture, physical inspection, reconnaissance, and social engineering techniques. White-box testing takes the opposite approach and provides the analyst with extensive documentation, network topology, hardware and software specifications, organizational structure charts, process maps, and even suggested attack vectors when the client suspects they could be preferentially picked as the approach avenues of choice. Gray-box methodology assumes prior familiarity not extensive enough to constitute full information. White-box and gray-box can be employed to simulate insider threat: white-box for those with complete access, e.g., IT administrators, webmasters, and system operators, gray-box for those with limited access who can at best infer capabilities and features of the ICT infrastructure. While the difference is sometimes omitted and all insiders are hypothesized identical, the case study will honor the distinction. Gray-box model will be the methodology of choice. To prevent high-privileged individuals from exploiting their positions, the principle of least privilege (chapter 2.2.1) should be strictly enforced together with multiple accounts which can revert changes made by all others together with a policy of immediately revoking access in case of employment termination to prevent malicious tampering with critical assets. The second precaution should be implemented for any account which has been made obsolete; the attacker could exploit this vector by targeting dormant low-level accounts and escalate their privileges.

Case study stakeholders are IT managers, users, and system administrators. The decision to engage in testing must be made after presenting the benefits, pitfalls, and expected results. It is vital to repeat no product or service can objectively claim 100 % security. Both VA and PT evaluate readiness and resilience as a snapshot, but neither can anticipate developments which may render the target vulnerable in the future. Tests conducted on multiple occasions benefit the client by incorporating the factor of time which allows to evaluate security trends and determine if ICT has exhibited increase, decrease, or stagnation in the number of newly-discovered exploits threatening one or more components of the CIA triad. The analyst can also advise the customer on BYOD, employee training, patch management as well as aspects causing direct or indirect information leaks which help adversaries to amass comprehensive dossier on the victim.

Before particulars of the case study are devised, expected results should be hinted at. As the majority of the methods utilize networking to fingerprint, probe, and query remote hosts, some

data are expected to leak regardless of how well the target is secured. This is due to standards which require generating a response either in a fixed format, or from a set of options depending on the request type received. For example: a firewall is “. . . inserted between the premises network and the Internet to establish a controlled link and to erect an outer security wall or perimeter. The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and auditing can be imposed. The firewall may be a single computer system or a set of two or more systems that cooperate to perform the firewall function” (Aleshunas, 2009, p. 3). If configured not to send replies to probing requests, the lack of response strongly indicates a filtering module is present, and it is thus identified indirectly. Conversely, explicitly marking the electronic resource unreachable reveals firewall in place. The adversary thus applies probabilistic or deterministic logic: she either assumes perimeter defenses are present judging from missing response (probabilistic), or from a reply she actually received (deterministic). Various protocols have been devised to interconnect devices and standardize their behavior, but authorization and authentication are rare, and incoming packets are treated as harmless until marked as malicious server-side. This moves the threat closer to the victim who addresses it alone or in collaboration with Internet Service Provider (ISP). The tools in the case study benefit from such permissiveness not originally intended to facilitate reconnaissance and fingerprinting. The behavior will not substantially change over time as the amount of hardware and software relying on backward compatibility is considerable.

Lastly, the topic is of interest because it demonstrates that very little technical prowess is required on part of the intruder to execute campaigns with financial and business continuity repercussions. Any user can launch potent scenarios which force applications into unhandled states and allow executing arbitrary code for gaining system persistence, covert user surveillance, search results redirection, and using the compromised node as a stepping stone to reach internal hosts with sensitive data. The tools and specialized operating systems are available free of charge which leave marginal digital trail of evidence. Even though the programs are in some cases not trivial to operate, ready-made tutorials can be found and applied almost verbatim. We predict the results will corroborate that hardening ICT infrastructure including human element from basic attacks is underestimated.

In the case study, methods and results will not be decoupled and separated but recounted sequentially for maintaining flow and presentation integrity. The final phase will thus not aggregate results but just a brief conclusion based on findings in previous sections. Furthermore, reconnaissance and VA tools will be briefly assessed right before their first use for uninterrupted problem→method→solution→result sequence. Their thorough inspection is not within the thesis’ purview: help documentation and online sources cover most topics, CLI switches, data outputs as well as viable test cases extensively. Some networking terminology will be clarified without aspiring for exhaustive treatment, and will omit most details in favor of brevity and succinctness. Screen shots will be sanitized to leave out information which could identify vulnerable network resources and other sensitive data, as outlined in chapter 3. The case study will not result in a PT report assembled after a real-world test is concluded because each result will include recommendations to eliminate or mitigate risk associated with the particular vulnerability. Grouping the steps into a single document afterwards would be equal to producing a full report.

The last methodological note is about the preference for VA and PT in the case study. As established earlier, PT goes beyond VA in that the identified vulnerabilities are exploited as proofs of concept while referencing pertinent disclosures and advisories verified and tested in production environment, often accompanied by subroutines or program snippets. VA is a subset of PT which does not validate the vulnerability claims directly but through external resources

from third parties (vulnerability databases). The case study will primarily use VA as a method of choice: if a member of the target's IT staff deems the host under investigation is inessential for business continuity, PT will be attempted after prior notification. If system access is obtained, further tests will be abandoned because the vulnerability was provably demonstrated as leading to a compromise. Penetration testing will therefore be used scarcely, and vulnerability assessment along with interpreting output from the selected tools will be the main focus. This will supply plethora of information about perimeter and internal ICT infrastructure from the point of view of malicious insider, i.e., gray-box testing. The model simulates an agent with limited access, such as an employee who fingerprints services and gathers data to find exploitable weaknesses.

The same hardware setup from the questionnaire research and case study 1 will be used. No taxing computations are expected to take place with neither CPU nor RAM being performance bottlenecks. The components left out previously were network interface controller (NIC) which transmits and receives packets (chapter 2.1.1 mentioned network communication is handled using packets and the terminology will be obeyed). The following specifications are relevant for the purposes of the case study:

- NIC model: Realtek RTL8191SE Wireless LAN 802.11n PCI-E NIC,
- NIC driver version: 2020.4.620.2011 (20.06.2011),
- NIC supported modes: IEEE 802.11b/g/n.

Google Chrome (Chrome) web browser will be used for taking screen shots as it was the most widely-used in the questionnaire research (chapter 4.2.1). Major competitors (Internet Explorer, Mozilla Firefox, Opera, Safari) do not result in noticeable differences. Chrome development team opted for rapid release schedule which sees new iterations come out as frequently as 6 weeks with update engine initiating a download and installation automatically if a difference is detected between local and remote version numbers. While this leaves the browser version number unfixed, the case study does not depend on specific browser type and version.

Software base will consist of a Linux-based penetration testing distribution, Kali Linux². A Debian-architecture successor to a PT platform BackTrack Linux, it was first released as a disk image (.iso) on March 13, 2013 with a collection of tools exhaustively covering the three testing phases defined earlier. The system will be run in a virtualized environment and installed for storage persistence; an alternative, Live disc, does not retain any files permanently which would complicate analyses. Despite being proven vulnerable to exploits (chapter 2.4.6), virtualization is a preferred way to disallow changes from influencing the underlying system. Specifications of the virtual machine are as follows:

- OS substrate: Windows 7 Professional Service Pack 1 64-bit,
- browser: Chrome (stable channel) 32-bit with automatic updates turned on,
- VOS: Kali Linux 32-bit ISO,
- virtualization manager: Oracle VM VirtualBox v4.2.18.r88780,
- CPU cores: 1,
- HDD: 20GB (dynamically allocated),
- RAM: 1536 MB,
- NIC: Intel PRO/1000 MT Desktop (NAT enabled),
- additional: PAE/NX enabled.

Kali Linux will be occasionally referred to as a virtual operating system (VOS) due to it being isolated in a container without direct access to the OS substrate. VirtualBox allows bidirectional file transfers which makes exporting results for analyses in Microsoft Windows trivial.

²<http://www.kali.org/>

5.2.2 Phase 2: Information Gathering, Reconnaissance

At the end of the previous phase, Kali Linux VOS was installed into a VM and configured to use a portion of system resources. We assume this is still within the attacker's capabilities as tutorials are readily available and the steps intuitive. Before the information gathering (reconnaissance) process is started, a brief overview of the target is warranted as it narrows down the scope of the study and allows to direct efforts toward particular vectors. Focusing on predictable set of entry points is beneficial to IT administrators because the attack surface is reasonably pruned. However, it also poses a threat because the adversary may prefer overlooked vectors. An audit of all front-facing interfaces should be conducted because it is one of the first steps the attacker performs. Anticipating such action, system operators can deploy suitable countermeasures. Low-skilled agents may even get deterred at their lack of progress when no foothold is gained initially, and move on to a different target. The same cannot be expected in case of insider threat (gray-box testing model): the malicious party has a singular purpose to harm the organization using their partial knowledge of network topology and ICT infrastructure and cause maximum damage while remaining undetected or at least having the ability to claim plausible deniability.

The target was selected to be ICT infrastructure of the Faculty of Management and Economics, Tomas Bata University in Zlín (FaME). Basic information pertaining to the FaME are:

- address: Mostní 5139, 760 01 Zlín, Zlín, Czech Republic (GPS: 49.220922,17.657292),
- employees: 113 (December 1, 2013),
- employees, students: 123, 3 123 (March 1, 2013),
- URI³: <http://www.utb.cz/fame/>.

Prior to VA commencement, IT staff was contacted and an agreement sought for the tests. While an NDA was not signed, the author consented to report the findings in advance to close the identified attack vectors before releasing the information publicly. Also, no data threatening the CIA triad at the time of writing will be disclosed unless vetted by the IT staff first. The methodology outlined in chapter 3 will be followed and screen shots altered to redact sensitive information. The last requirement imposed on the VA is that it will be restricted to the FaME: third-party servers and services constituting viable attack vectors will be excluded. IP addresses of internal servers were received but will not be included. The information established the gray-box model's prerequisites of incomplete knowledge related to target ICT. A range of external IP addresses for accessing FaME website was also received, but Kali Linux has facilities to query Réseaux IP Européens Network Coordination Centre (RIPE NCC), a European registry for IPv4 and IPv6 address allocation.

We predict the testing will take several weeks, with information gathering and fingerprinting taking up much of the time. Reconnaissance is sometimes deemphasized in practice at the expense of producing immediate results. Enumerating totality of the attack surface should be given equal importance, though, particularly in cases where third parties interact with the target's information systems. While VA and PT may be restricted to a single ICT infrastructure, the adversary may utilize the uncontrolled interconnects to piggyback into the internal network. If the analyst is granted full privileges, she can uncover and preempt threats the organization did not address previously, e.g., obsolete supplier accounts with default or weak passwords for low-level access which can nevertheless be exploited. FaME, a tertiary educational institution, is bound to have a database of students and alumni for record-keeping purposes; should it be stored on the same virtual machines as user accounts, the perpetrator could traverse and reach it. Primarily a domain of PT, we predict privilege escalation vulnerabilities will not be seen in

³Uniform Resource Identifier

the case study. This does not lessen the importance of compartmentalizing running instances to separate VMs based on the CIA criteria:

- confidentiality: instances containing sensitive electronic assets should be run on two different sets of hardware,
- integrity: each instance should have at least one replica on an independent physical server against which inconsistencies are tested and corrected,
- availability: mission-critical instances should not be placed on the same set of hardware which turns it into a single point of failure.

Insiders have some information about the hardware and software, and need not to interact with the target to obtain them which would possibly reveal their actions early on. Direct observations may reveal plenty of actionable intelligence and are perfectly stealthy unless steps are taken which other employees view as suspicious or unacceptable, e.g., entering offices and using computers located there without permission of the owner or querying about login credentials and security habits, although this can be ameliorated by social engineering. The threat stems mostly from raising suspicion, but repercussions are minimal due to users' mindset of not reporting anomalies to IT personnel. Physical presence is classified low-threat when obtaining background information, particularly in educational environment where incoming parties are not asked to identify themselves and are automatically categorized as students or visitors. Authentication is required to enter certain off-limits premises and access domain accounts (circumvented by logging in locally without domain credentials), but anyone can otherwise inspect and observe the premises. Through direct observation and knowledge of internal processes, the following was discovered as of December 31, 2013:

- Users at the FaME are assigned either desktop personal computers or notebooks with uniform hardware configurations which can be freely dislocated from the building for working remotely.
- Offices are locked and keys marked with unique IDs which makes their duplication without permission a legal offense. A master key exists which can be requested by authorized parties under plausible pretext.
- The default OS is Microsoft Windows 7 Professional; Microsoft Windows XP has also been seen but is gradually phased out due to approaching the extended support deadline. The preferred office suites are Microsoft Office 2007 and 2010.
- AVG Anti-Virus Business Edition with automatic updates turned on is installed on every computer by default. While alternatives can be deployed, homogeneous software base and little deviations from pre-selected configurations can be expected.
- Users can select their own browser. No restrictions are placed on running additional software as long as it does not constitute copyright infringement, but preventative measures and checks are non-existent. Microsoft Windows 7 offers Internet Explorer 8 by default, Microsoft Update upgraded the version to 11, the latest one available for the OS.
- Domain management is based on endpoint Novell clients and obsolete NetWare OS. Migration to Active Directory has been announced and is scheduled to take place in 2014 with transition and testing periods likely spanning several months. The move suggests Windows Server with Lightweight Directory Access Protocol (LDAP) support will be deployed.
- If the user opts out of logging in to the domain, they can bypass the Novell authentication mechanism from the Welcome Screen. The default password for local workstation access is an empty string and simply pressing Enter is sufficient to log in. Even though remote printing capabilities and shared network volumes are disabled, standard functions are available.
- Local workstation privilege level group was set to Administrator for every account on PCs and notebooks by default. The superuser can perform system-wide changes. Windows 7

includes User Account Control (UAC) security technology which blocks untrusted sources from interacting with the file system until explicit administrative override lifts the block, but superuser can disable UAC entirely.

- Periodic OS update checks are not enforced and can be disabled.
- Password generation and revocation policies are not centralized. Email accounts have default password set to coincide with user's birth number of the form YYMMDD/SSSC (alternatively YYMMDD/SSS) where YY denote year, MM month, and DD day of birth; SSS are numbers uniquely identifying people born on the same day, C a check digit which ensures the whole string is divisible by 11. Password management is left up to individuals with a reasonable assumption majority of credentials are left unchanged.
- Personal information about academic employees are available on the FaME website. Usernames are hard coded and consist of lowercase-only person's surname, e.g., doe for John Doe; if several identical surnames exist, a two-letter string consisting of the first character of the person's name and a number is prepended, e.g., j1doe.
- Smart mobile devices (detailed in chapter 2.3) procurement and management is decentralized to the department level. Smartphones and tablets lack profiles and price is the primary purchase criterion, strongly suggesting many run outdated Android OS releases. Phones are handled to employees without security briefing and best practices manual.
- BYOD management is not taken into account: any device can remotely interact with the email service and information system on unsecured wireless networks without restrictions. While smartphones offer VPN connection establishment and the FaME supports the mode, we assume many users prefer direct access over the Internet.
- The faculty utilizes university-wide SAP ERP (Enterprise Resource Planning) information system. We can assume a Linux or Unix-like server is the OS powering the platform due to high penetration rates of Linux in server environment, reliability, and customization options. A particular distribution cannot be hypothesized about but SAP ERP is a mission-critical resource with serious consequences in case of disruptions: a stable, security-oriented flavor with conservative approach to updates, slower release cycles, and wider penetration base such as CentOS, FreeBSD, or OpenBSD would be consistent with these assumptions.
- Moodle (Modular Object-Oriented Dynamic Learning Environment) pre-2.0 e-learning tool is with high probability run from a VM locally. Teachers and students are strongly encouraged to create accounts in a domain-separate database instance. However, we argue at least some level of credential reuse, i.e., usernames, passwords, or both are shared between Moodle and Novell.
- The university is a member of the Czech Education and Scientific Network (CESNET) and the campus hosts a wireless Internet mesh, eduroam (educational roaming). Students and employees have to select unique, Novell-independent credentials usable for connecting at all participating institutions. Password management policies are not enforced.
- Each computer on the internal network is assigned a unique identification sequence: for notebooks, room number is included while for desktops, owner's surname is used. When the device is passed on and the OS is not reinstalled, the obsolete string remains in place. Moreover, notebooks moved to other rooms retaining the original IDs, making this form of identification prone to errors.
- VoIP (Voice over IP) endpoints have been deployed at the university, and telephony thus relies on low-delay, low-jitter, and low-loss networks to operate correctly. Any variations in these properties cause noticeable quality degradation to the point where the service becomes unusable.
- In 2013, website of the FaME underwent redesign and was migrated to the current address with the old website no longer updated and maintained. However, many resources are still

reachable including those which have not been ported over.

Several of the findings can be used in scenarios to gain foothold into the system or disrupt infrastructural services users rely on without considering their susceptibility to security breaches and outages. Those will be further expanded in chapter 5.2.3. While some are theoretical, many vectors can be exploited with relatively low resources.

The reconnaissance phase benefits from the gray-box model because otherwise, many information would have needed to be obtained by alternate means. Search engines have automated indexers which create website snapshots retrievable even after the original asset is unreachable. This allows to query for information, scrape document metadata as well as inspect directory structures while avoiding direct interactions with the target. Issuing the directive `filetype:doc site:fame.utb.cz` into Google, the search engine of choice for data retrieval, returned files ending in `.doc` found on the old website which redirects to the new portal and should not be available. Analyzing the results, a directory was located storing employee CVs and publication histories as far as 2008. Figure 84 demonstrates the listing.

Index of [REDACTED]




















<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory			-
 anna.zapletalova_osob.doc	20-Oct-2011 14:59	27K	
 anna.zapletalova_publicace_web.doc	06-Jun-2013 15:31	20K	
 babkova_osob.doc	19-Feb-2009 15:24	21K	
 baresova_osob.docx	06-Jun-2013 15:31	13K	
 baresova_publicace_web.docx	06-Jun-2013 15:31	11K	
 bejtkovsky_osob.doc	16-Dec-2008 16:54	29K	
 bejtkovsky_publicace_web.doc	28-Feb-2012 08:02	185K	
 belas_osob.doc	21-Mar-2011 09:55	58K	
 benda_osob.pdf	16-Feb-2009 14:25	176K	
 benda_publicace_web.pdf	16-Feb-2009 14:25	152K	
 beranova_osob.pdf	16-Feb-2009 14:25	176K	
 beranova_publicace_web.pdf	16-Feb-2009 14:25	151K	
 bialic_publicace_web.doc	30-Mar-2013 18:39	46K	
 blahus_osob.doc	13-Nov-2012 08:27	32K	
 blahus_publicace_web.doc	28-Feb-2012 08:01	24K	
 bobak_osob.doc	17-Feb-2009 12:52	444K	
 bobak_publicace_web.doc	17-Feb-2009 12:52	84K	
 bocincova_osob.doc	16-Dec-2008 16:54	23K	

Fig. 84: Target directory listing. The header was redacted for security purposes as it was reachable at the time of writing.

Source: Own work.

While the information may seem innocuous, they nevertheless provide several important clues: last names of all employees from 2008 to 2013 inclusive which are exploitable for harvesting domain credentials; Microsoft Office 2007 in active use; and a Portable Document File (PDF) virtual printer installed on some stations (PDFCreator). Inclusion of `(.php)`, a server-side scripting language file, establishes target's preference for PHP instead of nginx, another widely-deployed web server software. Lastly, since the virtual directory leaks OS and HTTP server names, the adversary knows the former to be Debian GNU/Linux and the latter Apache HTTP Server 2.2.16. Exact version number allows to tailor a malicious payload by specifically targeting

vulnerabilities patched in later releases. She can also surmise beyond the intelligence gathered: chapter 2.4.7 referenced open-source toolchain for dynamic website creation and management, LAMP: because it has already been corroborated target infrastructure relies on Linux (L), Apache HTTP Server (A), and PHP (P), a reasonable assumption is to expect either MySQL or MariaDB (M) database back-end to complement the stack. Even though only a hypothesis, the evidence is in line with streamlined implementation the LAMP is praised for when deployed in its entirety. Deviations would mean risking incompatibilities and potential stability issues. Both MySQL and MariaDB are based on the relational model and share many similarities: focusing on exploits for one has high probability of affecting the other, too.

Modified directives which request files with particular extensions led to additional findings. In fact, multiple student lists were extracted together with another virtual PHP scripts directory. While none present imminent threat by itself, analyzing and collating them reinforce background for social engineering campaigns. Referencing past events builds trust with the recipient who is more likely to comply with requests within acceptable bounds, e.g., opening an infected document under the pretense of requesting victim's professional opinion.

Another search engine directive revealed all indexed webpages with the keyword "admin" in title, indicating presence of administrator interfaces used for remote access. The search returned two results for sites hosted on the old webpages. Although no further action was attempted, parsing a custom string could cause SQL injection for remote database interactions. Even though both sites were abandoned in favor of newer versions, these uncontrolled interconnects could lead to sensitive data leaks, e.g., email addresses, passwords, and usernames should they be exploited. The second result was an administrative dashboard cutout of which is depicted in Figure 85.



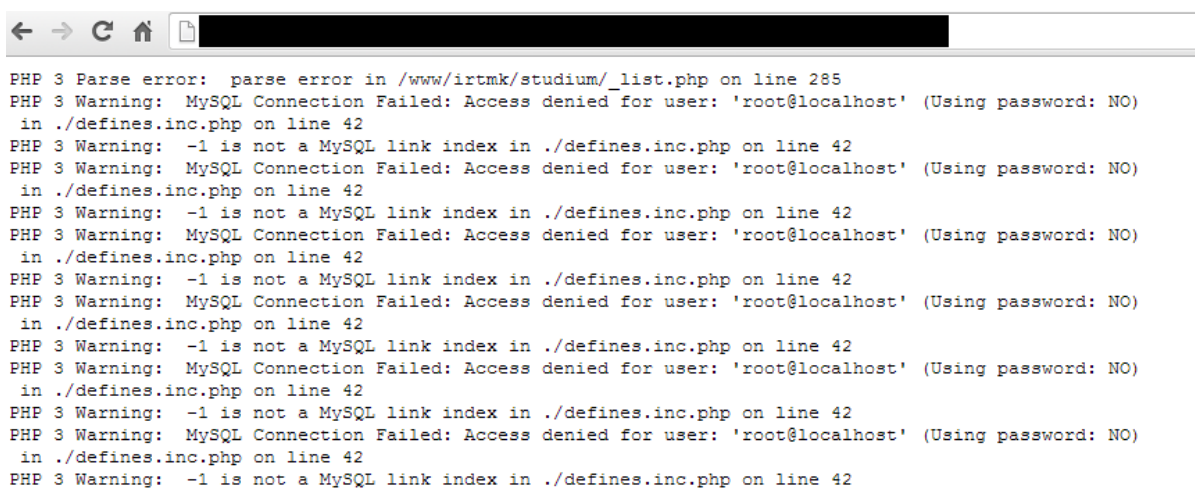
Fig. 85: Administrative panel disclosure. Font rendering issues are caused by improper handling of PHP file character encoding most likely saved as ISO-8859-2 at source but forced by the server to be displayed using UTF-8.

Source: Own work.

The panel provided a final confirmation the complete LAMP stack was indeed deployed: phpMyAdmin is a tool designed to manage MySQL databases through a web interface. The link redirects to a login page leaking version number (2.9.2), and possibly vulnerable to SQL injection. Administrator account, despite its high-privilege status, is treated identically as any other user entity and is prone to the same attacks which threatens the system. Interestingly, the

panel also made available a PHP error log counting 264464 entries. Figure 86 shows its header with the first few lines.

Log security should be prioritized because data aggregated in a single place can be used to reconstruct asset naming schemes, directory structures, error logging granularity, loaded modules, parsing errors leading to improper query handling (SQL injection), supported and unsupported functions, system parameters (execution time limits), table names, and others. While all serious, leaking a list of loaded modules is particularly dangerous because it can be used to tailor exploits for known software, thus increasing attack potency. We can only hypothesize why the logs were made reachable from the Internet without authentication but comfortable remote access without the need to log in can be assumed as one of the reasons. The very last link shown in Figure 85 leads to a page where three errors are displayed along with an absolute path to a .php file in a Linux-based OS (earlier identified to be Debian) which hierarchically lists all directories up to and including the filename itself.



```
PHP 3 Parse error: parse error in /www/irtmk/studium/_list.php on line 285
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
PHP 3 Warning: MySQL Connection Failed: Access denied for user: 'root@localhost' (Using password: NO)
in ./defines.inc.php on line 42
PHP 3 Warning: -1 is not a MySQL link index in ./defines.inc.php on line 42
```

Fig. 86: PHP error log disclosure. The log consists of human-readable plaintext file with each entry corresponding to a database interaction. Source: Own work.

Historical website analysis provided us with information about the software infrastructure the target either runs locally or shares with other faculties. We identified server OS, all components of the LAMP stack, in some cases down to version numbers, as well as office suites and utilities installed on endpoint machines of some users. Before moving on, a brief enumeration of the current portal was conducted. The target is closely tied with the university ICT-wise as evidenced by the administrative panel in Figure 86 which lists reports for all faculties. Expanding the search to www.utb.cz may lead to information pertaining to IT infrastructure shared by the constituents.

Of note is that information gathering has been partly based on historical data from virtual directory listings in an environment no longer kept updated. To ascertain the data is not obsolete, an online software fingerprinting tool titled Netcraft was queried for then-current information. The results are presented in Figure 87.

The query returned version numbers for Apache HTTP Server and PHP both of which were obsolete at the time of writing. Due to their wide deployment in production environments, vulnerabilities are frequently discovered and patch deployment for the front-facing components of the LAMP stack should therefore be assigned high priority. Keeping the infrastructure updated relegates the adversary to zero-day exploits which deter low-skilled individuals because the knowledge required is beyond their abilities. Nevertheless, keeping up with release cycles from

Network

Site	http://utb.cz	Netblock Owner	Tomas Bata University
Domain	utb.cz	Nameserver	sun.utb.cz
IP address	195.178.88.67	DNS admin	hostmaster@utb.cz
IPv6 address	Not Present	Reverse DNS	moon.utb.cz
Domain registrar	nic.cz	Nameserver organisation	whois.nic.cz
Organisation	Univerzita Tomase Bati ve Zline, nam. T.G.Masaryka 5555, Zlin, 760 01, Czech Republic	Hosting company	utb.cz
Top Level Domain	Czech Republic (.cz)	DNS Security Extensions	unknown
Hosting country	CZ		

Hosting History

Netblock owner	IP address	OS	Web server	Last seen
Tomas Bata University, part 1 Zlin	195.178.88.67	Linux	Apache/1.3.33 Debian GNU/Linux PHP/4.3.10-22	14-Nov-2013

Fig. 87: Partial Netcraft fingerprinting output. The service acts as a proxy by making requests from a dedicated pool of IP addresses. The attacker does not send any packets to directly interact with the target.

Source: Own work.

different vendors is challenging as the pre-deployment testing phase needs to be repeated for every program to ensure stability and backward compatibility. Dependencies mentioned in chapter 5.1 may remain dormant until a system state is reached which results in an error.

Entering the old address, www.fame.utb.cz redirects the browser to www.utb.cz/fame, most likely accomplished by a Domain Name System (DNS) modification. The system provides services which “. . . have an important function in helping users readily access the many resources that are available through the Internet. [They] make communications convenient for the user by translating the unique resource identifier that is known as the Internet Protocol (IP) address into a domain name that is easy for the user to remember. . . . The DNS transforms human-readable domain names into machine-readable IP addresses and also does the reverse process, taking a query with an IP address and returning the domain name associated with it” (Radack, 2006). A snapshot of the old site in the search engine cache will be accessible, though, unless a request is generated to remove the contents permanently, or letting the data expire gracefully. The process lasts several months until the automated indexer scans the page again and receives a notice the resource was relocated or is unreachable.

The portal on <http://www.utb.cz/> was analyzed next. Changing the default language from Czech to English revealed a `.php` script with a `text=` parameter (1 for Czech, 2 for English), confirming at least PHP continues to be utilized. Login screen contains a message that the credentials are identical to those for the Novell domain, leaking information about the software. Both findings corroborate previous discoveries and lends credibility to the assumption vulnerabilities related to LAMP stack patching delays are present. The majority of links do not use explicit PHP script declarations which should mitigate certain attacks. File download links also do not expose the underlying PHP but include username of the employee who uploaded the asset. Because they have a distinct pattern, the perpetrator can infer university-wide credential assignment policy which coupled with the observation regarding default passwords threaten to expose email and domain accounts. Documents are not scrubbed for metadata and reveal programmatic means and Internet services for producing PDF files. Lower-level links to employees’ personal pages still use potentially exploitable parameters.

Internal redirects pointing to a login screen in case the resource was not intended for outside parties leak name of another tool for web applications: xoops. Only compatible with MySQL database system, we confirmed the final LAMP stack component. The portal has administrator interface, sometimes accessible by appending /admin or similar string behind the address, disabled as well as generic error pages when illegal address is entered. On the old website, they leaked information about the OS type and PHP version number.

An institution funded from public sources, Tomas Bata University is legally bound to release annual reports covering major educational, scientific, and technological milestones during the previous year. The last are of particular interest because they help reconstruct and track technological developments. The reports confirm ICT infrastructure has been gradually shifting toward centralized solution off the target premises, preserving only services such as application license servers, local antivirus definitions updates, economic software, and closed-circuit television information system. This will make differentiation between valid and off-limits assets challenging. Overall, the information supports the evidence obtained from direct observations on multiple points such as SAP being the information system of choice. The sole erroneous assumption about a Linux distribution was corrected in a report from 2011 which mentioned HP-UX, a proprietary HP Unix implementation, to be the underlying OS platform.

Email addresses on the target website are not obfuscated by any mechanism, e.g., CAPTCHA (chapter 2.1.1), server-side encryption, encoding the strings as pictures, and are prone to automated harvesting. Since local parts of email addresses correspond to domain usernames, the attacker can simply run a regular expression to remove the domain part: `doe@fame.utb.cz` is transformed into `doe` and can be amassed to launch brute-force attack against email and domain services. A secure alternative would be to decouple local parts from domain names using a non-trivial method, e.g., a per-user generated sequence, and a lockout period after several failed attempts to access an account. Even though chapter 2.4.2 claimed the technique can be exploited to lock legitimate users by entering bogus passwords until a lockdown procedure is initiated, CAPTCHA prevents such situations: when the threshold is reached, a challenge-response authentication must be successfully passed, otherwise no more data will be accepted from the client. Security is dependent on password strength and hardness of the CAPTCHA.

Concluding the website analysis, we will now use Kali Linux. DNS has already been defined as a service which allows translation of human-readable names to numerical values called IP addresses. Every device which complies with the Internet Protocol is assigned at least one unique address identifying it for other hosts on the network. Kali Linux has a reconnaissance tool, TheHarvester, which can scrape the target for valid email addresses if they are in plaintext (the fact was previously established), and resolve subdomains. An email address list extracted by inputting `fame.utb.cz` as the target host is depicted in Figure 88.

Figure 89 shows hosts identified by brute-forcing the entire IP range within which the target host resides. Resolving each subdomain, two were found to point to unavailable resources. Purging obsolete entries from DNS prevents data leaks about the network, and even though they pose only a marginal risk, DNS management should incorporate scavenging old data. For larger organizations, neglecting the step could cause slowdowns from expensive database searches and IP address blocking. So far, information gathering did not use any malicious technique. The process relied on DNS which supplies the requested information to anyone without authentication, a weakness of the TCP/IP protocol suite discussed earlier. For the sake of completeness, RIPE NCC database will be queried for data. Kali Linux has the `whois` command, but the output from the official authority is more verbose. The result is demonstrated in Figure 90.

The `inetnum` entry denotes complete IP footprint of the university on the Internet; while we restrict ourselves to a single target, the whole range should be scanned for live hosts with prior

```
root@kali: ~
File Edit View Search Terminal Help

[+] Emails found:
-----
popesko@fame.utb.cz
benda@fame.utb.cz
knapkova@fame.utb.cz
drimlova@fame.utb.cz
pastuszkova@fame.utb.cz
bialic@fame.utb.cz
nibedita@fame.utb.cz
studium@fame.utb.cz
zimola@fame.utb.cz
jenys@fame.utb.cz
michlova@fame.utb.cz
gregar@fame.utb.cz
striz@fame.utb.cz
pasekova@fame.utb.cz
hromkova@fame.utb.cz
hajek@fame.utb.cz
bris@fame.utb.cz
brazdilova@fame.utb.cz
pavelkova@fame.utb.cz
nghaihang@fame.utb.cz
homolka@fame.utb.cz
dobes@fame.utb.cz
rosman@fame.utb.cz
friedel@fame.utb.cz
chodur@fame.utb.cz
ppalka@fame.utb.cz
steker@fame.utb.cz
zdohnalova@fame.utb.cz
sasinkova@fame.utb.cz
otrusinova@fame.utb.cz
zavodna@fame.utb.cz
pilik@fame.utb.cz
novosak@fame.utb.cz
mhorakova@fame.utb.cz
```

Fig. 88: *TheHarvester* email address list. While the output is not exhaustive, the accounts can be exploited to trivially generate domain usernames. Redaction was not needed as every address is publicly accessible.

Source: Own work.

permission. It is expected Internet-facing resources are adequately protected but the internal network may be open to exploitation by malicious parties.

Kali Linux also incorporates a tool which automates metadata (chapter 2.2.2) extraction, *metagoofil*. Specifying a domain, file extensions and how many objects should be downloaded and analyzed locally, the utility proceeds to request the files and extracts strings from fields to generate a report. A sample output is provided in Figure 91.

Metagoofil was instructed to download exactly three files with `.xls` extension and parse them. The search space consists of all documents hosted on the university portal because the target website cannot be referenced directly as it resides on shared ICT infrastructure and does not have a dedicated subdomain. This pollutes the results and necessitates additional filtering.

It has been already established local parts of email addresses are used as usernames; a more streamlined approach would be to automatically collect them via *TheHarvester* or *Metasploit Framework*, an extensive open-source platform built into Kali Linux, and perform a search for `@fame.utb.cz` to separate the results of interest. Structuring the trimmed-down strings one per line into a plaintext file would then serve as input for online brute-force or dictionary reverse engineering. Combined with the information that birth numbers are assigned as passwords by

```

root@kali: ~
File Edit View Search Terminal Help

[+] Hosts found in search engines:
-----
195.178.88.75:web.fame.utb.cz
195.178.88.73:olympiada.fame.utb.cz
195.178.88.120:vyuka.fame.utb.cz
195.178.88.73:absolventi.fame.utb.cz
195.178.93.91:striz7.fame.utb.cz
195.178.88.73:www.kancelar.fame.utb.cz
195.178.88.73:www.fame.utb.cz
195.178.93.92:striz8.fame.utb.cz
195.178.88.67:sumec.fame.utb.cz

[+] Starting active queries:
[-] Performing reverse lookup in :195.178.88.0/24
    195.178.88.255[-] Performing reverse lookup in :195.178.93.0/24
    195.178.93.255
-----
195.178.88.72:nw-fame.utb.cz
[-] Starting DNS TLD expansion:
root@kali:~#

```

Fig. 89: *TheHarvester* subdomain list. Some subdomains are no longer available but DNS has not been updated to reflect the changes.

Source: Own work.

```

Search results

% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf
%
% Information related to '195.178.88.0 - 195.178.95.255'
% Abuse contact for '195.178.88.0 - 195.178.95.255' is 'abuse@utb.cz'

inetnum:          195.178.88.0 - 195.178.95.255
netname:          UTB-T34CZ
descr:            Tomas Bata University
descr:            Zlín
country:         CZ
org:              ORG-TBUI1-RIPE
admin-c:          TBUI1-RIPE
tech-c:           TBUI1-RIPE
status:          ASSIGNED PA
mnt-by:           TENCZ-MNT
remarks:          Please report network abuse -> abuse@utb.cz
changed:          tkpv@cesnet.cz 19970321
changed:          tkpv@cesnet.cz 20131011
source:          RIPE

```

Fig. 90: Partial RIPE NCC WHOIS query output. Sections redundant for the purposes of the case study were edited out for brevity.

Source: Own work.

default, the candidate pool can be pruned substantially with marginal reduction in the success rate. TheHarvester-powered email extraction was demonstrated earlier.

The concluding step of the second phase will be to gather information about the email system. In 2013, a migration to a new platform was completed with both services running simultaneously for what is assumed to be a transition period. The old front-end, `webmail.utb.cz`, is powered by SquirrelMail, an open-source application. Figure 92 depicts the login page along with information about a digital certificate the site uses for data encryption.

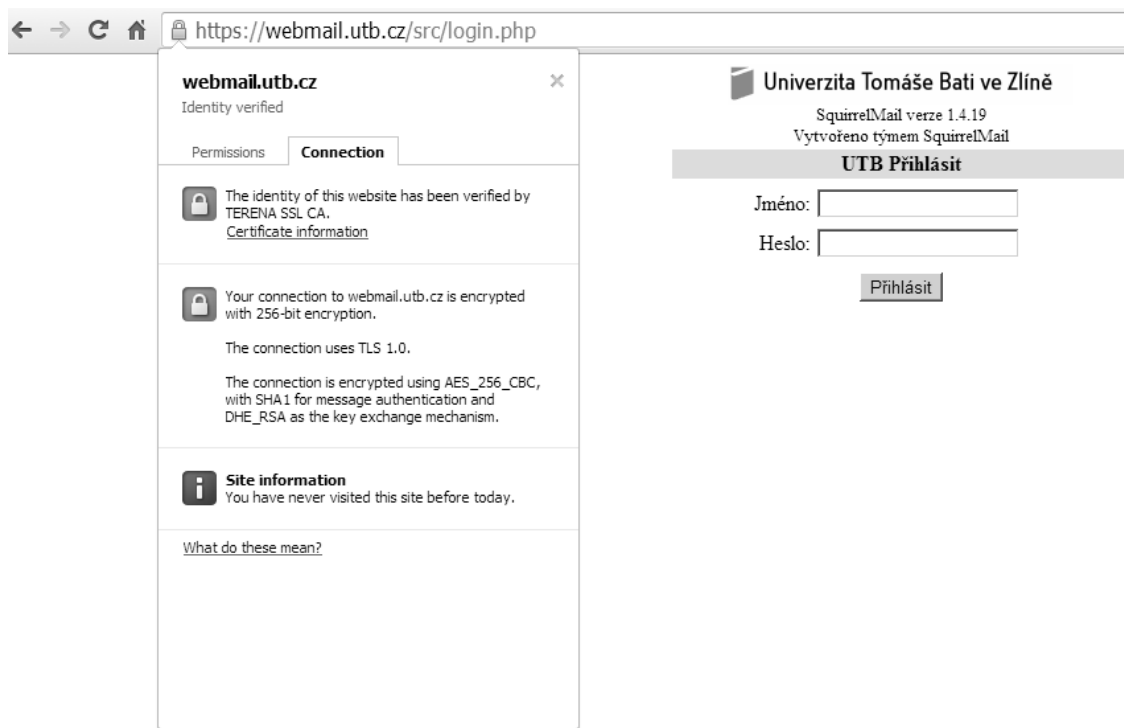


Fig. 92: SquirrelMail online login screen. An Extended Validation (EV) digital certificate information panel is displayed which lists details about the Certificate Authority (CA) and the encryption scheme for sending data to and from the server.

Source: Own work.

hamper security, particularly when critical vulnerabilities are discovered and patched. Suhosin partly leverages the disadvantage but should not be relied on as the sole protection for PHP.

The new email platform does not include any additional HIP mechanisms when accessed from outside the university IP range. Switching from SquirrelMail to a free PHP-based client, Roundcube, the platform was at the time of analysis operated on Debian Linux 7.2, Apache HTTP Server 2.2.2, and PHP 5.4.4. Login screen with a certificate is depicted in Figure 93.

The front page does not leak any information about the Roundcube version number which makes it challenging to tailor exploits for a particular release. However, active exploits for Roundcube exist and are hypothesized to increase in volume as it becomes picked by more organizations. We have already established presence of a MySQL instance in the target's ICT infrastructure; it is reasonable to assume Roundcube uses it as well due to native support. Users can access their emails on the premises from a client of their choosing: both Microsoft Outlook included in the Microsoft Office suite and Mozilla Thunderbird were spotted, but the former seems to be preferred. A freely-available manual was discovered on the target website and the following information extracted:

- Post Office Protocol (POP) 3 and IMAP are mapped onto ports 995 and 993, respectively; address: imap.utb.cz,
- SMTP is mapped onto port 25; address: smtp.utb.cz,
- outgoing emails are not encrypted.

The first two points eliminate uncertainty with port numbers, the third hints at a viability of passive data interception if users are connected to the eduroam wireless network while sending emails. With insider threat, the adversary is assumed to have legitimate access to her own email account and is therefore able to send and receive messages. When passed over the network, email

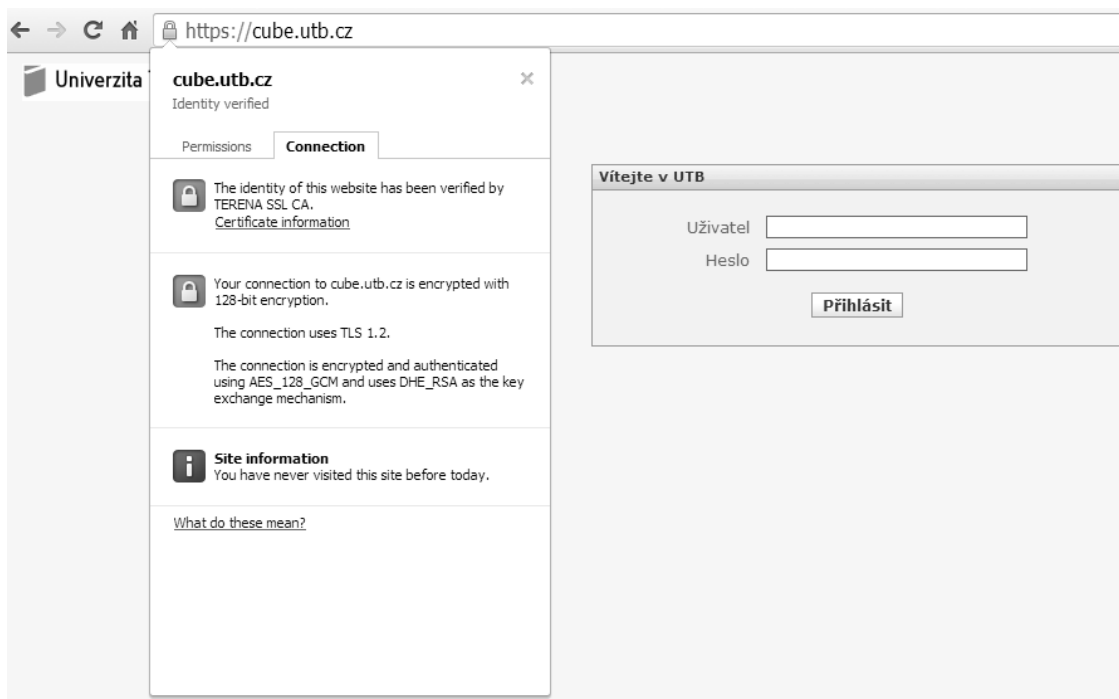


Fig. 93: Roundcube online login screen. The EV certificate supports TLS 1.2, the latest protocol version available which is recommended for production environments.
Source: Own work.

metadata are modified in transit to include various information. Users do not have to review nor understand the data because it is hidden and accessible only on explicit request. The attacker can identify software which processed the message by name, version number, and additional information which could help tailor social engineering campaigns while bypassing heuristic-based signatures. After experimenting with loop-back sending, it was concluded that messages sent from the target's faculty address to an address within the same domain (@fame.utb.cz) are not subjected to the same scrutiny as external domain names. For example, an email with an executable (.exe) and a batch file (.bat) attached was correctly received at the destination, bearing no overt marks of being flagged as suspicious. On the contrary, if a suspicious outside message arrives, it is clearly marked as spam by prepending its Subject field with "[SPAM]." However, the email is still passed to the recipient and not automatically rejected. A partial header for a message classified as spam is shown in Figure 94.

Any insider can view the information by sending a loop-back email where the sender and the receiver are identical. The data establishes a few facts about the infrastructure which will be described in order in which it appears in the header. First, a server chain through which the email is passed can be fully reconstructed: nod32.utb.cz→sun.utb.cz→mailbox.utb.cz. Issuing whois in Kali Linux, it was found the first subdomain is an alias for the second; the sequence is therefore simplified to sun.utb.cz→imap.utb.cz. The university name server has built-in ingress/egress antivirus protection capabilities supplied by NOD32, specifically ESET Mail Security for Linux/BSD suite. Version number cannot be ascertained but Debian Linux fingerprinted earlier is very likely the underlying OS platform.

Another two programs the header references are Postfix and SpamAssassin. The former is a mail transfer agent (MTA) which facilitates convenient handling of SMTP-compliant sending and receiving, SpamAssassin is an email spam filtering agent which employs various heuristic checks to classify incoming messages. The header leaks SpamAssassin's version (3.2.3) which was released in 2011 and is listed as obsolete on the official website. Because it only provides the

```

Return-Path: <dr.xie05@vip.126.com>
Delivered-To: sarga@mailbox.utb.cz
Received: from sun.utb.cz (unknown [192.168.1.13])
    by mailbox.utb.cz (Postfix) with ESMTP id E586B78348F3
    for <sarga@mailbox.utb.cz>; Mon, 18 Nov 2013 10:18:36 +0100 (CET)
Received: from sun.utb.cz (localhost [127.0.0.1])
    by nod32.utb.cz (Postfix) with ESMTP id CD48E3409419F
    for <sarga@mailbox.utb.cz>; Mon, 18 Nov 2013 10:18:36 +0100 (CET)
X-Virus-Scanner: This message was checked by ESET Mail Security
    for Linux/BSD. For more information on ESET Mail Security,
    please, visit our website: http://www.eset.com/.
Received: by sun.utb.cz (Postfix, from userid 1000)
    id C6666340939CC; Mon, 18 Nov 2013 10:18:36 +0100 (CET)
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.2.3 (2007-08-08) on sun.utb.cz
X-Spam-Level: *****
X-Spam-Status: Yes, score=5.7 required=5.0 tests=HTML_MESSAGE,MIME_BASE64_TEXT,
    RAZOR2_CHECK,RCVD_IN_BL_SPAMCOP_NET,SPF_PASS,URIBL_JP_SURBL autolearn=no
    version=3.2.3
X-Spam-Report:
    * -0.0 SPF_PASS SPF: sender matches SPF record
    * 0.0 HTML_MESSAGE BODY: HTML included in message
    * 1.8 MIME_BASE64_TEXT RAW: Message text disguised using base64 encoding
    * 0.5 RAZOR2_CHECK Listed in Razor2 (http://razor.sf.net/)
    * 2.0 RCVD_IN_BL_SPAMCOP_NET RBL: Received via a relay in bl.spamcop.net
    * [Blocked - see http://www.spamcop.net/bl.shtml?123.58.178.204]
    * 1.5 URIBL_JP_SURBL Contains an URL listed in the JP SURBL blacklist
    * [URIs: icmeat-conf.com]

```

Fig. 94: Spam message partial email header. The sender's domain name, based in the People's Republic of China, was associated with spam campaigns directed at academic employees.

Source: Own work.

infrastructure for test definitions updated continuously, the omission is not critical. The score for classifying an email as spam can be set arbitrarily (here at 5.0), and each check adds or subtracts from the total value: adding pushes the message closer to the “spam” category, subtraction hints at it being a legitimate email (“ham” in parlance of the software developers). The email in Figure 94 was determined to be spam by four criteria, three of which relied on polling third-party databases. The fourth one increased the score for non-standard encoding scheme (base64) which is found in regular emails with negligible probability. The `autolearn=no` key-value parameter in the header signifies an identical pattern has been encountered previously and SpamAssassin did not add any new rules to the database based on the message. A similar type of spam was thus received on `sun.utb.cz` before.

Destination address suggests all email addresses are internally converted to the same suffix, `@mailbox.utb.cz`. Indeed, a loopback test email was successfully delivered and its header corroborated the assumption messages within the same domain do not pass through ESET Mail Security for Linux/BSD. Figure 95 depicts the full header. Note the omitted information about the checks to determine whether the email can be considered spam, listed and explained in Figure 94.

No alert would have been generated had the email contained malicious links or files. This marks the same-origin domain inherently trusted. The practice is a security threat which does not acknowledge insider threat as a viable attack vector. Handling all messages as potentially malicious would create a line of defense against an adversary who only has to produce a valid password to impersonate arbitrary user. As demonstrated in case study 1, relying on proactive password management from users would be erroneous and could result in account compromise.

The main purpose of this phase was to find out as much intelligence as possible about the IT processes at the target. By correlating direct observations with publicly available information

Furthermore, the tools generate substantial traffic and may flood the network so that no other requests can be processed, a form of denial-of-service attack discussed in chapter 2.4.4. Rate limiting will be obeyed when sweeping sizable IP ranges or a single host repeatedly. EC-Council (2010, pp. 2-5) states that “[i]f the systems are affected by the attack, it may lead to loss of information, system breakdown, loss or misconfiguration of the company’s systems, and so on. Any damage to the systems or to the information in them may cause losses to the organization. Costs can run high on electronic assets such as client databases, proprietary code, documentation, and intellectual property.” Losses incurred by hardware unable to process additional requests which results in an inaccessible website can be substantial, depending on organization’s size and its dependence on online services.

IT personnel provided the author with a list of more than 40 internal servers, IP addresses, and operating systems. It was therefore decided to include all in the first step, i.e., determining their status, but only a subset will be selected for further tests. The rationale is that once the adversary gets a foothold in the target system, continuing with the PT increases the probability her activities will be discovered, the breach fixed, and the system audited and possibly hardened to thwart future incursions. Moreover, mission-critical services should be assumed under close scrutiny with very low thresholds set: the administrators are prepared to accept more false positives in exchange for decreased false negatives and overall security. Rather than the frontal approach, an indirect path presents comparatively lower risk of exposure.

The first step is to determine whether hosts reply to incoming requests, i.e., if they are “live.” A live host is reachable over the network if it returns a reply message to a request generated using the ping utility. As all communications over the Internet rely on packets, a single data unit is sent to the target. If an answer is received, the host accepts incoming connection attempts at the particular time. Kali Linux has a tool called `fping` which automates the process and allows ping sweeps where a range of IP addresses from a text input is processed sequentially. The server list obtained from the IT personnel was converted to a plaintext file and transferred over to the VOS where it can be invoked in the `fping` command. The number of destinations is small and so several sweeps can be performed because a host may not respond due to various reasons, the most plausible being inactivity (powered-down state). It may be prudent to repeat the test at various times to discern patterns, e.g., some hosts operational during working hours but inaccessible otherwise, while others in the up state continuously due to high availability requirements, e.g., antivirus updates, software license servers, and CCTV circuits.

The `fping` utility cycles the host in a round-robin fashion. When the last line is reached, the process is iterated again from the beginning unless specified differently. Each IP address is given equal amount of resources; if the host produced a reply, it is removed from the list by default and the file is thus dynamically updated to only retain destinations from which answers has not been received so far. Figure 96 depicts the results along with the time the command took to execute.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# time fping -a -A -f /home/ip.txt > /home/ipres.txt
195.178.93.104 address not found

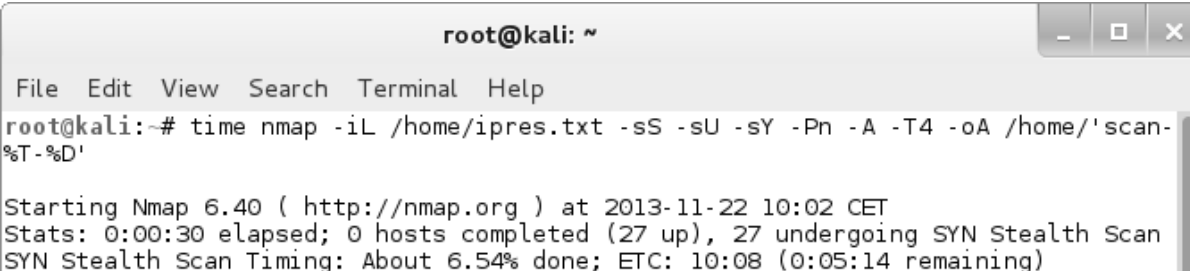
real    0m5.406s
user    0m0.008s
sys     0m0.140s
```

Fig. 96: Live host scan results. A single IP address, 195.178.93.104, was not found, the rest was categorized as either active or inactive based on whether a reply was received. Source: Own work.

The parameter specifying packet size was not changed, and each probe therefore defaulted to 56b for a traffic of around 2.5kB. However, should the value be increased to the maximum of 64kB, the utility would have generated about 3MB of data in a single pass. Cycling `fping` indefinitely, approximately 2GB of traffic would be generated per hour, causing noticeable uptake and possibly a DoS in smaller ICT infrastructure. A total of 27 stations were deemed active.

The second step is to port scan the stations using the Nmap utility. Nmap (Network Mapper) is a versatile tool which allows to specify options covering functionality of `fping` (host discovery), port scanner, service fingerprinting, and firewall/IDS detection and evasion. As with any tool in Kali Linux, it is available free of charge with unrestricted functionality. Ports are "... logical access points for communication over a network," (Mallery, 2009, p. 19) port scanners are "... utilities [sending] out successive, sequential connection requests to a target system's ports to see which one responds or is open to the request. Some port scanners allow... to slow the rate of port scanning – sending connection requests over a longer period of time – so the intrusion attempt is less likely to be noticed" (West, 2009, p. 42). While the delimitation is mostly accurate, Nmap does not probe the ports sequentially, but rather employs randomization and rate limiting to stay below the detection threshold. It can craft custom packets which do not conform to the standards and aim to elicit server-side response. Even though such packets are rare because networking hardware mostly obey the specifications, designers had to handle the edge cases according to practices which were left free to vary.

There are 65 536 ports numbered sequentially from 0 to 65 535, with 0 being a reserved port not allowed to be used in any OS. Both core networking protocols, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) each have their own port sets, totaling $2 \times (65\,536 - 2) = 131\,070$ available ports. Of these, 0–1023 are "well-known," assigned to system-level processes. Apart from those, hosts run software which accept inbound and outbound connections on various ports, necessitating them to be bidirectionally open. The adversary can deliver a payload by specifying any port number, but perimeter-level security defenses may intercept the incoming data and discard them outright (closed), or pass them into the internal network provided they don't violate the rules (open). Crafting packets so that the firewall/IDS protection routines are not triggered is challenging and will not be attempted. Instead, preference will be given to assess the results and hypothesize about the attack vectors they point at. Figure 97 depicts a port scan in progress.

A screenshot of a terminal window titled "root@kali: ~". The terminal shows the command `time nmap -iL /home/ipres.txt -sS -sU -sY -Pn -A -T4 -oA /home/'scan-%T-%D'` being executed. The output indicates that Nmap 6.40 is starting at 2013-11-22 10:02 CET. The current status is: "Stats: 0:00:30 elapsed; 0 hosts completed (27 up), 27 undergoing SYN Stealth Scan". The scan timing is "About 6.54% done; ETC: 10:08 (0:05:14 remaining)".

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# time nmap -iL /home/ipres.txt -sS -sU -sY -Pn -A -T4 -oA /home/'scan-%T-%D'
Starting Nmap 6.40 ( http://nmap.org ) at 2013-11-22 10:02 CET
Stats: 0:00:30 elapsed; 0 hosts completed (27 up), 27 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 6.54% done; ETC: 10:08 (0:05:14 remaining)
```

Fig. 97: Nmap scan progress. List of live hosts from the previous step was used as input, and Nmap instructed to bypass ping discovery which implicitly treats all hosts as reachable.

Source: Own work.

Nmap was instructed to treat hosts as live, forgoing ping probe discovery which was previously performed using `fping`. An alternative approach would be utilize only Nmap and simplify the toolchain. The behavior offers benefits: some hosts may be configured to drop ping requests with the sender assuming the host is powered off. Nmap scans all ports regardless of whether they are receptive to the probing which may result in discovering a destination with open ports the

operator did not harden as they expected resources would not be spent on apparently uninteresting hosts.

Only well-known ports (1–1 023) will be tested. An alternative would be to scan all ports: while more resource- and time-intensive, it is assured to pick services positioned to non-standard interfaces purposefully by system administrators. It is assumed only a small subset will warrant further investigation, the rest closed to reduce the attack surface and prevent association queries from being responded to. Another option would be to scan ports which host services such as database systems under the assumption default settings were not altered. For example, MySQL listens on port 3306 outside the well-known port range.

Finally, OS detection using internal database of signatures was turned off. Even when a perfect match is not made, Nmap has fuzzy algorithms which construct confidence intervals based on packets received from the target. Version detection has legitimate uses: administrators can query remote machines, determine possible avenues of attack, and audit devices on their network. The adversary can use OS fingerprinting to craft exploits for a specific OS version. In chapter 5.2.2, software versions were identified from logs and virtual directory listings: while the methodology is sound, some distribution maintainers may back-port security patches without changing the version numbers and even older OSs may thus have the latest additions incorporated. Nmap offers fine-grained OS detection capabilities to narrow down a pool of possible candidates. The output's header is depicted in Figure 98.

```
root@kali: ~
File Edit View Search Terminal Help
Host is up (0.0067s latency).
Scanned at 2013-11-22 10:02:32 CET for 100644s
Not shown: 1052 filtered ports
PORT      STATE      SERVICE      VERSION
3389/tcp  open      ms-wbt-server  Microsoft Terminal Service
2/udp    open|filtered compressnet
3/udp    open|filtered compressnet
7/udp    open|filtered echo
9/udp    open|filtered discard
13/udp   open|filtered daytime
17/udp   open|filtered qotd
19/udp   open|filtered chargen
20/udp   open|filtered ftp-data
21/udp   open|filtered ftp
22/udp   open|filtered ssh
23/udp   open|filtered telnet
37/udp   open|filtered time
38/udp   open|filtered rap
42/udp   open|filtered nameserver
49/udp   open|filtered tacacs
53/udp   open|filtered domain
67/udp   open|filtered dhcpc
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
80/udp   open|filtered http
88/udp   open|filtered kerberos-sec
111/udp  open|filtered rpcbind
112/udp  open|filtered mcidas
113/udp  open|filtered auth
120/udp  open|filtered cfdpckt
123/udp  open|filtered ntp
135/udp  open|filtered msrpc
```

Fig. 98: Nmap scan output. The scan of the 27 machines took 27 hours 57 minutes 59 sec, 172 197 raw packets (6.067MB) were sent and 30 174 (1.181MB) were received. Source: Own work.

For the rest of the case study, two targets were picked against which tests would be performed. It was decided two IP hosts would be representative of how scanning for vulnerabilities may lead to system compromise. The results for both hosts uncovered many UDP ports for which

no response was received, meaning the port is either firewalled or configured not to generate responses.

Details pertaining to how individual services can be exploited are beyond the scope of the thesis but Nmap managed to identify non-system services which could give the adversary leverage to the system. An example is eDonkey, a peer-to-peer (P2P) file-sharing client which comprises of a port list the software tries to attach itself to sequentially when the default port 4242 is unreachable. Another P2P network, Gnutella, was identified as well. Because the same services were found on all hosts, it is likely system administrators replicated and deployed OS instances from a centralized image file. Within the port range, a single UDP port was determined open, NetBIOS-NS, a service with known weaknesses (Zdrnja, 2012). Moreover, a vulnerability in the Microsoft Windows OS allowed packet broadcast and reception despite the built-in firewall explicitly instructed to drop all traffic on the port; indeed, the open state indicates no barrier exists between the internal network and the hosts.

Nmap provided a list of TCP interfaces and one UDP port listening for incoming connections. The host IPs are on the internal network and thus presumed unreachable to anyone from outside the faculty. The next step in the VA is to identify vulnerable services: stemming from substandard configuration but primarily patch deployment delays, they may allow to deliver payloads causing the system to enter a state in which executable code can be launched. The repercussions range from establishing a CLI session to planting malware and maintaining access so that the station can be used for compromising more network resources. Exfiltrating password files for offline reverse engineering, covertly adding a system account with superuser privileges, and expropriating sensitive data are also common aftereffects of a breach.

The case study will impose a restriction stating only non-malicious programs, i.e., CLI will be launched to demonstrate proof of concept. To that end, a vulnerability scanner called Nessus was selected to conduct the scan for possible vectors of approach. Not being included in Kali Linux by default, version 5.2.4 for Debian 6.0 32b was downloaded and set up. Nessus is accessed locally through a web browser. The GUI is depicted in Figure 99.

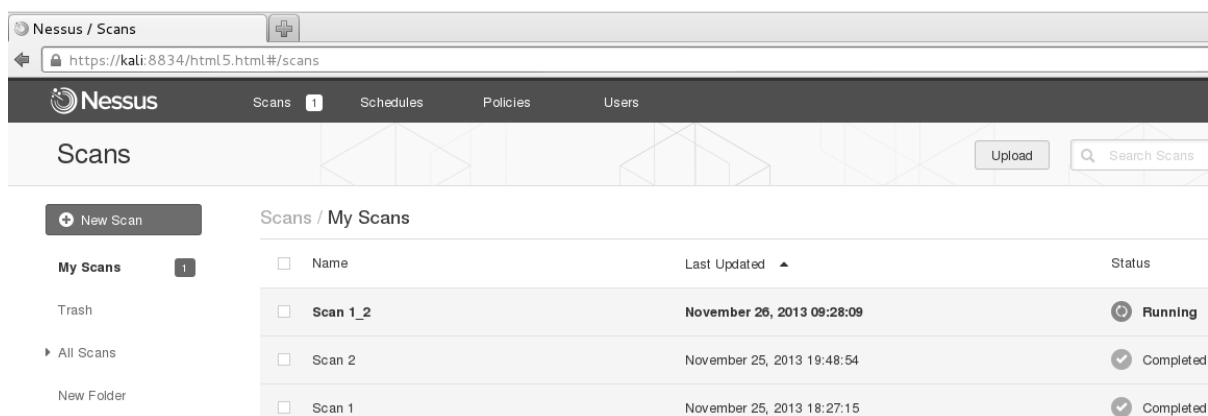


Fig. 99: Nessus GUI. The interface shows three tests: two completed successfully and one currently active. Each requires a policy and one or multiple targets to be specified.

Source: Own work.

Before VA can be launched, a policy with test settings needs to be created. Nessus offers multiple scenarios represented by templates the user can select and tailor, and an advanced option where specifications can be set in greater detail. For the case study, two templates will be used: Basic Network Scan and Advanced Policy. Basic Network Scan needs the following information: policy name and scan type (internal/external). The first is a descriptive string which the user

selects when setting up VA, the second specifies whether Nessus should scan the whole port range and load plugins for web-based vulnerabilities, or probe well-known ports and focus on presence of embedded devices such as printers. Both varieties were tried for completeness but no advanced options were otherwise changed. A total of four policies were created: internal and external for each IP address.

In the Advanced Policy, two settings were changed: “Safe Checks” and “Stop Host Scan on Disconnect.” The latter was turned on to stop VA in case the target goes offline, an indication the scan may have caused system instability. Safe Checks classify plugins into two categories: those minimizing impact on the host, and those with the potential to cause network and host availability disruptions. Performing a scan without Safe Checks may result in more detailed results if the analyst is willing to accept the risk of damage to the target. Both options were tested with permission of the IT personnel who had further been informed prior to test commencement in case a reboot or other intervention would be needed. A total of 8 tests were planned:

- 2 external basic network tests,
- 2 internal basic network tests,
- 2 advanced scans with Safe Checks on,
- 2 advanced scans with Safe Checks off.

The scenarios describe what intelligence can a determined low-skilled insider agent obtain from an internal network while not possessing any valid login credentials. Results from the external network scans for both IPs are depicted in Figure 100.

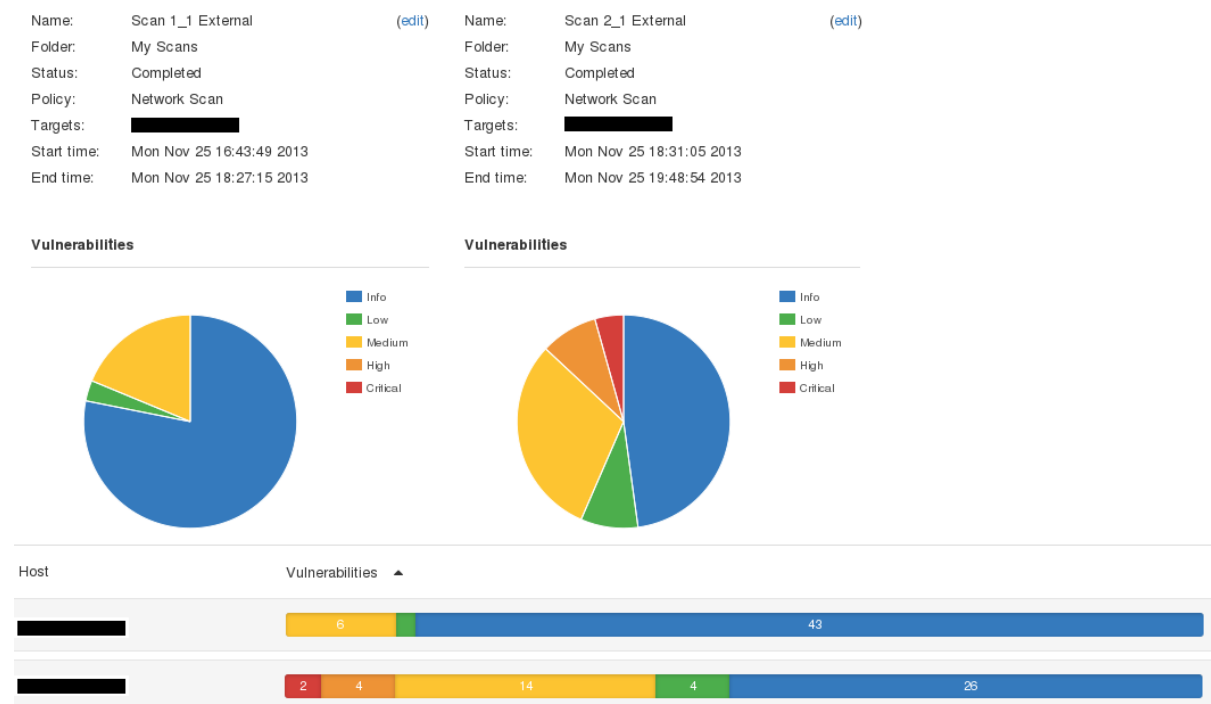


Fig. 100: Nessus basic network scans results. The image is a composite of several graphical outputs the tool offers along with a drill down of the vulnerabilities found.

Source: Own work.

The findings are marked according to their severity level: Informational, Low Risk, Medium Risk, High Risk, and Critical Risk. The levels are calculated according to Common Vulnerability Scoring System (CVSS). Informational vulnerabilities provide miscellaneous data about the target which do not constitute security risks but system misconfigurations and violations of best

practices resulting in information disclosure. To demonstrate: Nessus attempted connection to the host using Remote Desktop Protocol (RDP) and if successful, took a capture of the login screen. Despite not being a vulnerability, the screen with established user sessions is depicted in Figure 101.

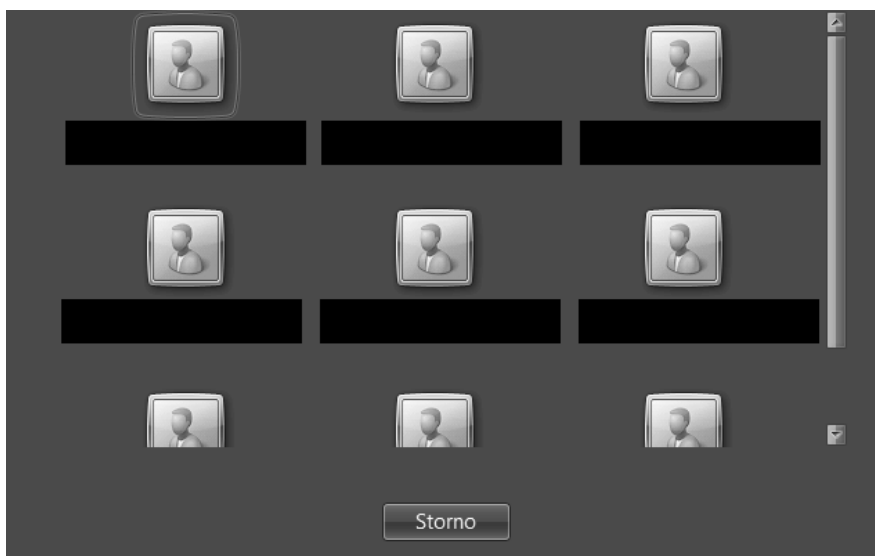


Fig. 101: Nessus RDP login screen screen shot. Established connections are listed by name which may leak information about presence of a particular user on the host.
Source: Own work.

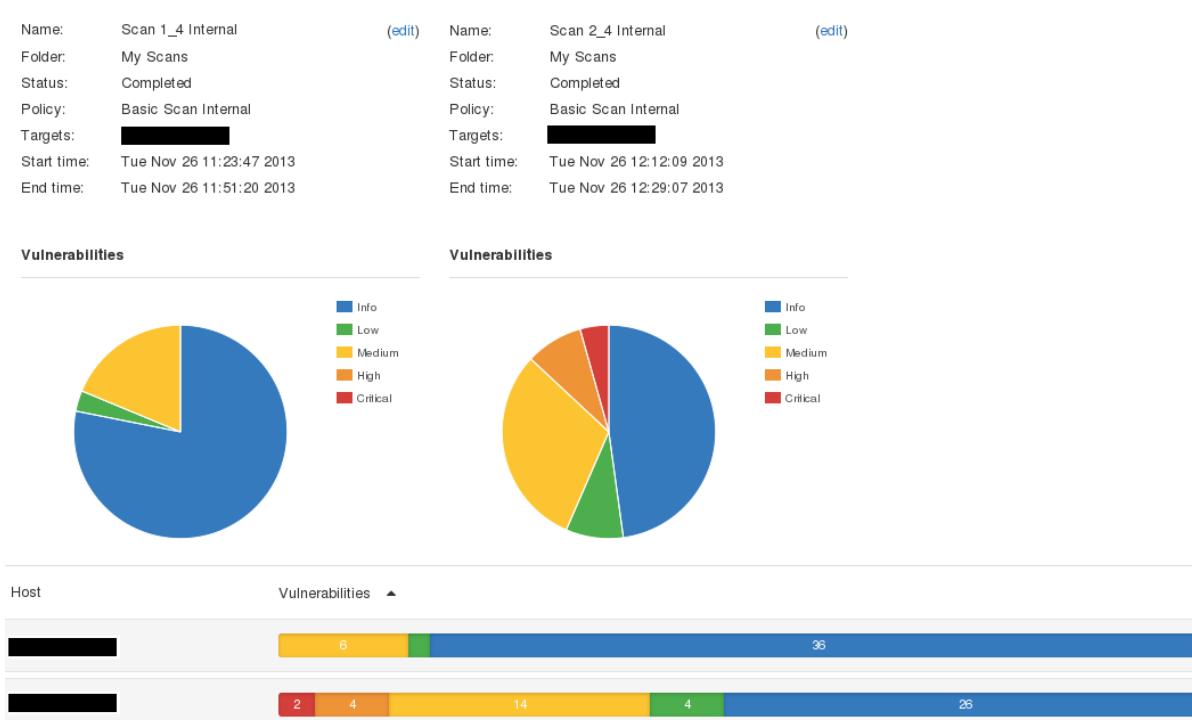


Fig. 102: Nessus internal basic network scans results. External and internal scans resulted in identical findings.
Source: Own work.

Figure 102 depicts results for basic scan results. A Medium Risk vulnerability for host 1 was identified which allows the adversary to inject herself into the path between the client and server,

either passively intercepting or actively modifying data in transit. Results for host 2 revealed a total of 24 vulnerabilities rated Low Risk or higher, with 14 Medium, 4 High, and 2 Critical Risk. Entries in the last category were related to outdated versions of Apache HTTP Server and PHP. The PHP version is no longer supported, a critical vulnerability in itself as newer security patches are no longer released. A medium-risk vulnerability uncovered a PHP configuration file, depicted in Figure 103, which lists server settings and active modules. The adversary can effectively tailor exploits based on known combinations of insecure plugins to maximize probability of system breach. Additionally, the host serves as a portal for authenticated students and employees to share information, but login credentials are sent to the server unencrypted and can be intercepted and used to hijack victim's account.

Nessus was also able to fingerprint the OS (AIX 5.3), discontinued in 2011 and removed from the IBM support pipeline in 2012 which opens it to vulnerabilities patched in later versions. Moreover, phpMyAdmin interface (3.5.4), and PHP (5.2.14) were identified; the former has been phased out and is no longer updated as of 2014, the PHP release has been obsoleted by version 5.3 in 2009 and is unsupported from 2011. At least 5 vulnerabilities could be exploited to execute DoS attack, run unsanctioned code, and others. Results for advanced scans with Safe Checks on are depicted in Figure 104 for both IP addresses.

System	Linux [redacted] 2.6.18-238.9.1.el5 #1 SMP Tue Apr 12 18:10:56 EDT 2011 i686
Build Date	Aug 27 2010 16:42:17
Configure Command	./configure '--host=i686-redhat-linux-gnu' '--build=i686-redhat-linux-gnu' '--target=i386-redhat-linux' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=./config.cache' '--with-libdir=lib' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-curl' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-syssem' '--enable-sysshm' '--enable-sysmsg' '--enable-wddx' '--with-kerberos' '--enable-ucd-snmp-hack' '--enable-shmop' '--enable-calendar' '--without-mime-magic' '--without-sqlite' '--with-libxml-dir=/usr' '--enable-xml' '--with-system-tzdata' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter' '--disable-json' '--without-pspell'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled

Fig. 103: Web server phpinfo disclosure. The file lists PHP configuration options, environment variables, and installed plugins.
Source: Own work.

Safe Checks on failed to discover 1 Critical, 4 High Risk, and 4 Medium Risk vulnerabilities, presumably due to unsafe modules not loaded for the test. The omitted critical entry pertains specifically to the outdated PHP. Overall, it was prudent to combine the information with the basic scan for a comprehensive coverage. Outputs from the advanced scan with Safe Checks off, which did not generate any stability or performance issues, are demonstrated in Figure 105. However, system logs will contain traces which could probe further investigation had the attempt been unsanctioned.

The scan uncovered a new High Risk vulnerability for host 1 in RDP on the Microsoft Windows OS which may enable remote code execution. On the other hand, PHP vulnerabilities were missing for host 2 in the Informational and Medium Risk categories. If the adversary relied

Name:	Scan 1_2 Safe Checks On	(edit)	Name:	Scan 2_2 Safe Checks On	(edit)
Folder:	My Scans		Folder:	My Scans	
Status:	Completed		Status:	Completed	
Policy:	Safe Checks On		Policy:	Safe Checks On	
Targets:	██████████		Targets:	██████████	
Start time:	Tue Nov 26 09:28:10 2013		Start time:	Tue Nov 26 10:48:16 2013	
End time:	Tue Nov 26 09:52:49 2013		End time:	Tue Nov 26 10:58:48 2013	

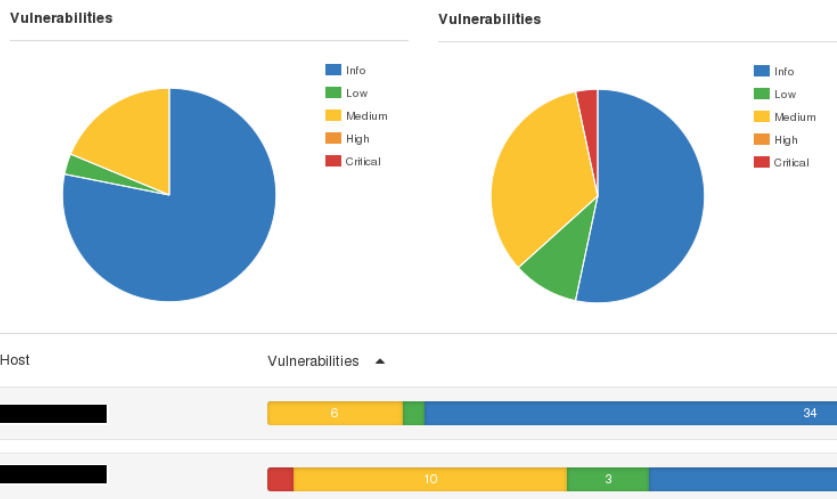


Fig. 104: Nessus advanced scans Safe Checks on results. Plugins deemed unsafe were disabled to prevent host instability.
Source: Own work.

Name:	Scan 1_3 Safe Checks Off	(edit)	Name:	Scan 2_3 Safe Checks Off	(edit)
Folder:	My Scans		Folder:	My Scans	
Status:	Completed		Status:	Completed	
Policy:	Safe Checks Off		Policy:	Safe Checks Off	
Targets:	██████████		Targets:	██████████	
Start time:	Tue Nov 26 13:01:46 2013		Start time:	Tue Nov 26 14:41:56 2013	
End time:	Tue Nov 26 13:28:06 2013		End time:	Tue Nov 26 14:54:09 2013	

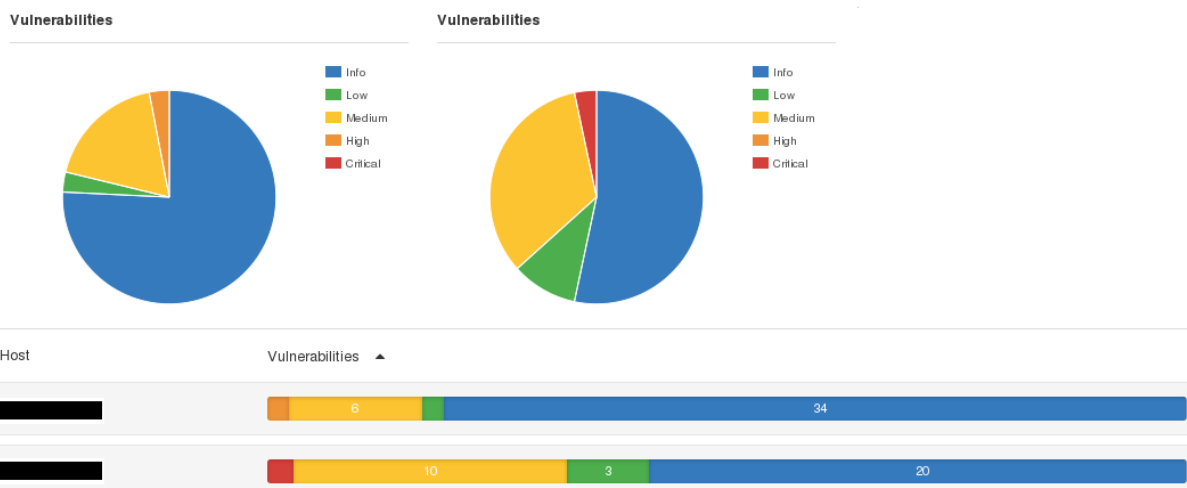


Fig. 105: Nessus advanced scans Safe Checks off results. An additional High Risk vulnerability was discovered for host 1.
Source: Own work.

solely on this scan for actionable intelligence, PHP would have been omitted even though it presents a viable avenue of approach. However, high false positive rates must be mentioned: for credentialed and superuser-level scans, Nessus achieves the best accuracy, but for unprivileged

testing, particularly with Safe Checks enabled, a degree of uncertainty is introduced. Aggressive techniques grouped under the Safe Checks off switch may give more accurate results but also increase the risk of detection and countermeasures deployed as a response. Collating reports from multiple tests may help filter out erroneous entries. Even though the insider should be detected when using noisy VA procedures, the target must be eventually interacted with either directly or through a proxy regardless of the technique employed. Firewall or IDS may be configured permissively on internal networks, but should nevertheless detect and pass a warning to IT personnel.

As a last step, we will attempt to penetrate the target by launching an exploit in Metasploit. The framework was mentioned in passing during metadata extraction to be a comprehensive open-source PT tool which provides a toolchain for tailoring and deploying malicious code. Creating an exploit is a process which consists of several steps, the result of which is a code bundle sent to the host. After processing the request, a malicious routine is executed and the payload delivered. In the case study, the CLI session will be used as a proof of concept. We will not attempt to exhaustively cover all of Metasploit's functions, but rather show how freely-available tools can be used for both legitimate (the framework is favored in penetration testing for its comprehensive, up-to-date collection of integrated exploits) and deleterious purposes. Metasploit runs in a terminal window and accepts textual input from the user. The reports from Nessus need to be correlated with available modules to see if a match exists. The framework also allows for port and host scanning which gives the insider multiple avenues to reach her goal: either utilize the framework exclusively, combine three specialized tools (Nmap, Nessus, Metasploit), or any combination thereof. However, the exploitation capabilities are unique to Metasploit.

A single exploitable vector on host 1 comes from Safe Checks off scan; titled "Vulnerability in Remote Desktop Could Allow Remote Code Execution," it specifies a situation where an unauthenticated remote attacker can succeed in launching arbitrary code on the target machine. Moreover, the flaw could result in a host crash under specific circumstances. Metasploit returned no results for the particular vulnerability, but knowing the target likely operates RDP and thus Microsoft Windows OS, another candidate was found that leverages a disclosed weakness in RDP. The steps are demonstrated in Figure 106.

The exploit supports a CLI to be executed on the host because the objective is to minimize detection probability and remain stealthy. No data will be saved to HDD so unless memory is actively scanned for anomalies, the technique should not trigger alert. Reverse TCP connection instructs the target to initiate the connection rather than passively accept it. The options set before the payload can be sent are remote (RHOST) and local host (LHOST) represented by IP addresses. When supplied, the `exploit` command generates the exploit and attempts to deliver it. Nevertheless, even though all conditions for the exploit's successful execution have been met, the result cannot be predicted in advance. The vulnerability identified in Nessus could be a false positive; firewall may filter incoming requests on the port (3389 for RDP) and drop it on arrival; outward connections may not be permitted; IDS may detect the breach attempt and deploy countermeasures, etc. Issuing the command, the package was sent. The output is shown in Figure 107.

The framework reported RDP is unreachable, disabled, turned off at the moment, or the port is firewalled. In the first two cases, repeating the process at a later time may result in success if the service is enabled, the last case denotes a situation where firewall rules are set to prevent unauthenticated connections from passing through any data. Microsoft's knowledge base states no Microsoft Windows OS has RDP accessible by default, a security precaution to thwart attempts to exploit the attack vector. The protocol therefore does not seem to constitute

```

root@kali: ~
File Edit View Search Terminal Help
msf auxiliary(ms12_020_maxchannelids) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf auxiliary(ms12_020_maxchannelids) > show options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     3389             yes       The target address
  RPORT     3389             yes       The target port

msf auxiliary(ms12_020_maxchannelids) > set RHOST ██████████
RHOST => ██████████
msf auxiliary(ms12_020_maxchannelids) > show options

Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     ██████████       yes       The target address
  RPORT     3389             yes       The target port

msf auxiliary(ms12_020_maxchannelids) > exploit

```

Fig. 106: Metasploit payload setup for host 1. The framework can deliver various payloads, a reverse TCP command-line interface was selected as a proof of concept. Target IP address was redacted. Source: Own work.

```

root@kali: ~
File Edit View Search Terminal Help
msf auxiliary(ms12_020_maxchannelids) > exploit

[*] ██████████ - Sending MS12-020 Microsoft Remote Desktop Use-After-Free DoS
[*] ██████████ - 210 bytes sent
[*] ██████████ - Checking RDP status...
[-] ██████████ - RDP Service Unreachable
[*] Auxiliary module execution completed

```

Fig. 107: Metasploit payload sent to host 1. A total of 210 bytes were sent to the target which contained the exploit code along with the payload. Source: Own work.

a viable approach strategy, and alternative means of running the payload is necessary. Inspecting the Nmap report, several open TCP ports were found and associated services fingerprinted. The Metasploit database was polled and after executing 12 separate exploits against the target, no CLI session was established and further attempts were abandoned. The same situation occurred for host 2 where Apache HTTP Server, PHP, and phpMyAdmin were targeted. The following test restrictions prevent to classify the hosts as secure:

- low-skilled adversary which is restricted from using custom-tailored exploits commonplace in practice,
- Metasploit framework does not contain zero-day vulnerabilities,
- stealthy approach was preferred over noisy techniques to simulate insider threat,
- remote code execution was the sole criterion for choosing exploits,
- social engineering was not used,
- network traffic to mount DoS was not generated.

Social engineering in particular may prove viable and is discussed below. Launching DoS

from internal network would be immediately detected and a digital trail established. Another disadvantage of the case study scenario is reliance on known exploits whose definitions were incorporated into Metasploit after a grace period where affected vendors can release security warnings and issue patches. For well-known software, the window between full disclosure and Metasploit database inclusion can span years, giving IT personnel time to harden the software. Some weaknesses are listed as having an excellent rank and high success probability after more than 10 years, though, which hints at excessively long or nonexistent patch deployment policies for some programs. Exploits were also limited to remote code execution; DoS, application crashes, and other classes of attacks were not pursued.

Reiterating what was stated previously, the outcome does not validate the systems as “secure” because security is an abstract concept necessitating a quantifiable metric. Instead, it proves the hosts withstood a few attempts from a low-skilled insider using publicly-available tools under restrictive conditions. The case study avoided resorting to attacks capable to saturate ICT resources or cause service disruptions, instead preferring stealthier techniques. Regardless, system monitoring entries were generated on multiple occasions, possibly on every test which generated network traffic.

Another vulnerability mentioned in the previous phase pertains to email addresses. As a countermeasure, it is possible to modify emails in a way they remain usable, e.g., user does not have to type them manually, which breaks extraction algorithms in automated scripts. One alternative is to save addresses as graphics files, optionally text-morphed. This forces the attacker to use Optical Character Recognition (OCR) with reduced efficiency compared to simple text parsing, but it also introduces discomfort because the email address cannot be copied. This could cause problems for ambiguous characters (1/l, O/0). Another option is to use scripting techniques to mingle the text on the screen while still allowing to copy the original string into the clipboard on mouse click. If implemented correctly, the technique significantly decreases the probability the string will be recovered due to inability of the harvesting bots to parse the script. Replacing “@” with [at] and “.” with dot, reversing the address from right to left, and character rotation do not thwart automated harvesting. Moreover, the user needs to replace or reverse the text as well which decreases comfort when dealing with obfuscated links. No definitive solution has been accepted to eliminate email collection and spamming.

Scrubbing metadata from Microsoft Office documents was made easier in post-2007 releases where facilities are provided to remove personal information entirely. Nevertheless, the adversary can narrow down the office suite version based on file extensions: if .docx, .xlsx, or .pptx are discovered, at least Microsoft Office 2007 is installed on the machines. Extrapolating further, the perpetrator may predict most users will have identical set of programs due to the university licensing policy and expected end-of-life support for Microsoft Windows XP SP3 and Microsoft Office 2003 in 2014. Metadata removal can be enforced via safeguards which scan for presence of identifying information, and the user is notified the document will not be released until they are scrubbed. Accompanied by a concise manual, the process could be effective in mitigating metadata leaks.

PDF metadata are also of concern. During website analysis, a PDF was found whose metadata included both author (local part of the email address) and a website, convert-jpg-to-pdf.net. Imagine the following spear phishing scenario: the user receives an email with an attachment purportedly from the site they frequently visit to generate PDF files on. The message body will state an error was found in the last document the user generated – because the original PDF contained a timestamp in the metadata, the adversary can simply transfer it over to the text to establish credibility. The email will have a single attachment, a PDF file allegedly containing the corrected output with approximately the same size as the file on the university website.

When opened, a payload is executed and a vulnerability exploited allowing the perpetrator to run arbitrary code on the target machine with the current user's privileges. A recent example is CVE-2013-3357⁴ which affects all Acrobat Reader versions prior to 10.1.8 and 11.x before 11.0.04 on Microsoft Windows and Mac OS X. As regular patch management is not practiced by many users, the vulnerability may be widespread because Acrobat Reader is the preferred PDF viewer at the faculty. The scenario exploits existing information unwittingly provided by the document creator to launch a directed attack which would likely not be classified as spam if sent from an attacker-controlled email address created to impersonate the legitimate domain name.

Vulnerabilities related to default user password assignment should be considered serious and the policy changed immediately to disallow local or remote account takeovers. While usernames are hard coded, passwords are changeable via unencrypted web service. Lack of HTTPS leaves users vulnerable in case they decide to change their password while on a wireless network. Local-area network is also susceptible to passive data interception which cannot be reliably detected in absence of digital certificate validation. As for the functionality, the site does not offer any strength meter to assess resilience of the string. A zero-order entropy, the most common measure of password strength, will be described later in the thesis along with other options which take into account not only the alphabet size but propensity to dictionary attacks, repeating patterns, and other shortcomings which reduce the work factor during reverse engineering. Password change should encompass both domain and email services.

Case study 1 dealt with consequences of improper password selection. However, if 50 % of employees and students left their authentication credentials unchanged, as many as 1 623 (as of March 1, 2013) accounts could be exploited via online enumeration because it was previously ascertained no CAPTCHA is presented when the Roundcube email front-end is accessed from outside the university IP range. To recount the password assignment policy: each employee and student is commissioned a string of the form YYMMDDSSSC. Table 18 lists the restrictions which allow search space pruning.

Tab. 18: *Birth numbers search space enumeration. Omitting the slash from the birth number did not change the password strength due to its fixed position.*
Source: Own work.

Part	Range	Search space
YY	00–99	99
MM	01–12 (men); 51–62 (women)	12
DD	01–31	31
SSS	000–999	999
C	0–9	0

Multiplying the first four search spaces naïvely, the number of combinations totals 36 791 172, viable for offline brute-forcing even on commercial hardware because the password consists solely of numerical characters. GPU-based cracking benefits from parallelism and would take much lower time to exhaustively map the whole candidate pool. Assuming the hash is not available, the adversary needs to resort either to online scenario where passwords are sent to the server successively or in batch through web interface, gain access to Novell login screen via Virtual Private Network (VPN) or physically on the premises. Regardless of the way, a list of

⁴<http://www.cvedetails.com/cve/CVE-2013-3357/>

strings need to be assembled. Focusing on a particular user, the adversary can prune the degrees of freedom in the following way:

- YY can be limited to 10 years using publicly-available sources (photographs) and inspecting CV (10),
- MM and DD cannot be trivially estimated (12+31),
- SSS cannot be trivially estimated (999),
- C is dependent on YMMDDSSS (0).

A brief note about the check digit: if the sequence is divisible by 11, $C=0$, otherwise it is added so that $YMMDDSSSC \bmod 11 = 0$. Hence, the figure is not free to vary and scripting the attack, the perpetrator can programmatically check if the divisibility condition holds and modify C accordingly. The search space size is 429 570; should the candidates be randomized instead of sequenced, the correct password is obtained after 50 % of guesses on average, i.e., 214 785. Assuming two login attempts can be generated per second, equivalent to where a legitimate user tries to simultaneously authenticate against VPN and access email, the account is successfully breached after at most 30 hours. With default passwords left unchanged for months (chapter 4.2.1), the adversary may even slow down the probing to stay under the threshold.

The password assignment policy is clearly ineffective for deterring coordinated efforts at obtaining user-level privileges, and IT personnel should strongly consider a change. A recommended way to generate authentication tokens is to decouple them from personal information. Strong password generators sourcing entropy from OS and user inputs are available; alternatively, one-time sequences can be distributed which are invalidated after first use and force account holders to procure their own passwords. Strength meters should calculate at worst zero-order entropy and at best blacklist known weak passwords. They could be based on word lists from database breaches as presented in case study 1 as well as contain common repeated patterns because "... simplistic strength estimation gives bad advice. Without checking for common patterns, the practice of encouraging numbers and symbols means encouraging passwords that might only be slightly harder for a computer to crack, and yet frustratingly harder for a human to remember" (Wheeler, 2012). The added server-side processing should not be an issue: changing a password is a discrete event not practiced by multiple users concurrently. Keeping the source dictionaries updated ensures users are objectively informed of the string's strength when creating or changing it. Graceful password revocation policy should be also put in place, although preventing account holders from entering identical strings with appended/prepended numbers and other symbols would increase storage requirements to retain historical records for comparison purposes. Mobile devices must have their own set of policies enforced by software means (profiles). BYOD management should not be underestimated due to increases in accessing sensitive resources over the Internet using smartphones and tablets with capabilities surpassing those of desktop PCs and notebooks.

Attacks against BYOD devices do not differ substantially from those for notebooks because the technology shares many similarities, as apparent from a brief overview in chapter 2.3. In line with model in chapter 2.4.1 and 5, large-scale sophisticated campaigns cannot be expected to take place. Active network interception titled man-in-the-middle attack has been presented earlier (see chapter 2.4.4) and is on the forefront of what can be realistically expected from the insider or a low-skilled adversary. Mobile devices may store personal data and login credentials which could be sent automatically to authenticate against a server any time Internet connectivity is available. For example: the data for email accounts may be cached so that the procedure takes place in the background and the user is permitted access instantaneously. While convenient, passive network interception may trivially copy the data stream for later analysis.

Physical device acquisition is a noisy technique which alerts the victim after a short window of opportunity expires. The damage assessment may involve revoking passwords to preempt unauthorized use. Therefore, the attack vector will not be considered in the case study as acquiring the device solely to obtain login credentials is a contrived means to penetrate perimeter defenses. Social engineering and spear phishing may result in comparatively higher success rates while being stealthier under realistic assumptions: victim who does not inspect email headers, and implicit trust in same-origin messages.

Another vulnerability which should be addressed are superuser privileges each user possesses on their local workstations. This makes every password breach critical, giving an unauthorized party complete and unfettered access to the resources of the legitimate user. Even administrators are discouraged from operating in the superuser mode and should only elevate their status when performing system-wide changes. The principle of least privilege discussed in chapter 2.2.1 must apply to every entity if security is a priority. Microsoft Windows on user machines is set to superuser on account creation which coupled with lackluster security practices significantly expand the attack surface. All hosts on the internal network are equally likely to be targeted using either technological means or social engineering campaigns. The latter promise high return on time investment: direct observation noted emails received from same-domain addresses illicit seemingly automatic response which sees the victim open the email and any attachment it may include uncritically, bypassing any psychological safeguards put in place, e.g., inspecting the text, verifying the email as genuine using a side channel (telephone inquiry), and scanning the attachment before opening. The sole criterion is the local part of the address: if it belongs to the “privileged” subset of users, the email is deemed authentic and its content trusted implicitly. Plausible social engineering scenarios can be constructed from publicly-available information, and organizational structure exploited to extract emotional responses to stimuli such as curiosity, fear, interest, and uncertainty. A sequence of steps is presented below in which the adversary can manipulate an employee into opening a PDF file with a payload which allows to execute arbitrary unsanctioned code with the victim’s privileges:

1. The attacker forges an email which purportedly comes from an entity positioned higher than the victim in the organizational hierarchy; selecting the fake sender to be dean of the faculty, the probability of emotional response could be increased while verification as to the email’s purported origin would likely be omitted entirely.
2. A PDF whose body corresponds to the email text is attached. When opened, malicious code is executed in the background which opens a remote terminal session to the target machine. In order to forgo alerting the victim, the document should still contain text pertaining to the email body crafted in a way which does not make the recipient doubt its validity.
3. The email should be self-contained and require no response to the fake sender as it would reveal the social engineering scenario and may result in heightened suspicion in the future. Real-world effect is challenging to estimate accurately, though, because employees may simply ignore the warning a malicious attempt has been previously detected.
4. Stealthy access to the local workstation with administrative privileges is granted. The adversary establishes system persistence and begins scanning the infrastructure, e.g., using the Nmap utility or Metasploit.

Details pertaining to how the exploit can force the OS to initiate a terminal session is outside the scope of the thesis. A loopback email is depicted in Figure 108. The author redirected the

message to his own mailbox using a weakness in the intra-domain delivery system discussed below.

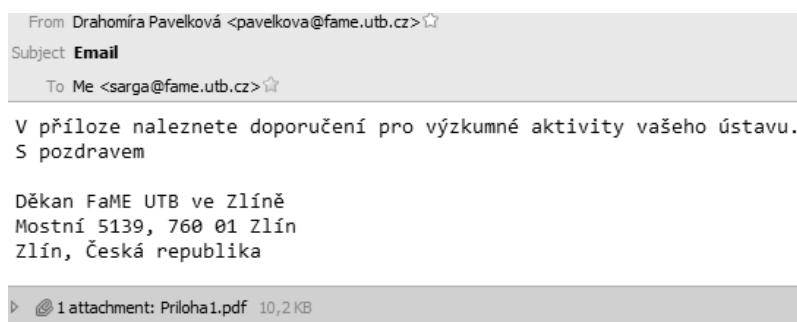


Fig. 108: *Loopback social engineering email.* The text in Czech translates to: “In the attachment you will find recommendations for future scientific endeavors of your department.” The PDF file contains a single blank page with a “test” string.

Source: Own work.

Note the local part of the email which corresponds to the one used by the then-current dean of the FaME, and bears no overt marks the sender’s electronic identity was spoofed. Exploiting lack of antispam controls for messages passed over the internal network, the email was delivered verbatim. The source is demonstrated in Figure 109.

```
Message-ID: <528DE6C5.3080705@fame.utb.cz>  
Date: [REDACTED]  
From: =?UTF-8?B?RHJhaG9tw61yYSBQYXZlbGtvds0h?= <pavelkova@fame.utb.cz>  
User-Agent: [REDACTED]  
MIME-Version: 1.0  
To: sarga@fame.utb.cz  
Subject: Email  
Content-Type: multipart/mixed;  
  boundary="-----090109030807060208090700"  
  
This is a multi-part message in MIME format.  
-----090109030807060208090700  
Content-Type: text/plain; charset=UTF-8; format=flowed  
Content-Transfer-Encoding: 8bit  
  
V příloze naleznete doporučení pro výzkumné aktivity vašeho ústavu.  
S pozdravem  
  
Děkan FaME UTB ve Zlíně  
Mostní 5139, 760 01 Zlín  
Zlín, Česká republika  
  
-----090109030807060208090700  
Content-Type: application/pdf;  
  name="Priloha1.pdf"  
Content-Transfer-Encoding: base64  
Content-Disposition: attachment;  
  filename="Priloha1.pdf"  
  
JVBERi0xLjUKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFivRm1sdGVyIC9GbGF0ZUR1  
Y29kZT4+CnN0cmVhbQp4nE2MsQoCMRBEUa/ar9gyKW7djUl20wo2dko6sTrQksX9F20iHDiv
```

Fig. 109: *Loopback social engineering email source.* Unless possessing at least basic knowledge of ICT, the recipient would not be able to validate message authenticity. Date and user agent string were redacted.

Source: Own work.

The attachment was encoded using base64 scheme which would get assigned a score of 1.8 according to Figure 94 had it been received from an external source. Decoding the data back into readable form reveals the obfuscated stream contains representation of the PDF attachment parsed and reconstructed on email load. Because there are few legitimate reasons to utilize

base64 apart from evading keyword-based spam detection, header analysis constitutes proof the message should not be trusted to come from the alleged source. However, recipients cannot be expected to inspect each and every incoming email for signs of malicious intent as the it would reduce productivity and user comfort.

Social engineering can be used to exploit known behavior patterns. Referred to as “dancing pigs,” the assumption states that “if [user] clicks on a button that promises dancing pigs on his computer monitor, and instead gets a hortatory message describing the potential dangers of the applet – he’s going to choose dancing pigs over security any day. If the computer prompts him with a warning screen. . . he’ll click OK without even reading it. Thirty seconds later he won’t even remember that the warning screen even existed” (Schneier, 2000, p. 262). System administrators must limit adverse effects of such uninformed decisions by imposing strict safeguards, e.g., locking users from installing and running obsolete software, mandating periodic update checks and patch deployment, or migrate to Virtual Desktop Infrastructure (VDI) which virtualizes endpoints and delivers computing service over the network. Notebooks and BYOD devices remain vulnerable in this scenario, though. Furthermore, security of data in transit would be more important with VDI, and repercussions of successful passive and active data intercept even more pronounced. Phishing, defined in chapter 2.4.5 and represented by emails claiming user’s mailbox quota is about to run out and their login is required to “upgrade” the account were delivered to employees at the faculty, but grammatical and stylistic shortcomings were sure to raise suspicion. The quality of Czech varies and the majority of phishing emails contain machine-translated strings which native speakers can detect trivially. Spear phishing directed at individuals which abuses their trust in technology could be efficient but necessitates prior intelligence gathering and reconnaissance. Should the attack succeed, though, the adversary gains superuser system foothold which may justify the added effort.

To summarize, a vulnerability in the email delivery system was identified which allows insider with a valid account to impersonate any user in the university domain. Using this technique, the perpetrator simply enters the From address presented to the recipient in the Return-Path field, and sends the message. A test scenario which involves a non-existent user account from which the email originated was executed, and the result is depicted in Figure 110.

At least four measures are currently available to prevent email spoofing: Sender Policy Framework (SPF), Sender ID, DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting and Conformance (DMARC). Technical details are beyond the scope of the thesis, but they validate the email address or tie it with the domain from which it was sent using a digital certificate.

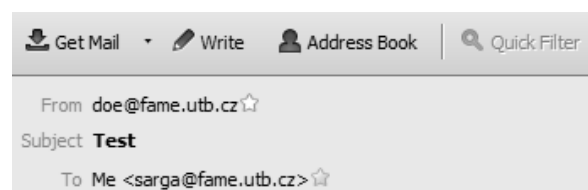


Fig. 110: Spoofed email as seen by the recipient. The email body was left empty and is not shown here. It has already been established that even executables, batch files, and PDF documents are not filtered. Source: Own work.

The only piece of information the message source revealed is that it was sent from within the university. It is depicted in Figure 111 with identifying data redacted.

```
Return-Path: <doe@fame.utb.cz>
Delivered-To: sarga@mailbox.utb.cz
Received: from sun.utb.cz (unknown [192.168.1.13])
  by mailbox.utb.cz (Postfix) with ESMTP id 4238578348E4
  for <sarga@mailbox.utb.cz>; Thu, 21 Nov 2013 15:33:23 +0100 (CET)
Received: from sun.utb.cz (localhost [127.0.0.1])
  by nod32.utb.cz (Postfix) with ESMTP id 36F50340939CC
  for <sarga@mailbox.utb.cz>; Thu, 21 Nov 2013 15:33:23 +0100 (CET)
X-Virus-Scanner: This message was checked by ESET Mail Security
  for Linux/BSD. For more information on ESET Mail Security,
  please, visit our website: http://www.eset.com/.
Received: by sun.utb.cz (Postfix, from userid 1000)
  id 3596E34094197; Thu, 21 Nov 2013 15:33:23 +0100 (CET)
X-Spam-Checker-Version: SpamAssassin 3.2.3 (2007-08-08) on sun.utb.cz
X-Spam-Level:
X-Spam-Status: No, score=-5.0 required=5.0 tests=ALL_TRUSTED autolearn=ham
  version=3.2.3
Received: from [REDACTED] (unknown [REDACTED])
  by sun.utb.cz (Postfix) with ESMTP id ADF334094197
  for <sarga@fame.utb.cz>; Thu, 21 Nov 2013 15:33:17 +0100 (CET)
Message-ID: <528E19AE.7000700@fame.utb.cz>
Date: Thu, 21 Nov 2013 15:33:18 +0100
From: doe@fame.utb.cz
User-Agent: [REDACTED]
MIME-Version: 1.0
To: sarga@fame.utb.cz
Subject: Test
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit
```

Fig. 111: Spoofed email header. The *Return-Path* field lists the spoofed source address instead of the attacker's. Should a reply be sent, an SMTP error would be generated.

Source: Own work.

Even though the email passed through the SMTP server, SpamAssassin did not assign it a score, all tests were bypassed and the ALL_TRUSTED flag set. This leads us to conclude emails originating from and delivered to the domain accounts are not checked for presence of spam, automatically classified as “ham,” and whitelisted. They still undergo scanning by ESET Mail Security for Linux/BSD which should result in a warning in case malicious content is found. Currently, the email delivery software configuration is excessively permissive and assumes user accounts will be used for legitimate purposes, a premise which sidelines checks with trusting large amount of individuals with potentially malicious intentions. Considering every student and employee at the university has their own account which they may access remotely from malware-infected machines, unencrypted wireless networks, and mobile devices lacking fundamental security practices, IT personnel should strongly consider replacing implicit trust with explicit policies, and weigh the advantages and disadvantages of including all emails in the antispam and antivirus scans. Processing overhead may be substantial but protecting sensitive data must be prioritized over low resource utilization.

5.2.4 Phase 4: Conclusion

The case study will conclude with a brief overview of results obtained in previous phases. Based on the findings, the threats which need to be addressed include centralized BYOD policy, password management, patch deployment, limiting user privileges, and server-side infrastructure modifications. BYOD policy will be discussed in chapter 6: spanning profiles, enforcing secure channels when interacting with sensitive electronic assets, password requirements tailored for touch interfaces, periodic software update checks, wireless connectivity management, temporary data retention and expiration policies, and countermeasures in case the device is lost or stolen

must be incorporated into a profile pushed onto the smartphones and tablets. Employees should be thoroughly informed about the permissions and restrictions imposed on the technology they own, and the fact the profiles only serve to secure transfer and handling of sensitive data across untrusted networks without any repercussions for personal use stressed. Otherwise, the disinclination toward the BYOD profiles uncovered in chapter 4.2.2 may hamper adoption.

Password management, particularly assignment of default authentication tokens, is a serious vulnerability allowing the adversary to brute-force her way into the internal corporate network domain and impersonate legitimate users. The strings at the FaME are currently based on the individual's birth number, a fixed-width numeric sequence with predictable structure and properties which permit search space pruning, limiting the time before unauthorized access is obtained. A scheme to decouple passwords and identities via (pseudo)random one-time sequences, prompting users to change their passwords after first successful login. Strength meters visually communicating string resilience based not only on zero-order entropy but other measures (repeated patterns, presence in word lists), should be implemented. It is further argued intuitive and easy-to-use credentials managers may result in higher quality passwords. A solution will be presented in the following chapter.

Patch deployment was identified as a weakness: both targets were found running obsolete software, the second one relied on versions no longer supported by the respective vendors. This makes it prone to vulnerabilities fixed in later releases unless the security patches were integrated manually. The approach is not justifiable as overly conservative: the software base was found not just dependent on insecure components, but also on substandard security practices such as accepting user login credentials without applying any encryption which makes them vulnerable to passive interception. Coupled with BYOD proliferation and uptake in wireless Internet access, basic hardware and software are sufficient to launch potent attack scenarios. Setting up a test server where dependencies (discussed in chapter 5.1) and stability issues can be assessed in a controlled environment is trivial with virtualization capabilities. Nmap, one of the utilities used in the case study, ascertained both hosts were run as virtual machines.

Currently, each user has administrative privileges on their local workstation or notebook, exercising full system control. This not only turns them into viable targets, significantly expands the attack surface and fosters environment where a single successful exploit threatens the security of the whole network, but also allows to trivially bypass system-level policies. An example is User Account Control (UAC), a technology which relegates untrusted software to lower-privilege mode until overridden by administrator, disabled on the majority of the machines to increase user comfort. Furthermore, software updates were habitually ignored. Platforms such as Acrobat Reader, Flash, and Java are pervasive and receive frequent updates which address critical vulnerabilities, but users are given the authority to override the process by either disregarding or ending it. This opens the underlying OS to remote unsanctioned code execution, giving perpetrators and insiders a means to expropriate sensitive data, install tracking software, and use the machine as a stepping stone for malicious actions. Those will be logged and associated with the victim who has the burden of proof. The solution is to lower the superuser privileges and enforce periodic update checks to mitigate known security risks.

Server-side, the IT personnel should reduce the verbosity of error messages and information available to third parties. Disabling directory listing, custom error pages, requesting cache purges from popular search engines, and hardening the internal network so that users are treated as malicious mitigate some security concerns found during the VA/PT phase. Firewall rules can be set to identify port scans by matching incoming packets against templates and drop them automatically. Leaks cannot be fully prevented because some information are publicly available, but sanitizing metadata from documents, obfuscating email addresses, expiring sessions when

idle for a specified time, and requesting HIP-based authentication are just a few steps to minimize impact on legitimate users, but challenging the adversary. While the thesis will not delve into details of technical hardening, chapter 6 discusses some precautions.

Quantifying financial impact of a breach is challenging. One way is to equal the monetary loss to the value of data the attacker can access and expropriate should they penetrate the system, but organizations can rarely operate with precise figures and can at most rely on estimates from disclosed known cases. Apart from explicit costs (audits, ICT infrastructure hardening, employee training, VA/PT), loss of customer confidence and revenue from clients dissuaded from dealing with the organization due to the breach together with reputation risk also contribute and should be reflected in the calculations. Another variable unique to every incident is the value of the data: objective, standardized, universal baseline does not exist. The probability of a breach increases with factors such as whether employees store and encrypt electronic assets on portable devices (removable drives, notebooks, tablets), types of sensitive data processed, organization's industry classification, and global footprint (local, international), but quantification methodologies vary. For example: the following information were entered into a 13-step online form provided by Symantec Corporation⁵:

- industry classification: education,
- privacy and data protection security policy: none implemented,
- types of sensitive information handled: citizen records, employee records, student information,
- most likely cause of breach: malicious or criminal attack,
- employees storing sensitive data on laptops/removable storage: yes,
- encryption of sensitive data on laptops/removable storage: no,
- dedicated information security officer: no,
- global headcount: less than 500,
- global footprint: operations in one country,
- remote access: using either personal or corporate devices,
- strong authentication measures: none,
- headquarters: all others (used for currency code),
- estimated records at risk: 25.001–50.000.

The results denote the likelihood of a breach in the next 12 months to be 9.9 %, with average cost per record 146 USD and average cost per breach 5470833 USD. However, it was pointed out "... most companies experience opportunity costs associated with the breach incident, which results from diminished trust or confidence by present and future customers" (Ponemon, 2013, p. 21). The calculator also "... utilizes a confidential and proprietary benchmark method" (Ponemon, 2013, p. 24) which prohibits academic and expert evaluation. The research was supported by Symantec, a commercial entity with a portfolio of security enterprise-oriented products, a clear conflict of interests. The methodology's shortcomings comprise non-statistical sample, non-response, sampling-frame bias, company-specific information, unmeasured factors, and extrapolated cost results. The first one prevents applying statistical tests, and the estimates serve as black-box estimates not suitable as precise inputs to organization's decision-making processes.

Overall, the lack of technical sophistication exhibited by the model insider should not prevent continuous auditing, evaluation, and improvements in all aspects of ICT. A comprehensive range of tools offering advanced exploitation techniques is available free of charge along with tutorials and documentation. Questionnaire research in chapter 4 proved a non-negligible proportion

⁵<https://databreachcalculator.com/Calculator/>

of respondents would be willing to engage in malicious activities if a financial stimulus was involved, or if their conduct was not revealed. Even though it is not reasonable to assume the respondents will at any point actually engage to such conduct, it nevertheless hints at online space being perceived less threatening compared to the real-world settings when it comes to illegal behavior. Preventative measures must therefore be implemented at the hardware, software, and human level to deter and protect against the threats originating from insiders and outsiders.

The question posed in chapter 5.2.1 was: **What relevant and usable data about the target's ICT infrastructure can an attacker who possesses low to moderate ICT security gain using freely-available software?** The case study demonstrated actionable intelligence can be obtained by anyone with basic knowledge of ICT, and potent scenarios launched using very few commands. It is the author's opinion the barrier to entry is very low and organizations should focus on major avenues the adversaries can exploit, namely balancing security with user comfort, employee training, strictly enforcing the principle of least privilege, and external auditing.

The following chapter proposes an ICT security governance model which covers both user-side and ICT-side security, and aggregates findings from the questionnaire research and case studies. The model lists best practices and solutions which should lead to improvements in security and may prevent financial costs if implemented in practice.

6 THE ICT SECURITY GOVERNANCE MODEL

In this chapter, the main methodological contribution of the thesis will be presented and discussed based on findings from the questionnaire and case study research. Despite being theoretical in nature, the model's constituents have already been detailed in both practical and theoretical part; here, the results will be aggregated into a cohesive structure and dependencies to organizational processes delineated. As a reminder, chapter 2.1.3 classified business processes into four categories: management, operational, supporting, and business network processes. ICT was agreed to be a supporting process which permeates others and enables value to be created for internal and external customers. In the value chain model pioneered by Porter (1985) and depicted in Figure 112, ICT is grouped in supporting activities related to Technology Development.

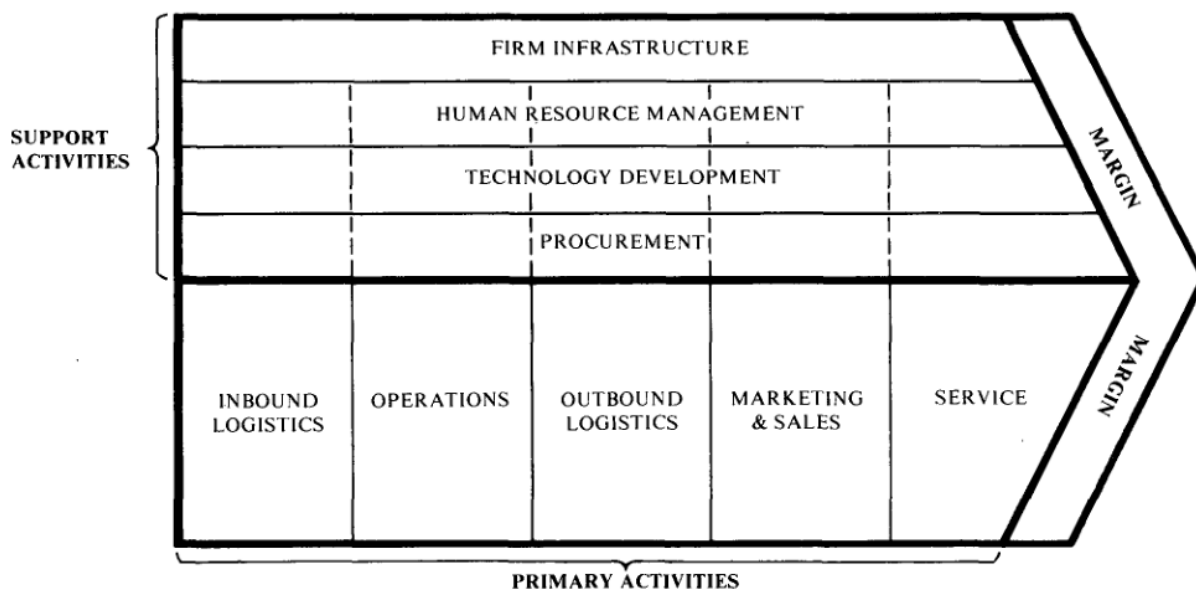


Fig. 112: Porter's value chain model. Primary activities add value to the inputs, support activities manage human resources, technology, and supplier relations.

Source: Porter (1985, p. 37).

It was stated “[i]nformation systems technology is particularly pervasive in the value chain, since every activity creates and uses information... Information systems technology also has an important role in linkages among activities of all types, because the coordination and optimization of linkages... requires information flow among activities. The recent, rapid technological change in information systems is having a profound impact on competition and competitive advantages because of the pervasive role of information in the value chain” (Porter, 1985, p. 168). Dependence on ICT has increased to the point where its correct functioning is not an enabler but a prerequisite for generating added value. Adequate ICT protection is thus necessary to ensure business continuity. Organizations which perceive technology as a ubiquitous, always-on service rather an area in need of investments are more likely to be affected by the negative aspects, e.g., electronic crime and insider threats.

The proposed model draws from an ICT strategy which should be in line with other organizational documents, but the framework also takes input from stakeholder groups, users and ICT personnel as well as external stimuli. Building on cybernetics (discussed in chapter 2.1.1), the organization is understood as a complex system interconnected to and interacting with outside environment. The connections are physical and virtual: the latter allow authenticated third parties to access information systems on a per-user or automated basis such as supply chain management

links. It is not reasonable to expect users would be interested in technical details of how such communication works but rather in the fact it provides the desired service. The model enforces separation between implementation, a specialty of ICT personnel, and a non-technical approach directed at users. Often, the two are mixed together in organizational documents which attempt to reconcile the two domains instead of tailoring policies for both target groups.

Before the model is represented graphically, its prerequisites will be outlined. Violating any of them will result in substandard performance, implementation difficulties, and diminished effects:

- organizational structure. The model presupposes organizational structure where policies and strategies are formulated and upheld, and stakeholder groups, employees and IT personnel have consultative power to influence the decisions. Hierarchical structuring is not required as long as authority exists with a decision-making mandate. Organizational structure does not have to be formalized but should be loose enough to give IT personnel authority for security-related operational decisions in non-standard situations. The scope of such authority should be delimited in corporate strategy.
- vision, mission, strategy. The first two fundamental documents are outside the thesis' purview, strategic plans which allocate resources to fulfill the goals and objectives must exist and be pursued. ICT should also be treated equally to other supporting processes with an investment roadmap based on recommendations from the CISO (mentioned below) or equivalent party.
- business process orientation. Emphasizing processes for bringing value to the customer rather than hierarchies is necessary. Defining process owners, responsibilities, and the service delivered helps to ascertain ICT's place in the process map, dependencies and response procedures in case an anomaly occurs.
- ICT utilization. The model is based on the assumption ICT supports or enables execution of business processes. For organizations critically dependent on its flawless functioning, implementing the framework should foster flexibility, incident response, and infrastructure resilience while not substantially decreasing user comfort.
- change management. Process owners and customers need to be aware and approve of the transition from one system state to the other via gradual adjustment, inclusive mindset, seeing positive consequences of the transition period as well as transparent communication. Altering ingrained patterns will be met with resistance, particularly when the previous state fulfilled the needs and expectations, but identification and engagement of key users (opinion leaders) in a small-scale testing, stressing the increases in productivity, thorough and repeated training, and encouraging questions, criticism and feedback may persuade stakeholders to be open-minded about the change.
- CISO. Even though the Chief Information Security Officer may be redundant for smaller organizations, at least one IT employee should be dedicated solely to security. The area is markedly dynamic and discrete policy revisions are inadequate because the documents quickly fall into obsolescence, depending on the industry and the attack surface. The CISO should be the link between management, users, IT personnel, and stimuli coming from beyond the organizational boundary.
- incident response. Business continuity planning is a prerequisite for scenarios where ICT threat potential is realized and affects one or more processes. Examples include targeted (DoS) or physical (severed cabling) connectivity drops, hardware/software failures in mission-critical systems, large-scale malware infections, natural disasters, power blackouts, etc. Redundant resources must be activated with minimum delay and coordinated actions invoked as specified in contingency plans. Even operational anomalies (detecting port scans, social engineering campaigns in progress) should be appropriately responded to.

- employee training and development program. The organization should facilitate training and development programs with a curriculum suggested in the model. ICT auditing and business continuity plans should be periodically conducted under controlled conditions to uncover weaknesses, and the results used to improve the documents with corrective actions. The scenarios may include fake malware infection, simulated internal network breach to test intrusion detection algorithms, vulnerability assessment, penetration testing, real-world training sessions, controlled connectivity drops, and social engineering campaigns. Providing the target group with real-world experience helps to stress the implemented changes are meaningful and necessary.

Human factor is critical and represented in all prerequisites. Hardware and software, albeit important, are configured and managed by people and therefore a function of their knowledge and skills acquired by training according to the human resource development strategy. Technology usually has a finite number of states and its output is predictable with high probability unlike that of people whose behavior patterns cannot be ascertained, and are modified only via long-term efforts. The model does not delve into specifics of what training methods are effective but it is implicitly assumed they are known and applied. ICT benefits from experimentation, demonstrations, and hands-on involvement showing how the technology critically depends on the operator. As each employee is a viable attack vector, training should encompass everyone regardless of their position in the organizational structure.

The model is depicted in Figure 113. The dependencies are demonstrated by unidirectional and bidirectional arrows, depending on how the interaction is expected to take place. Of particular interest are external stimuli which affets user-side and ICT-side in one direction because an argument can be made that infrastructure hardening influences organization's surroundings. If the adversary conforms to the model in chapter 2.4.1 and her objective is damage maximization, resilient ICT decreases the probability of a breach while increasing the likelihood an alternative victim will be selected instead, an example of how measures deployed internally have consequences for other entities. However, such scenario does not consider insider threat and industrial espionage where the malicious actors are recruited from within the organization, or hired to expropriate sensitive data for commercial purposes. Here, the target is probed repeatedly until an uncontrolled vector is discovered, a much more realistic scenario reflected in the arrow pointing toward user-side and ICT-side, one of which frequently constitutes the weak link for unauthorized access. Social engineering in particular presents multiple vectors due to lacking training and development programs and trust-based employee conduct toward third parties. Insider are classified as external threat: while originating from within the organization, they violate the security perimeter around the sensitive assets as opposed to weaknesses in hardware and software which threaten the CIA triad's constituents internally (chapter 2.2). Infrastructural deficiencies arise from sources outside of organization's control: software bugs are addressed in patches (patch management and deployment is governed by the ICT policy), hardware failures and replacements are outside the scope of the thesis.

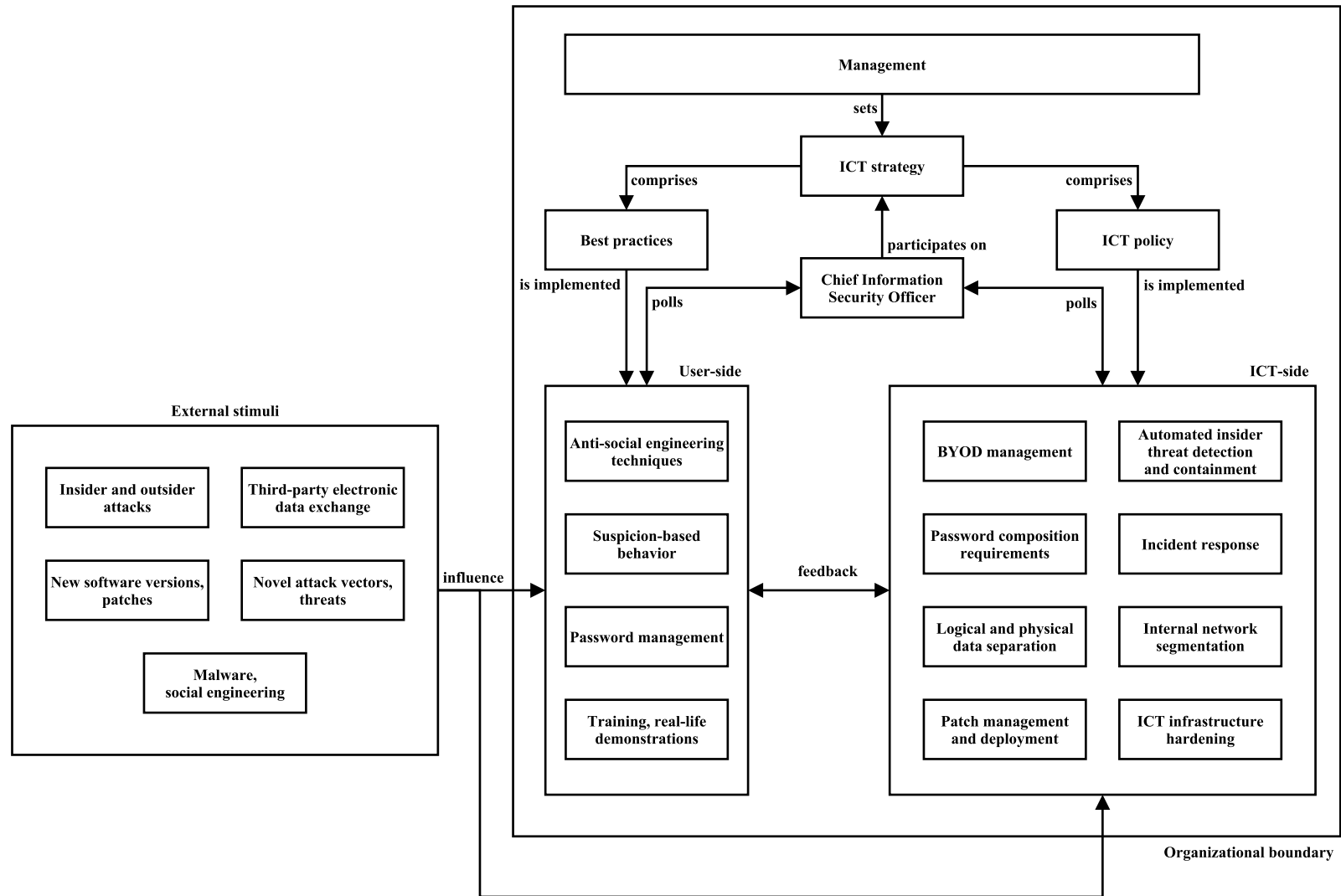


Fig. 113: The proposed ICT security governance model. Organizational boundary depicts the border between external and internal agents. Source: own work.

6.1 User-Side Security

The chapter details user-side security elements in the model. Product and service names do not constitute endorsements but are meant to be representative of the functionality expected and recommended. Free and open-source alternatives to commercial solutions will be provided at the time of writing, but software landscape is dynamic and should the model be used to design the best practices and ICT policy, up-to-date research needs to be conducted before deployment. Also of note is that despite sophisticated technical means present in the production environment, a weak link in synergy of hardware, software, and people with the emphasis put on the last group is likely to render the whole organization vulnerable.

User-side security consists of four recommendations: anti-social engineering techniques, password management, suspicion-based behavior, and training and real-life demonstrations. The last two are coordinated programs to increase competencies, knowledge, and skills which minimize exposure of internal corporate assets in the organizational network through social engineering (chapter 2.4.5). No method conveying the information to employees will be evaluated in detail because employee training is a broad discipline cursory to the thesis' main focus. With sophisticated means implemented to protect sensitive data, rational attackers (chapter 2.4.1) will prefer vectors promising maximum utility by expending minimum effort, a behavior which complies with the economic theory presented in chapter 2.1.4. Preventing disclosures of benign information which can be valuable in the right circumstances, eliminating cognitive biases when dealing with unauthenticated third parties, encouraging suspicion in place of trust as well as training and reinforcement are ways to mitigate information leaks.

Cognitive biases such as halo effect which sees people "... apparently affected by a marked tendency to think of the person in general as rather good or rather inferior and to color the judgments of the qualities by this general feeling," (Thorndike, 1920, p. 1) Forer effect which makes persuading an individual by general statements possible due to subjectivity, (Forer, 1949) or defensive attribution hypothesis which purports that "... the tendency to assign responsibility to someone possibly responsible for an accident increases as the consequences of the accident become more serious" (Walster, 1966, p. 6) confirm security is a process, not a product or a final state the organization can achieve (Schneier, 2000). Mitnick and W. L. Simon (2003, pp. 12–13) add that "... the human factor is truly security's weakest link... As developers invent continually better security technologies, making it increasingly difficult to exploit technical vulnerabilities, attackers will turn more and more to exploiting the human element." The human attack surface is challenging to reduce due to psychology governing the patterns of human behavior. Examples of techniques used to extract information and reconnaissance the target were presented in chapter 2.4.5.

Educational and training programs must mention information disclosure risks and critical role of each employee for security. Hughey and Mussnug (1966, p. 2) note the distinction between education and training: "Education typically takes place in a classroom and involves a transfer of knowledge through the use of formal methods such as lectures and directed discussion... Training, on the other hand, typically entails personal involvement, commitment, and experiential gains. Training involves training by doing... True training occurs when skills that can be measurably defined are enhanced until the competence level is visibly enhanced." Competent employees are less likely to cause false negatives and false positives in non-standard situations. The former occur when a third party is denied access to an asset despite meeting all the prerequisites, false positives occur when access is granted even though some of the requirements were not met. Both constitute security failure and may incur penalties. A sample curriculum in Table 19 comprises core competencies employees should possess to reduce the risk of social engineering techniques being successfully executed.

Tab. 19: *Sample employee training curriculum. Real-world examples may help employees realize pervasiveness of social engineering in routine situations.*

Source: Own work.

Module	Name	Short Description
1	Basics of psychology	Introduction to how human behavior follows observable logic which could be exploited maliciously.
2	Cognitive biases	The shortcuts taken when assessing unknown parties (Forer effect, halo effect).
3	Information leaks	How publicly-available sources can help to instill trustworthiness in the victim.
4	Basics of social engineering	Terminology (phishing, spear phishing), exploits in personal interactions and phone conversations.
5	Phishing	Detailed overview of techniques employed to convince the victim that the verbal or written claim is genuine.
8	Human exploitation	Practical examples of how the adversary can modify her behavior to activate the shortcuts from module 2.
9	Pretexting	Injecting a victim into a scenario where reasoning gives way to emotional response and heuristic thinking.
10	Countermeasures	What can the victim do to sideline instinctive reactions in favor of logic to counteract phishing, human exploitation, and pretexting.
11	Case study	A demonstration of how chain of events in a social engineering campaign leads to system compromise.

The expected result is for the attendants to prefer suspicion-based behavior in electronic communication and personal interactions. This requires a mix of technology sophistication, attained by ICT courses and certifications, and changes in behavior response reinforced by repetition, and testing selected employees who participated in the course to get feedback on how effective the curriculum has been. However, the management should not expect significant improvements after a single run, measured by comparing pre- and post-training performance. Continuous involvement, controlled scenarios the employees are subjected to, and ingraining the desired patterns into everyday conduct over a long-term will result in observable advancements in psychological human resilience.

From the ICT standpoint, another point of weakness is password management. Even though IT personnel can enforce strict composition rules, widespread password reuse and shortcuts in handling personal sensitive data give rise to legitimate concerns. Some practices cannot even be covered by organization-level policies, e.g., storing unencrypted strings on employees' mobile devices for easy access. Restricting applications which allow users to type in plaintext input via BYOD profiles is unduly restrictive, reduces the device's usability and introduces needless obstacles. Even then, ways exist to circumvent the measure, such as by storing the text in

a contact list database. Requirements for optimum solution are thus:

- user should not be required to install and manage additional software; updates should be deployed automatically,
- seamless webpage integration, form-filling capabilities,
- robust password-generation algorithm,
- portability, device- and platform-agnostic synchronization,
- encrypted centralized storage of user credentials,
- fostering the concept of security as a service (SECaaS),
- electronic identity and access management.

Moreover, the solution should have the following ICT-side features which will make integration into the ICT infrastructure seamless:

- batch addition, deletion, and modification of user accounts,
- support for distributed directory information services,
- the ability to run the platform locally for backup purposes,
- strong encryption,
- customizable password-generation algorithm,
- global policies for generating, expiring, and revoking passwords,
- optional multi-factor authentication,
- reporting,
- portable database file format to avoid vendor lock-in.

Cloud services and virtualization are two prerequisites for the security as a service (SECaaS) concept. Both were discussed in chapter 2.2.3: the former refers to a pool of scalable resources provisioned according to client's specifications and deployed over the Internet in a pay-per-use business model, virtualization is a technology capable to emulate hardware or software abstracted from the underlying physical medium. For the proposal to work, cloud computing will transfer user files geographically close to their actual location, and the VM will host the security system in a logically-separated instance which reduces the risk of unauthorized access and data expropriation. The model also lists password composition requirements, ICT infrastructure hardening, and internal network stratification as concerns which must be addressed in the organizational ICT policy; cloud computing and virtualization address the issues as well. Disadvantages deserve a mention, though: sacrificing physical control over data and hardware by off-premise relocation, replication across multiple data centers which makes permanent data removal challenging, security concerns for increased volume of data in transit, processing overheads in case resource-intensive encryption is used, and the need for high-speed, redundant, high-availability physical connections. An alternative would be to utilize existing ICT infrastructure for local deployment, but the decision is outside the scope of the thesis which is focused on methods minimizing user discomfort while reducing the sum of exploitable vulnerabilities, i.e., the attack surface.

The model builds on the premise employees consider security as a service rather than a product, and use it without knowledge of low-level details which should be hidden from them, similar to the black-box principle. Decoupling human factor from security to the highest extent possible lowers the probability of employees becoming proxies through which the adversary infiltrates corporate information systems and gains unfettered access to sensitive electronic assets.

However, black-box models may lead to distrust if the underlying principles are not clearly communicated. Striking a balance between generality and specificity should not be underestimated; if opinion leaders form a negative opinion at the onset, other users will adopt the view as their own and supplant personal experience with that of a trusted party which prolongs the change

management phase and could result in rejection of the new system. For example: password meters distinguish various levels of strength based on zero-order entropy (equation 6.2.2), but users do not get to see how the value is calculated, only a visual indication which may prompt questions about how the algorithm functions. A concise help manual along with the option to select alternative, more time-intensive means of evaluation, e.g., predictable patterns (repeated characters, appended/prepended numerals, capitalized first letter), match in a server-stored database of leaked passwords, or known weak strings (12345, password) will make the process transparent so that users trust the security system implicitly. Lack of trust may lead to unforeseen consequences: subverting the system with “trusted” solutions, writing passwords down, and selecting strings tying user to the password (place of residence, birth number, spouse name) instead of random data exhibiting high uniqueness, length, and unpredictability.

Ideally, users should not be able to see the string at all and memorize it, the only exception being the authentication token for the encrypted password database. Two points of failure exist in such an arrangement: implementation omissions and the need for a master password or authentication. Implementation omissions acknowledge software as insecure because it is developed by people: while formal verification methods are available for auditing the code and certifying it as security compliant, commercial software rarely undergo such scrutiny due to high price and time requirements, and weaknesses in the program thus have non-negligible probability of occurring. Cryptographic algorithms used to encrypt the data may be sound but the secret key can leak via an uncontrolled software interconnect, rendering the mathematical properties irrelevant. The need for a master password or authentication stems from the fact access to the database which stores the sensitive data is permitted solely to the respective owner who needs to authenticate themselves. Should it be lost, even IT personnel cannot recover the contents. Biometric scans which uniquely identify the user based on physical features can augment or replace traditional string-based authentication, obviating the need for memorization or storing the password in paper form. Nevertheless, their disadvantages include dedicated hardware components and permanent feature compromise. While the first one has been gradually diminishing due to economies of scale, permanent feature compromise is a flaw originating in the system design: if the adversary somehow gets hold of the physical characteristic (retinal scan, fingerprint, voice sample), she can impersonate legitimate user with no way of revoking the token. While time-based tokens partly solves the problem, once the data is in attacker’s possession, biometric authentication ceases to be trustworthy. Moreover, false positive and negative rates result in the following two scenarios:

- false positive: an entity who should not have been granted access is authenticated successfully; various methods have been proposed, e.g., silicone fingerprint impressions (Matsumoto, Matsumoto, Yamada, & Hoshino, 2002) and probabilistic iris reconstruction (Galbally, Ross, Gomez-Barrero, Fierrez, & Ortega-Garcia, 2012),
- false negative: an entity who should have been granted access is not authenticated; if the system has threshold on the number of failed attempts, the flaw could result in a DoS attack.

False negative rate is decreased by storing multiple copies of the physical characteristic for comparison purposes, false positive rate could be decreased by a hardware revision utilizing more precise components which, however, still leaves current models susceptible and incurs additional costs. Data expropriation cannot be ruled out as well if the ICT policy does not cover the incursion vector nor provides steps to secure the data in transit and at rest. Biometric database compromise a serious security incident which should be planned for and a fallback identity-verification scheme put in place immediately.

Some users may have objections with providing personally-identifiable biometric data which is stored out of their control. Despite the advantages of cloud computing mentioned earlier in

chapter 2.2.3, the argument is legitimate because cloud operators are not required to disclose internal policies, preferring security through obscurity rather than security through openness. Even if they disclose the policies, however, data in transit needs to be encrypted from the moment it leaves the device up to and in the data center. The links are schematically depicted in Figure 114.

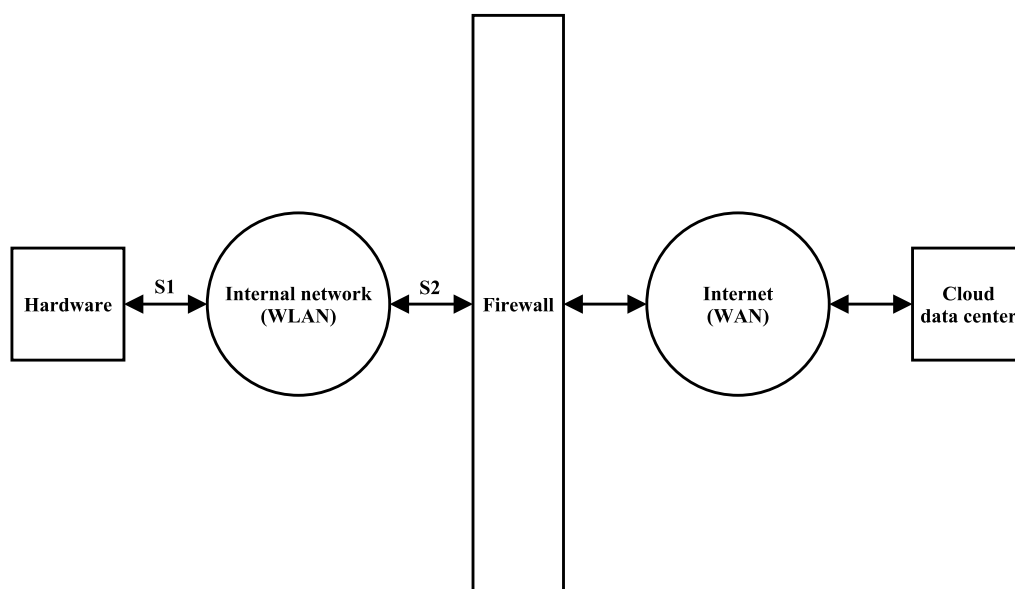


Fig. 114: Sensitive data encryption start points. If the data leaves the internal network unencrypted, it can be intercepted by any insider, on the Internet as well as by the cloud operator.
Source: own work.

The start points where sensitive data could be encrypted before it is sent to the internal network are on the user's machine (S1), or before they leave the network (S2). S1 should be preferred to thwart insider threat because malicious users can otherwise intercept the packets and reconstruct the message. An ideal situation would see S1 as a point of encryption for all communications, even if the data moves over the internal network. Increased processing overhead is counterbalanced by abundance of spare CPU cycles in PCs, notebooks, mobile devices, and other hardware components. While the model does not require the assumption, strong encryption scheme mitigates many of the threats malicious insiders pose for sensitive electronic assets.

Of the enterprise-grade password managers available at the time of writing, the one meeting the majority of the requirements put forward previously is LastPass. The service synchronizes user passwords with a remote data center by encrypting it locally first, i.e., at S1 in Figure 114, which protects the stream from both insider threat and interceptions over untrusted Internet channels. LastPass webpage contains a form which allows to calculate expected return on investment and days to break even, i.e., before the financial benefits equals purchase costs. Three pieces of information are required: number of employees, inclusion of Security Assertion Markup Language (SAML) add-on, and hourly wage per employee. Figure 115 depicts the form.

For more refined calculations, income classes for university employees were included at the time of writing. As of December 1, 2013, Faculty of Management of Economics, Tomas Bata University in Zlín had the following workforce structure: 6 professors, 23 associate professors, 36 senior lecturers, 15 assistants, 6 lecturers, and 27 non-academic employees. For each group, income tariff has been converted from Czech crown (CZK) to United States dollar (USD) using a fixed conversion rate of 20 CZK/USD. In case multiple tariffs were available, the lowest income was preferred and as such, the calculations represent the lower bound on return on investment

Number of employees using LastPass	<input type="text" value="113"/>	Employees
Include SAML Add-on	<input type="button" value="No"/>	
Average logins/employee/month*	220	Logins
Average time/login (seconds)*	15	Seconds
Average time spent per employee per month on logins (hours)	.91667	Hours
LastPass Price Per User	\$20.00	Price Per user
Hourly wage per employee (input your average)	<input type="text" value="10"/>	USD Per Hours
Annual time value spent on logins	\$12,430.05	USD Per Year
LastPass annual investment	\$2,260.00	USD Per Year
Annual recurring ROI	\$10,170.05	USD Per Year
Days to Break even	66	Days

Fig. 115: LastPass return on investment calculation form. The data shows that for 113 employees at the faculty with average wage of 10 USD per hour, the total savings amount to 10 170.05 USD and break-even point is expected after 66 days.

Source: Own work.

and break even. Each employee is expected to spend at least 40 hours per week on work-related tasks, totaling 160 hours per month, again denoting a lower bound. Hourly wage HW is then obtained as:

$$HW = \frac{INC}{WT}, \tag{6.1.1}$$

where INC denotes the income level, and WT the working time. Results are demonstrated in Table 20. LastPass supplies data for average logins per day (11) and average seconds per login (15) based on a sample of 3 500 enterprise clients. It is assumed each employee will use the service if deployed.

To demonstrate, the results for non-academic employees in Table 20 are interpreted as follows: should LastPass be purchased for 27 users with respective average wage and working time, annual return on investment would amount to 2322.01, and 80 days would be needed to break even. Summing individual values is not logical but number of employees and average wage influence the figure. A crude measure which would encompass all 113 faculty employees is to input average of the hourly wage groups (3.418 USD): return on investment and days to break even are 2 164.57 USD and 97, respectively. However, both figures represent lower bounds, and because income levels are fixed, increased working time lowers the average hourly wage and thus both return on investment and break-even point. Dependency chains are as follows (ceteris paribus):

- \uparrow income $\rightarrow \uparrow$ average hourly wage $\rightarrow \uparrow$ return on investment $\rightarrow \downarrow$ break-even point,
- \uparrow working time $\rightarrow \downarrow$ average hourly wage $\rightarrow \downarrow$ return on investment $\rightarrow \uparrow$ break-even point,
- \uparrow employees $\rightarrow \uparrow$ total time spent on logins $\rightarrow \uparrow$ return on investment $\rightarrow \downarrow$ break-even point.

Provided a per-user LastPass license costs less than the annual time spent on logging in, the system is economically justifiable. The second condition of economic viability is return on investment compared to days:

Tab. 20: LastPass return on investment calculation. For lecturers and non-academic employees, lower income class was included and the results thus represent the worst-case scenario for the two groups. Source: Own work.

	Number of employees	Income (CZK)	Income (USD)	Working time (hrs)	Average hourly wage (USD/hr)	Return rate (USD)	Break even (days)
Professors	6	26350	1317.5	40	32.9375	2029.88	24
Associate professors	23	22800	1140.0	40	28.5	6658.53	28
Senior lecturers	36	18200	910.0	40	22.75	8145.03	35
Assistants	15	17000	850.0	40	21.25	3146.26	37
Lecturers	6	17000	850.0	40	21.25	3146.26	37
Non-academics	27	8000	400.0	40	10.0	2322.01	80

- if return < 365, the investment is recommended,
- if return = 365, LastPass expenditures equal cost spent on logins,
- if return > 365, the investment is not recommended.

The second case is ambiguous but should be viewed as an indication the current security model is sufficient return-wise because change management would incur additional costs and decrease productivity due to user adjustment period. The higher the (return – 365) spread, the higher the savings. Nonetheless, factual accuracy of the results depends on average employee logins per day and average seconds per login in the organization. LastPass provides no information about the companies included in the representative sample, including industry, user base and ICT literacy, security habits, how long LastPass has been deployed (a new system may initially exhibit slower login times because users are not familiar with it), etc. An on-premise pilot study to determine how the figures differ in the organization is a basis for accurate results and exact return on investment calculations.

Alternative password management applications exist for various platforms, the following list does not attempt to be exhaustive: 1Password, Clipperz, Dashlane, KeePass, Keeper, RoboForm, SplashID, and Sticky Password. User-side and ICT-side requirements should be taken into account, especially regarding synchronization and enforcement of password policies. Storing a copy of the password database on a notebook or a mobile device is strongly discouraged if password composition requirements were not complied with when creating the string as the device becomes a point of failure when misappropriated. The adversary can attempt to reverse engineer the token (demonstrated in chapter 5.1) and gain access to the file. Moreover, should the database file be inadvertently corrupted due to hardware failure or accidentally deleted without backup, passwords are irrecoverably lost. Coupled with the suggestion users should not be able to see and memorize their passwords, this would equal losing a sizable portion of one's electronic identity. Cloud computing alleviates the issue by storing the database off the device for remote access and management. Compliance with password policies also guarantees the adversary needs to spend prohibitive amount of time attempting to crack a single user's database in an online scenario. Because every password was generated automatically using a strong algorithm,

a feature LastPass supports, moving on to a different account does not increase the probability of success sampled from a uniform distribution unless a vector such as keylogging malware is planted.

The most effective way to address leaks is therefore to engage employees in training of anti-social engineering techniques, and decouple them from organizational security to the highest extent possible. Separate information classes can be introduced:

- public: available on corporate website, documents, brochures, and on the Internet; includes telephone numbers, email addresses, organizational structure, office address, employee names, customer and supplier lists, disclosed legal documents (financial statements, quality compliance certificates, audit results, Articles of Association), industry reference books, standards, terminology, etc.,
- internal: available for authenticated users; includes customer data, financial records, strategic and product plans, system access credentials, internal communications, memoranda, directives, manuals, handbooks, administrative documents, organizational policies, etc.
- sensitive: limited to specific subset of users, damaging if disclosed to third parties or disrupted; includes source code, research and development prototypes, preliminary strategic and financial plans, databases, mission-critical applications, etc.

Each successive class should involve more security checks, e.g., internal information shared only with full-time employees and vetted third parties. Sensitive information should not leave company's premises, be duplicated, and depending on the sensitivity, may be accessed exclusively from dedicated terminals not connected to the Internet (air gapping described in chapter 2.4) which allow plugging in removable devices only after multi-factor authentication involving human verification.

As a summary, the model suggests organizations should direct users to adopt the mindset that security is provided as a service, and decouple them from technical details with the exception of social engineering, the only attack vector directly aimed at the human element in the security process. Knowledge of how adversary is able to manipulate her victim to gain information or access for which she is not authorized (information systems, internal and sensitive assets, passwords) should be communicated in training courses and real-world demonstrations. Internalizing suspicion-based behavior and distrust toward third parties in personal and electronic contact necessitates long-term management involvement, repetition, and fostering employee feedback. IT personnel should not only enforce security policies but also supply solutions: generating a password which complies with the requirements, automatically setting the OS so that it conforms to the best practices, activating a BYOD profile which establishes a secure channel for interactions with internal resources, etc. While human factor will remain critical for security, burdening it with excessive requirements have detrimental effects on the attack surface. ICT can largely shift the responsibility away from the domain of psychology to a consolidated, controlled environment which aggregates majority of the functions users expect from security. Even though concessions need to be made, e.g., accepting the loss of physical control over data along with increased importance of ICT infrastructure management, in the author's opinion, the advantages outweigh the negative aspects. By viewing security as reliable, stable, trustworthy, and ubiquitous, organizations can reduce incidents involving unauthorized access, data expropriation, and identity theft. However, a synergy of user-side and ICT-side security is necessary to achieve the goal.

6.2 ICT-Side Security

The chapter details ICT-side security elements in the model. Less emphasis will be put on implementation details and technical background, and more on methodology. Case study 2

(chapter 5.2) mentioned omissions in selected faculty hosts which suggest the ICT policy is either non-existent or inconsistently deployed. Critical systems in production environment, particularly those accessible over the Internet, should undergo thorough scrutiny before deployment, and patch deployment should be made a priority. Nevertheless, insecure internal networks do not contain any countermeasures to stop the adversary or malicious insider from accessing internal and sensitive resources as per the delimitation presented in the previous chapter.

A concept titled defense in depth (DiD) is defined as “. . . practical strategy for achieving Information Assurance in today’s highly networked environments. It is a ‘best practices’ strategy in that it relies on the intelligent application of techniques and technologies that exist today. The strategy recommends a balance between the protection capability and cost, performance, and operational considerations” (NSA, 2010, p. 1). Originally, DiD was devised as a military strategy, and is depicted in Figure 116.

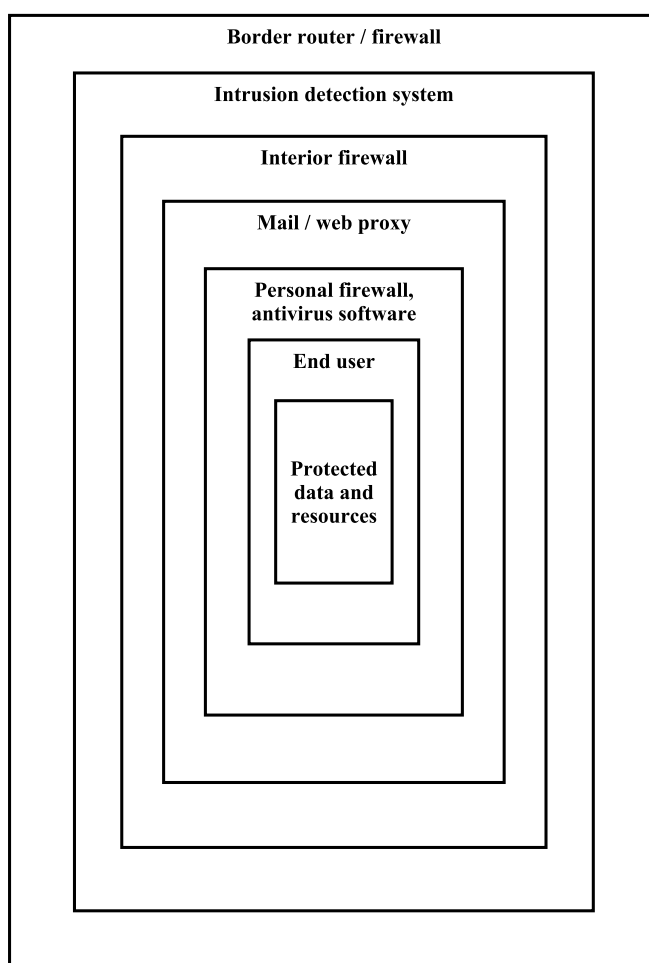


Fig. 116: *Defense in depth model. The attacker needs to penetrate several distinct layers to access protected data. The measures are technology-based except for end users which turns parties authorized to interact with the data viable targets for social engineering attacks. If successful, the perpetrator bypasses all other security barriers.*

Source: Harrison (2005, p. 2), modified.

The concept was criticized, for example because “. . . individuals, corporations, and government entities are being made victims of an attack strategy that is really more akin to [d]efense in [d]epth in reverse. The attackers provoke maintenance of a layered defensive stance that is massive, difficult to manage, requires extensive skill sets and is extremely costly. In essence, the attackers are forcing an unsustainable posture, exhausting resource and adapting advanced

persistent and advanced evasive techniques to slip right past [p]eople, [p]rocess and [t]echnology” (Small, 2012, p. 5). At the same time, human behavior was mentioned susceptible to social engineering which nullifies the effect of DiD by allowing the adversary to use the shortcut to reach the location of interest. The fact is acknowledged in chapter 6.1 where an advice was given people be decoupled from organizational security and subjected to continuous, extensive anti-social engineering training curriculum so that the human attack surface is reduced over long-term. Dismissing the exploitation as unfeasible is a critical omission which reduces security to decision-making processes of individuals swayed by emotional impulses and ignorant about the consequences of their actions.

However, claiming the organization is secure when the suspicion-based approach and resilience to psychological manipulation is ingrained in employees would be erroneous. Complex dependencies between software and hardware, demonstrated in chapter 5.1, makes eliminating weaknesses a challenge, i.e., regardless of the measures taken to protect it, ICT infrastructure will have inherent deficiencies the adversary can reverse engineer and exploit. As depicted in Figure 23 on page 71, at a certain point the law of diminishing returns will decrease benefits to security from additional investments which organizations are not recommended to pursue because of low cost/benefit ratio. The proposed model frames and tightens the three aspects of ICT (hardware, software, people) for better alignment with organizational process structure. Security gains are a positive byproduct, not the primary objective. The model’s premise is that most systems in production environment should be viewed as insecure by design and treated accordingly; the task of management and IT personnel is to implement policy- and technical-level support for users and minimize number of avoidable intrusions via known attack vectors. DiD does not prevent breaches but reduces the perpetrator’s scope of actions when she gets in.

The concept is based on deploying multiple layers of protective measures in the network. Reinforcing perimeter defenses, often understood to be the crucial element of overall system resilience, creates a “shell” of individually insecure parts (primarily people and software) which exhibits high porosity. Also, as per the asymmetric security premise discussed in chapter 2.4.5, a single weakness is sufficient for the attacker to gain system access. Rather than relying on a single layer, DiD hardens the infrastructure close to the sensitive electronic assets and prevents data expropriation in case it is accessed. It does so by various means, e.g., diverting the adversary to a contained environment (honeypot) which contains fake data unrelated to the asset being protected, and utilizing physical barriers for critical servers (an overview of red/black separation was provided in chapter 2.4). Even though each successive layer increases overall complexity, overlapping the tools is hypothesized to also increase security.

A complementary approach is defense in breadth, an extension of DiD which prioritizes deployment of multiple tools in the same layer, e.g., several firewalls to overlap and allegedly improve security. The methodology was criticized for introducing instabilities due to incompatibilities of software designed for identical purpose. It was stated that “[c]lustering redundant technologies just does not solve the issues facing information security professionals. When properly deployed, [d]efense in [d]epth could hypothetically provide the same benefits of routing attacks as [d]efense in [d]epth; however, upon the introduction of a new attack vector this methodology is lethargic and slow to adapt as the resources required to defend new vectors are considerable” (Cleghorn, 2013, p. 3). Besides, IT personnel need to spend much more time on installation, configuration, tweaking, and maintenance, reducing system’s self-sustainability. This is true for both DiD and defense in breadth.

The question of whether human element should be involved in any of the security checks for accessing sensitive data needs to be addressed as well. In line with the recommendation about users being decoupled from security to the highest extent possible, employees should not be

relied on as a separate security layer, but solely for control and monitoring. Defense in depth necessitates competent authority prepared for operational corrections in case the technology does not provide its intended function, exhibits performance degradation, generates high false positive or false negative rates, and impedes access to sensitive electronic assets for authorized parties. Employees assigned for physical identification should be trained to strictly enforce policies of no removable media, mobile devices, wearable computers, or any other technology capable to store and retrieve data. Depending on the asset protected (internal, sensitive), computers used as connection points may be separated and therefore unreachable from the internal network, requiring physical presence monitored in real time. Another policy may include no physical copies printed at the terminal, and if so, their destruction supervised by the security personnel. Such measures hamper user comfort, though, and should only be implemented for data whose misappropriation threaten business continuity. Replication to secure storage, strong encryption scheme with tightly-controlled key distribution as well as ensuring high confidentiality, high integrity, and limited availability should be defined at the management level (ICT policy) and enforced at the technology level.

Most organizations do not require dedicated air-gapped terminals and physical security, but nevertheless possess information in need of restricted handling. Access Control Lists (ACLs) and Role-based Access Control (RBAC) are two mechanisms allowing fine-grained specification of users and groups who have been permitted access to data. These programmatic means can be custom-tailored, and are highly flexible when changes in the organizational structure occurs which may give rise to the insider threat. For example: if an employee is demoted and a possibility exists of retaliatory action, the account can be withdrawn from the RBAC list immediately which precludes the insider threat. Even though RBAC is a popular option for larger corporations, it will not be utilized because employee-wise, FaME cannot be classified as a large organization. The scaled-down version of RBAC, access control lists, delivers advantages over manual control when implemented diligently.

The following chapters detail some elements in the proposed model, namely BYOD management, internal network segmentation, incident response together with ICT infrastructure hardening, and password composition requirements. Logical and physical data separation aims to eliminate single points of failure by distributing data across physical (servers) and logical (VMs) instances. It deals with load balancing and database-replication techniques (chapter 2.2.3 described selected paradigms). Patch management and deployment is a formalized approach whose objective is to reduce windows of opportunity for the adversary (chapter 2.4 contains schematic depiction and explanation) when software updates are released which address exploitable vulnerabilities, increase stability, and harden the software by closing known attack vectors. Patches may introduces instabilities and backward compatibility issues: a theoretical scenario of infrastructure consisting of five tools was analyzed in chapter 5.1, but in large production environments the amount of interacting parts may reach dozens. Automated insider threat and detection is primarily concerned with anomalies in packet-based communication on the internal network which may hint at a presence of malicious insider. If DiD or defense in breadth is implemented, proper configuration, thresholds, and non-standard actions can be simulated and tested without user intervention before being integrated into the existing security model. It is the author's opinion the aspects provide a representative cross-section of issues the industries currently face most often.

6.2.1 BYOD Management

Software and hardware, both complex systems whose high-level overview was presented in chapter 2.3, form an interconnected structure which brings about new challenges to sensitive

electronic asset security. Before the advent of smartphones, majority of mobile subscriber data were self-contained on physical tokens.

Feature phones include internal memory chips ranging from several kilobytes (kB) to several megabytes (MB). Smartphones are equipped with internal memory sized in gigabytes (GB), further extendable to dozens of GB. Contact lists and SMS are stored on the device or remotely in the data center (cloud) with sufficient space overhead. Because mobile devices can be attached to computers and data bidirectionally transferred, users may use them in place of removable disks, turning them into a security risk should the device be misappropriated. Increasing the work factor to obtain actionable intelligence must be balanced with ease of use: for example, products running iOS, a popular mobile OS developed and maintained by Apple, allows to turn on “complex” passcode. When an attempt is made to unlock the device, a code needs to be entered which may include numbers, capital and lowercase letters, and special characters instead of only numbers offered by default. Coupled with a data protection feature ensuring the data is wiped after 10 failed tries, the adversary cannot employ brute-force nor dictionary attacks and has to look for alternative ways to access the memory. Moreover, user can set restrictions, protected by another passcode, which selectively disable functions such as wireless networking. The device works as expected otherwise which makes restrictions highly recommended in BYOD management.

Users should be encouraged to separate work-related and personal contents on their devices using software a profile, a collection of permissions and best practices bundled into a package distributed to compatible devices. Organizations can either assign users phones with mobile device management tools already put in place, or install profiles to employee-owned smartphones from a central repository using remote programming; future changes to the configurations are handled over secure channels. Thorough testing is necessary to prevent personal data loss, hardware and software instabilities, and misconfigurations. Policy on when to activate the profile must be clearly communicated: users must be allowed to know the extent of privileges along with limitations since they own hardware on which it will execute. Mobile device management uniformly diffuses explicitly-set policies and ensures large-scale consistency. Profiles can incorporate periodic checks for patches, timely software updates delivery, battery-saving measures for Wi-Fi and mobile data networks, etc. Tying configurations to ACLs is reasonable as it grants programmatic permissions with respect to user position and level of privileges in the organizational structure. Per-user profiles are another option redundant for flat hierarchies.

The sole BYOD management tool for iOS devices is Apple Configurator by Apple. Compatible with Active Directory, the management console can synchronize and assign accounts to specific profiles, enforce patch management policies by decoupling users from the decision when to deploy OS updates, and install applications from the official App Store. A secure tunnel can be enforced for every service which interacts with the internal corporate network, encrypting data in transit to prevent eavesdropping. For Android, many competing commercial solutions are available: Google offers Device Policy for Android as part of the Google Apps suite for organizations, but the OS’s open-source nature spawned services which offer unified mobile device management capabilities supporting multiple system versions. Blackberry offers Blackberry Enterprise Server, a system-agnostic platform for deploying profiles on devices running Android, Blackberry, and iOS. The proposed model does not incline toward a particular vendor and only requires the devices to be configurable with BYOD profiles.

At the time of writing, Android suffered from version erosion. Unlike the Apple ecosystem hardware and software of which is tightly controlled and device classes cut off from future updates without prior notification, Android’s update model is atomized among multiple vendors which results in substantial delays between source code release and the OS patches available on

the official release channel. This was particularly noticeable with version 2.3 released in 2010: massively popular due to its low hardware requirements and stability, the OS was customized by manufacturers and deployed to a wide range of entry-level smartphones. This created a situation which saw users satisfied with the functions, postponing upgrades to newer models. Android 2.3 lacks critical security features introduced in later iterations, opening multiple attack vectors to internal networks if devices running it are used for interacting with sensitive electronic assets. Gradual hardware replacement is expected to take place with discontinued support and moral obsolescence, but IT personnel should protect any device used for checking emails and remotely connecting to information systems, two actions taking place in every organization on a daily basis.

Wireless-enabled devices should be set with security as a primary objective, and more so with those which access sensitive electronic assets. Any temporary data left on the device should be periodically purged and scraped from caches. Smith (1982, p. 1) defines caches as “. . . small, high-speed buffer memories used in modern computer systems to hold temporarily those portions of the contents of main memory which are (believed to be) currently in use. Information located in cache memory can be accessed in much less time than that located in main memory. . . .” Caches proliferated with decreasing storage costs as an efficient way to retain data (login credentials, passwords, web page contents) for repeated use. While increasing user comfort, cache memories must be assumed accessible to third parties and their contents considered sensitive. Users should be aware of the risks associated with caches and advised to wipe them after each session. Even though it impacts convenience, the policy hampers any attempt at surreptitious use of the data contained therein. Some mobile OS utilities prompts for a permission to temporarily save data; users are strongly discouraged from doing so for wireless networks because they are prone to active and passive interception and must be assumed under attacker’s control.

To be effective, BYOD best practices and policies must be amended to reflect novel developments in ICT and security. Clear best practices free of technical details which instead communicate main points on why the particular measure was implemented are preferable by users. They may also serve as an accessible explanation for BYOD profiles including implementation, restrictions, and settings. It is recommended to treat best practices as internal document and protect it accordingly. Profiles themselves should be distributed solely through the management console because reverse engineering their content could give the adversary insight into the security features implemented on mobile devices in the organization.

As a last note, BYOD management administers a set of common rules across heterogeneous hardware base. For scalability reasons, per-device policies should be of secondary importance to per-OS profiles. This approach necessitates finding the lowest common denominator in terms of features supported. Even though the proposed model explicitly acknowledges mobile devices as a part of organizational ICT infrastructure, procurement is a viable alternative which solves the challenge of unifying disparate user-owned hardware and software. Purchasing smartphones and distributing them to employees already loaded with profiles help separate personal and work-related data by explicitly prohibiting the former to be stored at any point. Choosing a single vendor also enables tailoring and managing profiles uniformly, and patch deployment policies can be synchronized to keep the mobile OS updated at all times without compatibility issues. The disadvantage is reduced user comfort due to several devices (personal, work).

6.2.2 Internal Network Segmentation

Network-protective measures such as IDS (chapter 2.1.5) and firewall must be in place, tested, and functional. The concept of firewall was described as follows: “A more flexible way to

control access is to prevent certain packets from entering or leaving an organization through its gateways. This allows greater flexibility than an application-level gateway, although as with any power tool it also requires greater vigilance” Mogul (1989, p. 1). Application-level interface whitelists certain applications and disallow the rest to send or receive traffic from the Internet. Network communication uses packets which can be identified, classified, analyzed, and segmented according to their properties set in standards for Transmission Control Protocol (TCP) by Information Sciences Institute (1981), and User Datagram Protocol (UDP) by Postel (1980). Both are firewall-filterable to a degree, although attackers must be assumed able to craft malicious packets capable to pass through poorly-configured defenses. Application firewalls should thus be preferred as they analyze aggregated streams instead of individual packets.

Case study 2 (chapter 5.2.3) included Nmap network scanning tool which offers several techniques to gauge responses from the firewall and fingerprint it based on its behavior. Exploiting situations not covered in the standards which permit responses based on what the designers considered the best course of action, Nmap polls internal database and estimates the vendor despite the host configured being not to leak software versions as recommended in chapter 5.2.4. The program sends malformed, non-standard packets to the target port and awaits answers. Even disabling replies and disregarding the incoming data units can be collated with results of other tests, and probability a particular firewall is present assigned. This makes designing an Internet-facing system leaking no information whatsoever comparable “perfect” security: additional resources produce diminishing benefits. The ICT governance model assumes the adversary can gain intelligence about both internal and external networks such as (information in brackets suggests a tool or service):

- general overview of ICT infrastructure (search engines, official reports, website analysis),
- email addresses and spam filters in use (TheHarvester, email headers),
- LAMP stack components and versions (Netcraft, Nmap, Nessus),
- firewall and IDS versions (Nmap, Metasploit),
- operating systems (Nmap, Metasploit),
- services communicating on ports (Nmap, Nessus, Metasploit),
- virtualization platform specifics (Nmap),
- wireless connectivity details (physical premise inspection).

Apart from physical premise inspection, the interactions with the target system can be tailored to stay under IDS or firewall thresholds and it is realistic to consider them as stealthy. On Internet-facing IP addresses, the perpetrator can stay below the detection level by relying on the background traffic, “. . . either destined for addresses that do not exist, servers that are not running, or servers that do not want to receive the traffic. It can be a hostile reconnaissance scan, ‘backscatter’ from a flooding attack victimizing someone else, spam, or an exploit attempt” (Pang, Yegneswaran, Barford, Paxson, & Peterson, 2004, p. 1). If the target ports are randomized and the requests do not exhibit a predictable time pattern, the traffic is unrecognizable from the background noise and will not be picked up by any filter. IT personnel should therefore forgo threshold tweaking for attack precursor detection, and instead focus on hardening the infrastructure, reducing system porosity, and devising long-term employee education plans.

Organizations should consider establishing ubiquitous Wireless Local Area Network (WLAN) connectivity as smartphones are not equipped for wired data transmissions. To ensure electronic asset safety, network security “. . . needs to guard networked computer systems and protect electronic data that is either stored in networked computers or transmitted in the network” (J. Wang, 2009, p. 1). Andress (2011, p. 115) points out that “[a]s network dependent as the majority of the world is, loss of network connectivity, and loss of the services that such network provide,

can be... potentially devastating to businesses.” Such losses occur as a result of targeted attacks, misconfigurations, and outages (some attacks were discussed in chapter 2.4.4). Misconfigurations occur when hardware or software are subjected to conditions outside of expected parameters by deliberate attempts or through circumstantial combination of factors. Software is prone to bugs and it is a best practice to conduct controlled tests prior to production environment deployment. Business continuity requires “. . . functionality and performance under unusual or anomalous conditions. . . [when] a program receives excessive, or insufficient inputs. . . A common approach to assuring such continued operations of large, networked software systems is penetration testing” (Underbrink, Potter, Jaenisch, & Reifer, 2012, p. 1). Chapter 2.4.8 encompassed penetration testing methodology in more depth.

WLAN must be separate from LAN and policies reflect the differences. By enabling wireless connectivity, system operators cannot enforce the same rules because the transmission medium is not a shielded physical cable but a radio signal. For cables, data can be intercepted either physically or programatically, both of which are noisy. Conversely, wireless data interception is trivial to accomplish because packets move in an uncontrolled space, depicted in Figure 117. All hosts connected to the access point (AP) receive messages intended for a single recipient, but are instructed to disregard packets whose destination identifiers do not match the one assigned to them in manufacturing. A malicious party can modify the behavior in software, turning her machine into a siphon for incoming traffic. The result is a complete overview of packets on the selected AP, with unencrypted streams posing severe risks for the CIA triad’s elements (chapter 2.2) if organizational information systems are accessed over Wi-Fi. The proposed model builds on an encryption scheme which makes real-time decryption unfeasible for data inside and outside the internal network. The perpetrator can also resort to active interception where the packets are changed as they pass through her node, altering their contents arbitrarily. The man-in-the-middle attack was mentioned in chapter 2.4.4 and is undetectable when no encryption is implemented unless advanced monitoring tools are run. This scenario is outside the realm of possibility for most users who seek an easy way to access the Internet, and strongly prioritize convenience over security. A BYOD profile can enforce an encrypted channel to be established when an attempt to connect over Wi-Fi is encountered.

Design considerations should be taken into account as well. Wireless network are deployed by creating an AP mesh, positioning each device on elevated surface for direct line of sight, which limits environmental signal absorption and decreases attenuation. Care should be exercised to ensure the signal does not leak into uncontrolled space, depicted in Figure 117. If such situation occurs and strong encryption scheme has not been integrated or is misconfigured, the network could be remotely compromised even without physical access.

Another measure is segmenting WLAN into layers. This way, every employee will have access to the network but sensitive data will be available on a per-user basis. The arrangement improves security by containing one segment in case of compromise while leaving others unaffected. The attack surface is expanded, though, and threats for k unique networks increase exposure risk k times. Configuring a suitable encryption scheme should be assigned the highest priority, preferably Wi-Fi Protected Access II (WPA2) complemented by strong authentication protocol because vanilla WPA2 and its previous iterations have been found susceptible to attacks under realistic assumptions (chapter 2.4.4). Computational overhead is negligible on devices supporting multi-threaded applications, and the network must be assumed continuously scanned for exploitable vulnerabilities by third parties. Furthermore, a policy of strong passwords must be enforced which prevents brute-force and dictionary attacks. Two ways of distributing Wi-Fi passwords are viable: each user can select their own, or a single string is used for authentication onto the WLAN. Whatever approach is preferred, Wi-Fi should not be left unprotected. Numerous

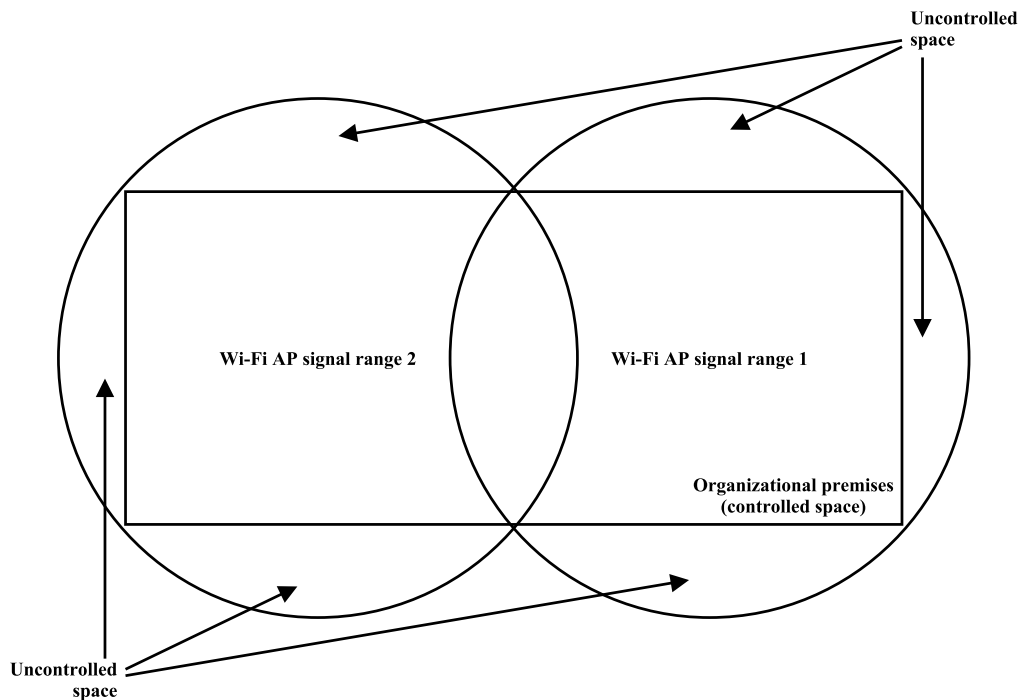


Fig. 117: *Wi-Fi signal leaks. Uncontrolled space is freely accessible to unauthenticated third parties and poses a security risk, especially in areas where it cannot be monitored.*
 Source: own work.

attack vectors for “open” wireless networks exist and can be launched using free tools. The following precautions must therefore be implemented for a baseline Wi-Fi security:

- WPA2,
- authentication server,
- strong password policy,
- separate subnetwork.

IT personnel should review the network topology and decide whether segregating mission-critical systems onto subnetworks will increase resilience and harden the infrastructure by introducing the ability to manage separate networks and contain threats within isolated segments. Sample topology is schematically depicted in Figure 118.

Individual subnetworks may contain a single VM instance running the service (information system), or a range of devices (user hosts, VoIP telephony, WLAN). Segmentation enables proactive incident response, simplifies real-time logging, and provides fewer access points into the network because packet flows in the subnetwork can be predicted and modeled with high probability which, if translated into IDS and firewall rules, result in an effective anomaly-detection system. Also, because the segments are established virtually in software on the same set of hardware, the threat domain spans only hosts in the current subnetwork, not the entire LAN. This is especially useful for Wi-Fi with multiple uncontrolled interconnects, off-premise signal leaks, and transient nature of clients: contrary to the user hosts subnetwork where the number of nodes is relatively stable, Wi-Fi associations are ephemeral, i.e., a device may be seen only once. Coupled with unprotected wireless networks and lacking authentication features, the malicious party is free to exploit others almost anonymously. A convenient one-click Internet access offered to visitors and other third parties should reside on a subnetwork strictly detached from internal resources. While Wi-Fi for employees and other trusted entities can be collocated on the user hosts segment, it is strongly recommended to separate the two.

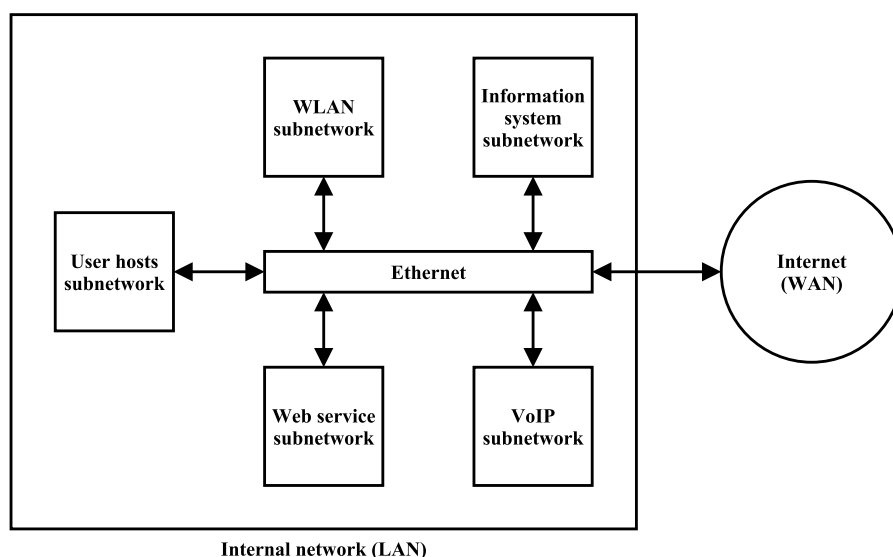


Fig. 118: Sample network stratification topology. Hosts in each subnetwork are isolated from others which results in smaller threat domains.
Source: own work.

6.2.3 Incident Response, ICT Infrastructure Hardening

Incident response, formally computer security incident management, denotes a sequence of steps executed manually or automatically when an anomaly is registered on a monitored interface. The steps should be formalized and tested for validity in simulated and real-world scenarios because the feedback ensures incidents are contained before their potential to compromise any of the sensitive electronic assets' CIA is realized. As an example, Figure 119 depicts a simple decision tree for a scenario where non-standard network activity was detected on a user machine.

Several responses are available ranging from global (block all traffic from the offending port) to targeted (inspect the packets and establish forensic trail for analysis). When the activity is detected, the very first step should be to temporarily disallow the connection until its origin is verified. While perhaps overly strict, the approach designates any non-standard activity as potentially malicious and takes preemptive steps. The second step polls several internal or external databases for known services operating on the port to reduce false negatives and false positives. If a true positive match is made, the connection is dropped or redirected into a controlled environment for analysis. Further, the originating device is isolated from the rest of the network in case another egress data transfer is attempted. Even if a true negative is established and the service appears benign with high probability, the packets should nevertheless be inspected. A series of checks such as presence of encrypted traffic and keywords, are employed. Encrypted traffic assumes the source employs it to thwart detection, keywords scan for terms (login, password, etc.) which may indicate an attempt to expropriate sensitive personal or corporate data. While heuristic in nature, the techniques are conservative and prefer terminating the stream than risk security incident. Even if neither check raises any warnings, a log entry should at least be generated and copy of the traffic stream retained for a predetermined period.

The proposed model does not recommend allowing data transfers unrelated to core business interests. Peer-to-peer clients in particular may piggyback large volumes of uncontrolled data into the internal network from potentially malicious users. Moreover, the software probes different ports which may trigger the decision logic for each of them separately which saturates CPU cycles, and retaining the stream also taxes finite server storage capacities. It is recommended

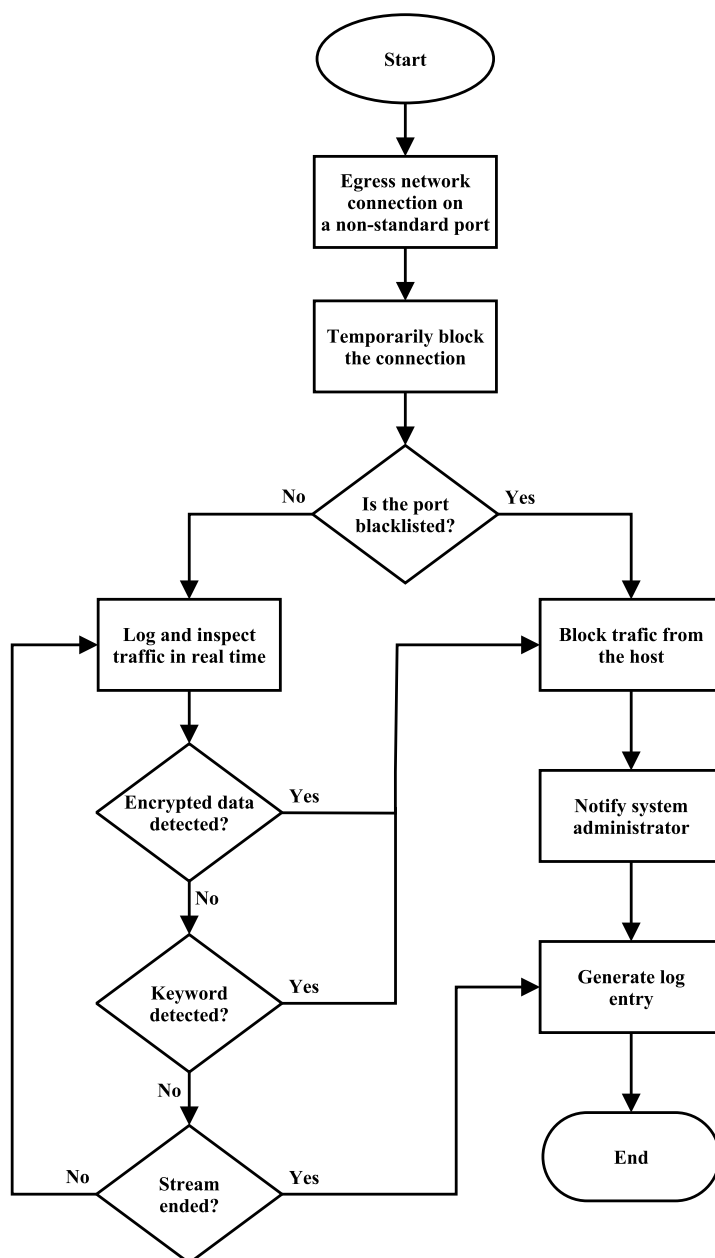


Fig. 119: Incident response. The logic can be programmed into a firewall rule and automated. When any of the incident conditions is met, system operators are notified while countermeasures are deployed. Source: Own work.

to forbid such traffic explicitly in the ICT policy, and establish incident response in case the directive is violated.

The example demonstrated a single anomaly, but in production environment, unique situations which necessitate human intervention and immediate actions are commonplace. Instead of formalizing procedures for each, IT personnel should agree on a set of core principles governing their conduct when an incident takes place. The fundamental supposition the network is inherently insecure from external and internal agents is incorporated in the model, and was discussed previously. Maintaining asset confidentiality, integrity, and availability should be another priority: overly conservative approach is warranted and if balance cannot be maintained, decreasing false negatives when identifying threats is preferable at the cost of increasing false positives. For instance: classifying an egress connection attempt as malicious and blocking it at the source

is acceptable despite it being innocuous compared with no action and possibly allowing data expropriation to take place. Users should be encouraged to contact IT personnel when in doubts about the consequences of their actions. However, some users may launch software indiscriminately: communicating the risks and instilling a sense of responsibility by stressing each employee affects stability of the system can alleviate some challenges with managing such individuals. If no change is observed after multiple warnings, corrective measures (process blacklisting, lowering privileges) need to be used for threat mitigation.

While organizations face many technological challenges, one pertaining to both incident response and ICT infrastructure hardening is a backup solution in case of network outage. Every organization depends on an ISP (Internet Service Provider), a business entity delivering hardware (cables, devices) and services (electronic mail, IP telephony) pertaining to bidirectional data packet exchange in a timely fashion, specified in an SLA (chapter 2.2.3). Scheduled and unscheduled downtimes can render the client unable to interact with supply chain partners, without access to corporate data if they are stored in a remote location (cloud), and stall productivity if thin desktop clients are used. Diversifying risk by negotiating backup connectivity with an alternative vendor guarantees redundancy for mission-critical applications, and supports business continuity management (BCM). ISO (2012a) defines it as “... a holistic management process that identifies potential threats to an organization and the impacts to business operations those threats, if realized, might cause, and which provides a framework for building organizational resilience with the capability of an effective response that safeguards the interests of its key stakeholders, reputation, brand and value-creating activities.” BCM is closely associated with the Plan–Do–Check–Act (PDCA) model whose schematic representation is depicted in Figure 120.

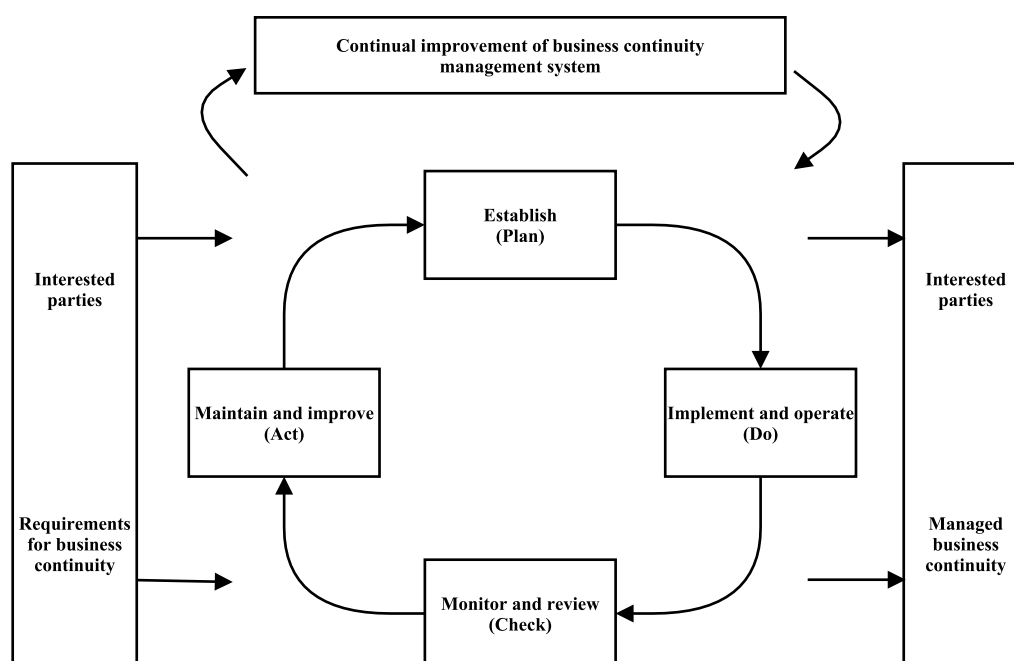


Fig. 120: Plan–Do–Check–Act model. The process takes two inputs: information from interested parties (stakeholders) and business continuity requirements. Stakeholders should be notified of any developments in business continuity assurance plans as well as be allowed to check the plans for feedback. Source: ISO (2012a), modified.

Partial or complete disruption of ICT services is an imminent threat to business continuity and the PDCA methodology is a way to mitigate it through continuous improvement. Industries whose operations require uninterrupted Internet access, high availability, and stable network parameters must factor in the possibility the connection will be disrupted due to blackouts,

corrective or preventive maintenance, hardware or software failures, natural disasters, targeted attacks, and unanticipated causes. If no failover solution is devised and seamlessly transitioned to, loss of network connectivity will occur. In case the switch to redundant channels is instantaneous, mission-critical systems will register a short-lasting spike in traffic but in case no backup is deployed, active transfers will be interrupted. Consequences range from revenue loss, decrease in employee productivity, paralysis of Internet-dependent services (cloud computing, VoIP telephony), and costs to establish alternative connectivity. The threat is pronounced in organizations which rely on off-site data centers for VDI, backups, and activities contingent on Internet access. A simplified PDCA sequence for the redundant connectivity is:

- Plan (Establish): map competitive landscape for suitable vendors based on predefined set of criteria: availability, time to backup connection transfer, support level, network data transfer rates, shared/dedicated channels, uptime guarantees, etc.; establish legally-binding business relationship with the ISP by means of an SLA,
- Do (Implement and operate): deploy infrastructure; initiate controlled network outage to test procedures for switching to backup connectivity,
- Check (Monitor and review): evaluate results and identify bottlenecks which affect business continuity, e.g., unacceptable response time duration, misconfigurations, high network latency, etc.; analyze alternatives (mobile broadband modems, satellite Internet access),
- Act (Maintain and improve): negotiate corrective actions with the ISP to improve network resiliency; establish an out-of-band channel such as mobile phone service with the provider.

Redundant topologies, i.e., reduction in network downtime by provisioning the system with additional resources which eliminate single points of failure and enhance reliability, necessitate a secondary channel separate from the one for which the topology is designed. For example: two different ISPs with independent sets of hardware and cabling used for packet routing are necessary to reduce risk of component failure affecting the whole network. While the deployment cost needs to be taken into consideration as well, critical infrastructure provisioning scheme must result in a usable system: the goal is a failover which minimizes data losses bound to occur during connectivity loss. Advantages of additional Internet uplinks include load balancing which distributes traffic across channels, reduced latency, and congestion control. Integrating the backup links to the ICT infrastructure is strongly recommended for testing purposes and minimum transition delays: when one ingress/egress point is disabled, the load balancing algorithm redirects traffic immediately. Purchasing redundant connectivity solely for BCM reasons without utilizing it is not economically justifiable. Website-replicating services that cache resources even if they are unavailable at the source may provide a means to maintain Internet presence despite network outages.

Connectivity loss is one point in ICT BCM considerations, and was included for demonstration purposes. Along with power outages, briefly discussed next, they should be considered priorities in physical infrastructure hardening. A PDCA sequence detailing the decision-making process when planning for power cuts is as follows:

- Plan (Establish): identify mission-critical hardware for which auxiliary power sources must be installed, and outline their specifications; determine transient data to be saved when transferring the system into a power-off ready state,
- Do (Implement and operate): purchase and deploy backup power sources; initiate controlled power outage during off-hours to test failover switching,
- Check (Monitor and review): evaluate results and determine bottlenecks: data transfer times, comparing power supply performance to specifications, transition period to auxiliary sources, and user reactions to the outage,

- Act (Maintain and improve): implement corrective actions; designate more systems redundant to allow more time for moving critical data, observe best practices for auxiliary power sources, periodically test infrastructure readiness, inform users of impending tests, and suggest course of action during uncontrolled outages.

Blackouts can paralyze the organization if uninterruptible power supplies (UPS) are under-specified and left untested, leading to faulty incident response. The proposed model assumes redundant components and services are thoroughly tested before being deployed to production environment.

To access sensitive internal data outside the corporate-controlled premises, at least one precaution should be implemented: virtual private network (VPN). Packet-switched digital communication on the Internet relies on the Transport Control Protocol/Internet Protocol (TCP/IP) suite designed without security considerations, thus constituting an unreliable medium as described in chapter 2.2.2. Widespread use of TCP/IP makes any backward-incompatible changes unsuitable for large-scale implementation. Many programs pose significant threats when connected to the Internet, "...[t]herefore it is essential that applications are kept up-to-date by applying patches or service packs that address new exploitable vulnerabilities. Other problems are caused by uneducated users or shortcomings in the [organisation's] security policy" (B. Harris & R. Hunt, 1999, p. 13). Cryptographic protocols have been devised but found to contain serious vulnerabilities. Virtual private network is defined as "... a generic term used to describe a communication network that uses any combination of technologies to secure a connection tunnelled through an otherwise unsecured or untrusted network. Instead of using a dedicated connection, such a leased line, a 'virtual' connection is made between geographically dispersed users and network over a shared or public network, like the Internet" (HKSAR, 2008, p. 4). The technology establishes a dedicated encrypted tunnel through which data is passed between remote connection endpoints, i.e., a server inside internal corporate network or data center, and a host. Many options and alternatives exist which should be scrutinized as to their security, ease of use, processing overhead, supported authentication, and data encryption schemes. VPN, along with many other identity-based services, rely on authentication methods using the following factors, each of which increases the work factor for the adversary:

- knowledge: challenge–response, one-time password (OTP), password, pass phrase, Personal Identification Number (PIN),
- possession: digital certificate, biometric card, smartphone, Trusted Platform Module (TPM), security token,
- inherence: deoxyribonucleic acid (DNA) analysis, fingerprint and retinal scan, signature, voice recognition, gait style.

When compromised, possession factors can be obsoleted which incurs replacement costs ranging from CPU cycles for generating new digital certificates, issuing biometric cards as well as smartphone, and security tokens procurement. Trusted Platform Modules cannot be easily switched as each is unique for the device on which it is placed, necessitating replacement of the underlying hardware. Inherence factors are impossible to revoke due to their innateness in every individual. Therefore, the proposed ICT model strongly discourages organizations from using this form of authentication unless biometric tokens are implemented as part of the authentication process and can be immediately revoked. If biometric identification is the sole means of proving one's identity, obtaining the electronic data (iris and fingerprint scans, DNA and voice samples, signatures) means permanent feature compromise. IT personnel should consider such assets mission critical, and deploy the same security measures as for the information systems processing sensitive-class business data. An overview of asset classes was provided in chapter 6.1.

Biometrics is susceptible to false positives and false negatives as well as feature interception. For example: a fingerprint impression from a suitable surface and partial retinal image from a high-resolution photograph are viable vectors of approach. Unless additional factor is employed in combination with biometric authentication, perpetrators will focus on reverse engineering the scheme by exploiting human element, the weakest link in the ICT security infrastructure context. As employees are the sole source of biometric data, security is tightly coupled with how vigilant and resilient to adversarial attempts people are in real-world situations. Such approach presents the inverse of the ICT model which claims security and human factor should be decoupled to the highest extent possible. By making security exclusively dependent on an unreliable agent, technological countermeasures will be rendered ineffective.

In the authentication process, tokens must not in any way identify the owner, be trivially and immediately replaceable with minimum (zero) cost, easily manageable to ensure user comfort, be platform agnostic, usable without significant modifications, and a compromised token must not in any way weaken the algorithm used to generate it. A password supplied with possession two-factor artifact (mobile phone) is recommended in the proposed ICT security governance model. Password composition requirements will be discussed in the following chapter, BYOD profiles (chapter 6.2.1) allow to deploy specific applications on the device: applications which present the user with a sequence to be entered along with the password are available as enterprise-grade solutions from multiple vendors. Physical tokens can be also supplied.

Hardening workstations is a broad topic but two recommendations based on observations from the reconnaissance and information gathering phase of case study 2 (chapter 5.2.2) are lowering privileges and installing software designed to give system operators options to tighten application security. Lowering privileges from superuser to user should constitute a baseline feature where employees are not expected or permitted additional software. Microsoft Windows requires administrative privileges for software updates, and disallowing administrative privileges relegates users to patch management policy enforced by the IT personnel. Microsoft traditionally releases security fixes for products enrolled in Microsoft Update on the second Tuesday of every month, and non-security updates on the fourth Tuesday of every month. Other vendors either stream updates continuously, or in discrete intervals. Even though users should have a choice of software, particularly browser, once selected, it should be kept up to date automatically so that no further action on part of user is required. Installing OS patches can be non-intrusively implemented at the end of current session when the user indicated their willingness to turn off the device rather than being forced into restart by means of pop-up windows and system-imposed time limits.

IT personnel may also want to subject hosts to blacklisting or whitelisting sites (social networking, online games, keyword-specified objectionable content) and applications. Both protect sensitive assets based on different assumptions: blacklisting "... focuses on matching specific aspects of application code and particular actions being attempted by applications (behavior monitoring or heuristics) for detection," (Shackleford, 2009, p. 2) whitelisting "... refers to software that resides on a host computer and maintains a logical 'fingerprint' of applications that are allowed based on policies tied to users, groups, systems, and other potential attributes... then it denies everything else" (Shackleford, 2009, p. 3). In the OS environment, whitelisting employs the "guilty until proven innocent" principle: "... all software presented to an operating system for execution is presumed to be malicious unless known to be non-malicious. Users are disallowed from executing any software that is not listed in a 'safe code registry'" (Harrison, 2005, p. 3). Blacklisting is optimistic and permissive, assuming applications outside the disallowed pool are harmless, whitelisting is pessimistic and restrictive, applying the "deny all" policy except for a subset of applications. Blacklists are favored in firewalls and antivirus software, whitelists in

mission-critical systems. Townsend (2011) admits that “[w]hitelisting is fundamentally the better security solution. . . . Against this, the administrative effort involved in blacklisting is minimal compared to whitelisting; and the difference increases as the size of the whitelist increases. However, the efficiency of blacklisting decreases as its size increases.” Schneier (2011) believes “. . . the whitelist model will continue to make inroads into our general purpose computers.” Access control lists are a form of whitelisting which permit access to assets if and only if the authentication procedure is finalized successfully. A combination of blacklisting and whitelisting can improve the defense in depth model which seeks to delay the attacker after they penetrated the system. It was schematically depicted in Figure 116 and discussed in chapter 6.2.

Recommended software for Microsoft Windows OS includes the Enhanced Mitigation Experience Toolkit (EMET) and TrueCrypt along with antivirus, firewall, and on-demand antimalware scanning tools. While the last three mentioned constitute baseline security configuration, EMET and TrueCrypt are not yet commonplace in production environments despite being regularly used in highly-competitive industries. The former is a collection of application-agnostic technologies designed to thwart most prevalent electronic threats by enforcing appropriate policies. The advantage stems is that even programs developed without security in mind can benefit from EMET. The tool can be deployed in an automated fashion and turned on immediately. The proposed ICT model considers end-node susceptibility to malicious attacks, especially when running Microsoft Windows, a serious threat which could propagate and endanger ICT infrastructure, however resilient, by exploiting internal hosts. Coupled with diligent implementation of the DiD concept, any workstation connected to the Internet and running a web browser should have an instance of EMET deployed.

TrueCrypt is a full-disk encryption utility. Supporting several cryptographic algorithms separately or in a cascade, a new partition holding the sensitive data is created and managed by the application. The volume is protected by a password or a compound of a password and a keyfile, a two-factor authentication which proves both exclusive knowledge and possession. While technical details are outside the scope of the thesis, TrueCrypt should be used for backups of password databases when software such as KeePass, mentioned in chapter 6.2.4, is used. In the proposed model, IT personnel must install TrueCrypt if data misappropriation is a concern (health care, financial and banking sector, military and legal applications), or where users explicitly request secure storage container. It should be noted, though, that gaining access to the encrypted partition by means other than inputting the password, supplying the key file, or using a Rescue Disk (a file built during the setup process) along with the password is not possible to the best of author’s knowledge. A malware-infected host capturing key strokes or device expropriated with TrueCrypt open at the time are two cases where the adversary can gain leverage into the encrypted drive. The former is mitigated by implementing best practices, keeping the security features updated, and employing suspicion-based behavior in electronic and personal interactions with unknown third parties. For expropriating a device with TrueCrypt open, a set of techniques was developed which “. . . pose a particular threat to laptop users who rely on disk encryption products, since an adversary who steals a laptop while an encrypted disk is mounted could employ [the] attacks to access the contents, even if the computer is screen-locked or suspended” (Halderman et al., 2008, p. 1). This scenario is an edge case and should be treated as such by focusing primarily on closing malware infection vectors.

Lastly, a short note pertaining to a choice between Microsoft Windows and alternative systems, particularly Linux-based, in terms of security is warranted. Several vendors offer commercial deployment and support for Linux, addressing the disadvantage of community-supported distributions which lack accountability, security, and stability assurances. Convergence of GUI appearance should lead to a short learning curve when users migrate from Microsoft

Windows to Linux, although the former tends heavily toward GUI and touch interfaces while Linux offers broad customization. The ICT governance model is system-agnostic, although many security recommendations discussed throughout are available in some versions of Linux. If Linux deployment is considered, usability field testing on a subset of users should take place prior to full-scale migration.

6.2.4 Password Composition Requirements

Password composition requirements are the most challenging aspect of ICT-side security as should encompass both desktop and touch-enabled devices where typing efficiency is hampered by long, random-looking sequences. Because most smartphones have limited screen estate, users need to switch between keyboard types (letters, numbers, special characters) in both directions. Therefore, a password consisting of three lowercase letters followed by a number, a special character, and an uppercase letter, e.g., pas4|W, requires:

- typing the three lowercase letters, switch to numerical keyboard (1 extra tap),
- typing the number, switch to special characters keyboard (2 extra taps),
- typing the special character, switch to alphabetic keyboard (3 extra taps),
- turning on the Shift key (4 extra taps) and typing the uppercase letter.

Longer strings would necessitate multiple extra taps to produce the desired result which significantly reduces comfort and strains patience if the string is rejected due to mistyping or omission. Modeling password input is possible using a four-state Markov chain, briefly described in chapter 5.1.2 and depicted in Figure 121. Each state denotes a keyboard which the user needs to activate in order to type the character at the current position.

The system is always in one of two states: either the current character belongs in the same set (lowercase–lowercase, uppercase–uppercase, number–number, symbol–symbol), or a state change needs to occur with the following properties: if alphabetic set is active and the next position in the password is occupied by a symbol, the keyboard must be first switched to numbers (p_{12}) and only then to special characters (p_{23}), a two-state transition $p_{12} + p_{23}$. Another two-state change occurs when numeric or special characters keyboard is active and an uppercase letter is required: the sums are $p_{21} + p_{41}$ and $p_{31} + p_{14}$, respectively. If either numeric or special characters keyboard is active and the next position is a letter or a symbol, a single-state change is initiated, denoted by p_{21} (number to letter), p_{23} (number to symbol), p_{31} (symbol to letter), or p_{32} (symbol to number). The proportions of state changes, i.e., character 2-tuples differing in adjacent positions, should be approximately equal in the string. A series of same-state probabilities ($p_{11}, p_{22}, p_{33}, p_{44}$) means the password contains multiple characters from the same alphabet set. Space is treated as a letter because it is available on any keyboard type. While not a vulnerability, if the characters form a predictable pattern (repetition, proximity on the keyboard), the adversary may resort to enumeration of common sequences (dictionary attack) with high probability of success. The Markov chain makes it possible to devise an algorithm which transforms any string into a sequence of probabilities. Suppose 4knQjE]M5w is picked; the chain is as follows:

$$p_{12}, p_{21}, p_{11}, p_{14}, p_{41}, p_{14}, (p_{41} + p_{12} + p_{23}), (p_{31} + p_{14}), (p_{41} + p_{12}), p_{21} \quad (6.2.1)$$

Even though $p_{41} + p_{12} + p_{23}$ may look like a three-state change, smartphones do not require any action on part of users to transit from uppercase to lowercase keyboard (p_{41}), making

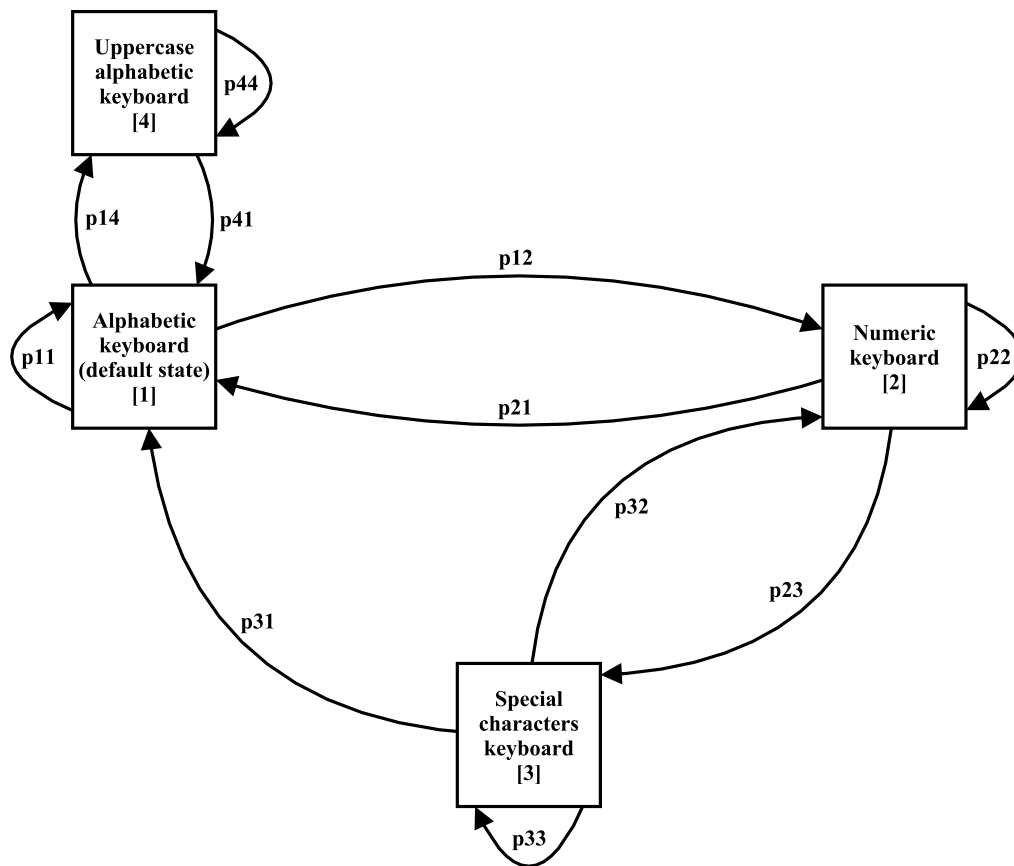


Fig. 121: Markov chain for mobile password input process. Each state represents a keyboard where the respective character is located.
Source: own work.

the change simply a matter of switching to numeric set. The Markov chain includes p_{41} for completeness. ICT policy can use the approach when programming BYOD password requirements to balance ergonomics with security by minimizing state changes while keeping the string unpredictable. Entropy, described below, is a suitable quantifying criterion for this property. On workstations, such efforts can be safely disregarded with the only user comfort consideration being the need to switch keyboard layouts in case the character is unavailable in the currently-active set. Even though Markov chains will not be discussed further in the thesis, they are a viable future research venture which integrate ergonomics, GUI design, human-computer interaction, and security. Analyzing state changes via software tracking, e.g., a program which measures the period it takes until the desired keyboard set is reached, can help quantify how long users spend on these tasks during a login attempt. The results may help reduce the “non-productive” parts of the input process while maintaining the same level of security, and provide accurate data for enterprise password manager deployment (chapter 6.1).

Mobile password composition requirements may permit users to select length and alphabet size (full, reduced) from which the string will be sampled. However, the algorithm should prohibit creation of reduced-set passwords shorter than the specified number of characters. Overall, the following scenarios should be prevented:

- string shorter than X regardless of alphabet size,
- string shorter than Y using the reduced set,
- string shorter than Z using the full set.

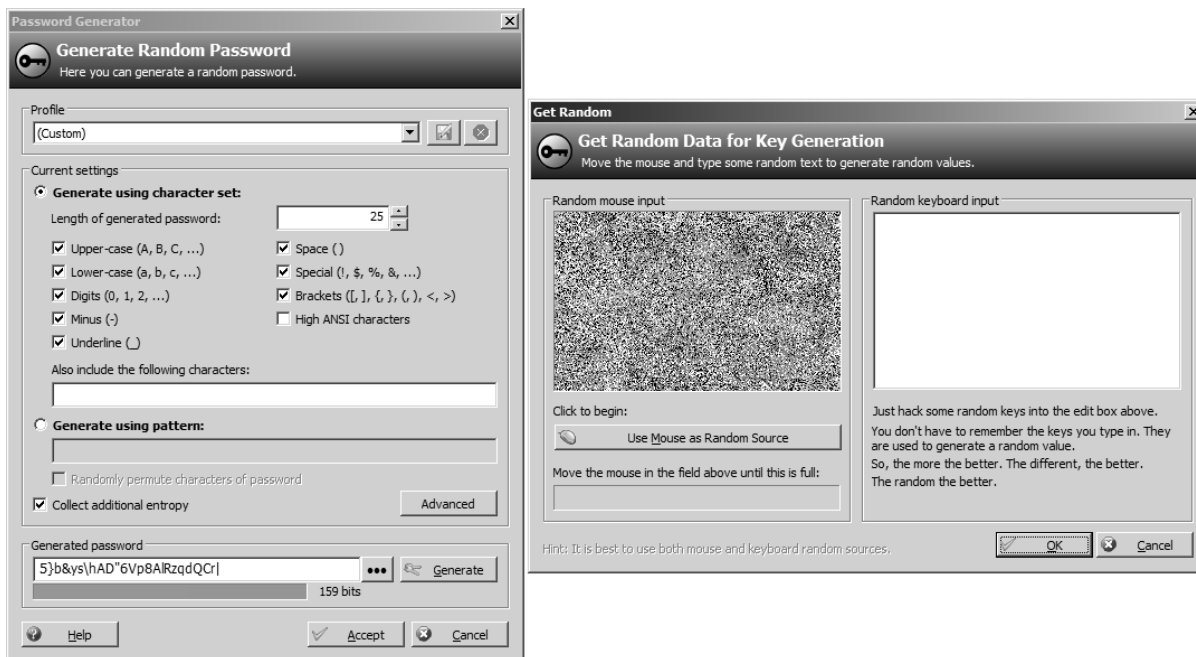


Fig. 122: KeePass Classic Edition password generator. To ensure the password meets the randomness property, the user can supply entropy by typing a text, generate mouse movements, or combine the two. Source: own work.

Tweaking X , Y , and Z based on advances in computational performance, user feedback, and security ensures scalability and less time spent on password management. A script automating the process can be programmed and executed periodically, coinciding with the password expiration policy: users will thus be notified to change their passwords and the new sequence length X_t will be longer compared to X_{t-1} , i.e., $X_t > X_{t-1}$, with $X_{t-1} - X_t$ the absolute difference. While the proposed model does not assume any fixed rate at which X , Y , and Z should increase and no recommendation will be given, if enterprise credential manager is deployed, users can generate passwords of arbitrary length and not required to remember any password, they may hit limits imposed by websites. In a sample of services, it was found some did not provide encryption for data in transit, opening up passive or active interception vulnerabilities discussed in chapter 2.4.4, but also prohibited some characters to be included, limited length, or both. This considerably reduces security and improves brute-force and dictionary attacks' probability of success (T. Hunt, 2011b). The sites in question were revisited at the time of writing and even though few rectified the shortcomings and secured the data transfers, the rest continued to ignore baseline security practices.

Password managers offer sets from which the password is generated. KeePass Classic Edition, an open-source utility, is depicted in Figure 122. The result is displayed in the box on the left side of the figure: the password contain uppercase and lowercase letters, numbers, minus and underline symbols, space, special characters as well as brackets. Length was set to 25 which led to a password whose brute-force work factor is beyond computational capacities of any adversary, and dictionary attack would be ineffective as the sequence does not contain discernible patterns and exploitable shortcomings. To validate the claims, the string was entered into Password Haystack¹ strength estimator. The result is demonstrated in Figure 123.

If the password is kept secret and encrypted, transmitted exclusively over secure channels, and the user is not required to remember it, the adversary has no choice but to resort to brute-force

¹<https://www.grc.com/haystack.htm>

6 Uppercase
 11 Lowercase
 3 Digits
 5 Symbols

5}b&ys\hAD"6Vp8a1RzqdQCr|

Enter and edit your test passwords in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26+26+10+33 = 95
Search Space Length (Characters):	25 characters
Exact Search Space Size (Count): <small>(count of all possible passwords with this alphabet size and up to this password's length)</small>	28,034,052,602,738, 549,436,590,497,089,977, 609,984,418,179,126,495
Search Space Size (as a power of 10):	2.80 x 10 ⁴⁹

Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: <small>(Assuming one thousand guesses per second)</small>	8.91 trillion trillion trillion centuries
Offline Fast Attack Scenario: <small>(Assuming one hundred billion guesses per second)</small>	89.14 thousand trillion trillion centuries
Massive Cracking Array Scenario: <small>(Assuming one hundred trillion guesses per second)</small>	89.14 trillion trillion centuries

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.

Fig. 123: *Password Haystack statistics. The estimates are based on brute-force enumeration under realistic assumptions in online and offline modes, with the latter utilizing parallel computations to speed up the search.*
Source: *Password Haystack, modified.*

attack. The model assumes the first condition (secrecy) is handled by the ICT infrastructure on which password databases for all users are stored. The second condition cannot be enforced locally because some sites do not employ encryption for logins. Social engineering may be an effective way to obtain users' password fraudulently, and organizations should continuously educate employees about the pitfalls of personal data leaks. Decoupling users from security, eliminating password reuse, and providing trivial means of generating and obsoleting passwords tailored for mobile devices and workstations are steps IT personnel must consider, especially when sensitive assets are handled and BYOD is on the rise. Any organization can benefit from the ICT model because it reduces attack surface and streamlines ICT security management.

Three factors influencing password security are complexity, length, and uniqueness. Preferring one at the expense of others may lead to the following vulnerabilities:

- high complexity: authentication by manual input leads to decreased productivity and running the risk of poor credentials management, e.g., writing the string in an unencrypted file; some sites impose arbitrary restrictions on which characters can be included,
- high length: proneness to combination mode dictionary attack in which word list entries are appended or prepended to produce longer strings, often with additional rules employed, e.g., numbers in between, substitution masks, case toggling; some sites impose arbitrary length restrictions,
- high uniqueness: decreased memorability, propensity toward poor password management and if the string was mingled from dictionary entries, proneness to Markov chains and PCFG attacks which generate lexical patterns from existing grammar structures.

A balance must therefore be found and passwords generated using random or pseudo-random data: services exist which provide access to entropy generated from various local sources, e.g., hardware interrupts, quantization noise from analog audio inputs, hard-disk drive seek times, low bits from temperature measurements, key presses and mouse movements (implemented in KeePass as depicted in Figure 122), network packets sequence numbers, and OS entropy pools. Alternatively, data from external entropy sources, e.g., atmospheric and thermal noise, Brownian motion from lava lamps, and clock drifts is available online.

Incorporating additional factor, preferably using another device than the one from which the request was initiated, into the authentication scheme can mitigate risk associated with malware-compromised hosts. FFIEC (2005, p. 3) claims that "... [a]uthentication methods that depend on more than one factor are more difficult to compromise than single-factor methods... The success of a particular authentication method depends on more than the technology. It also depends on appropriate policies, procedures, and controls." The most common procedure is entering a password, for the purposes of the thesis understood as a "... protected/private string of letters, numbers, and/or special characters used to authenticate an identity or to authorize access to data" (Committee on National Security Systems, 2010, p. 53). Passwords are favored for easy implementation, trade-off between comfort and security as well as no additional hardware necessary for correct functioning. On the other hand, "... [i]nsecure work practices and low security motivation among users can be caused by security mechanisms and policies that take no account of users' work practices, organizational strategies, and usability," (Adams & Sasse, 1999, p. 6) all make password composition and revocation policy design highly challenging. Additionally, "... a disturbingly large percentage of users' passwords can be guessed easily. Gaining access to a system is the first step to penetrating it, and so the first issue relevant to password management is the selection, or assignment, of a password that is sufficiently hard to guess so as to deter attacks" (Bishop, 1991, p. 1). Morris and K. Thompson (1979, p. 1) add that "[r]emote-access systems are peculiarly vulnerable to penetration by outsiders as there are threats at the remote terminal, along the communications link, as well as at the computer itself." Despite the weak link the passwords constitute for security, Herley and van Orschot (2012, p. 1) admit that "... passwords will be with us for some time, and moreover, that in many instances they are the best-fit among currently known solutions... For users, the main issue is usability. Major complaints are triggered by mandatory password changes... and complex policies." This suggests relaxing stringent rotation policies would be beneficial for users but detrimental to security due to plethora of existing attacks taking advantage of static, unchanged, weak, poorly-protected, and reused passwords. These were discussed in chapter 2.4.2 and found common in a representative sample of users in chapter 4. Indeed, Amico et al. (2010, p. 1) point out that "[t]o think sensibly about the security of systems that use passwords, it is therefore essential to properly evaluate their resilience to guessing attacks..."

As discussed previously, password selection policies should vary for desktops with dedicated input peripherals and smartphones with touch interface and virtual onscreen keyboards. Yan, Blackwell, Anderson, and Grant (2004, p. 2) state the following: "Good password appear to be random characters. The wider the variety of characters the better. Mixing letters with numbers is better than letters alone. Mixing special characters with number and letters is better still. One recommendation that seems increasingly popular is the pass phrase approach to password generation." Empirical results favor passwords of length at least 16 (Kelley et al., 2012) which consist of non-repeating characters not forming discernible patterns, or found in a dictionary. Otherwise, a dictionary attack can be mounted to reverse engineer the string which was practically demonstrated in case study 1 (chapter 5.1). Password policy must reflect changes in computational performance by increasing minimum number of characters (X, Y, Z parameters introduced earlier) over time.

Tab. 21: *Sizes of alphabet sets used in passwords. While smartphone users can input all 95 symbols, complexity results in higher zero-order entropy than total number of characters. Complex passwords sampled from the full set can be supplanted by longer ones composed from a subset.*
Source: own work.

Description	Size	Cumulative
Numbers	10	10
Lowercase letters	26	36
Uppercase letters	26	62
Special characters	32	94
Space	1	95

To determine password strength, entropy is most commonly used as a validating measure. proposed by Shannon (1948), it is defined as “a measure of ‘uncertainty’ or ‘randomness’ of a random phenomenon” (Ihara, 1993, p. 2). Entropy is at maximum when the next symbol in a sequence has the same probability of being selected as any other in the alphabet given all the previous symbols are known, i.e., the adversary can at most guess the next character (governed by discrete uniform distribution) regardless of what she knows about the password so far. For example: the string in Figure 123 and a dictionary word “tablet” differ in that the former is not predictable at any point. Even if the attacker manages to uncover part of the string, it would not give away any indication about the length or composition of the remainder. However, should “tablet” be partially uncovered, e.g., “tab,” entries containing the sequence can be trivially enumerated: choosing a word, albeit a long one, without applying any transformation or expansion technique makes the password susceptible to straight mode dictionary attack described in chapter 5.1.3. The situation cannot occur during real-world reverse engineering, though, because passwords are scrambled by a one-way function. Therefore, the password is either cracked in full when a positive match is made, or not at all. Per-character probabilities are useful solely when constructing candidate lists which apply techniques such as PCFG (chapter 5.1.3). Composition policies must disallow known weak strings and predictable patterns in favor of algorithms generating random-looking data from sources rich in entropy, optimally not displaying the result to the user and save it in a personal database encrypted using another strong password, or accessible via multi-factor authentication. The entropy type cited when measuring password strength, zero-order entropy, is calculated using Equation 6.2.2.

$$N \times \log_2 C, \tag{6.2.2}$$

where N is password length, and C number of all possible symbols from which it is composed. For ISO basic Latin alphabet and special keyboard characters, different sizes of C are given in Table 21. For a sample string “2w) :\$‘KnbY’DPK GK=‘”, zero-order entropy is calculated as

$$16 \times \log_2 94 = 104.87342 \dots \text{ bits.} \tag{6.2.3}$$

The password exhibits high uncertainty as to the next character in the sequence, but it also negatively affects productivity if typed repeatedly to authenticate, particularly on a virtual keyboard. Minimum password policy requirements specifically tailored to BYOD should be

Tab. 22: Password lengths for fixed zero-order entropy. To reach 128 bits of entropy, users would have to choose password of lengths 39, 23, 22, and 20 if they decided to compose it only from letters, lowercase and uppercase letters, the combination of the two, and a full 95 printable character set, respectively. Source: Toponce (2011), modified.

Fixed entropy [bits]	Numbers	Alphabet	Alphanumeric	All characters
32	10	6	6	5
40	13	8	7	7
64	20	12	11	10
80	25	15	14	13
96	29	17	17	15
128	39	23	22	20
160	49	29	27	25
192	58	34	33	30
224	68	40	38	35
256	78	45	43	40
384	116	68	65	59
512	155	90	86	79
1024	309	180	172	157

set either using the Markov chain discussed above, or by alternative means. It is safe to expect users will gravitate toward the minimum length rather than select stronger, random, and non-trivially predictable passwords (Florêncio & Herley, 2007). Two approaches are viable: either fix entropy the authentication strings should contain, or provide users with clues to gauge password strength. By fixing entropy to a predetermined value, everyone can select password according to their own preferences, the only limiting factors being the entropy and neither predictable patterns nor dictionary words. The second approach is to impose lower bounds on the number of characters from each class so that C in Equation 6.2.2 is maximal and the search space spans all sets, together with minimum password length restriction. Fixing entropy seems less restrictive because complexity may supplant length, i.e., users who do not wish to have special symbols may forego them in favor of a longer string composed from other sets as long as the entropy criterion holds. Table 22 supplies number of characters for each alphabet size to reach fixed entropy.

Compare the value KeePass Classic Edition strength calculator produced in Figure 122, totaling 159 bits, to line 7 in Table 22 which lists composition requirements to reach 160-bit entropy. For the 95-character alphabet size, 25 characters are needed which aligns perfectly with the sample password's length. Therefore, a conclusion can be made KeePass utilizes zero-order entropy as an estimator. The fact is corroborated by official documentation which also provides the following entropy-to-quality estimation (Reichl, 2013):

- 0–64 bits: very weak,
- 64–80 bits: weak,
- 80–112 bits: moderate,
- 112–128 bits: strong,
- 128 bits and more: very strong.

Entropy levels should be selected according to the privileges the account is granted. To future-proof, standard users are recommended to meet the 128b criterion while those with

elevated privileges are advised to at least double the amount. Both groups can store their passwords in encrypted containers and allow the software to supply them automatically into the respective form field during login, bypassing the need to copy and paste the string manually, thus making password length a non-issue. The database will prompt for authentication details (knowledge/possession factor or their combination) on open; users need to remember and supply a single password as others will be at their disposal in a secured file. This substantially reduces the attack surface. The encryption scheme should be selected based on whether practical, computationally-feasible attacks can be mounted against it, and IT personnel must reflect on future developments by expiring and supplanting alternative scheme if the current encryption is found broken or severely weakened.

Dictionary words and patterns exhibit very low entropy and are not suitable to protect user accounts, especially those containing sensitive assets, e.g., VPN credentials, desktop logins, and accounts with superuser privileges. Password security is positively correlated with length, i.e., longer strings adhering to best practices increase time to reverse engineer, and negatively correlated with attacker's computational strength: the higher the amount of resources dedicated to compromise passwords, the lower the security, primarily for sequences not conforming to composition rules. Those can be aided by a password strength meter, "... a visual representation of password strength, often represented as a colored bar or screen. Password meters employ suggestions to assist users in creating stronger passwords" (Ur et al., 2012, p. 1). The meter computes zero-order entropy in real-time and graphically notifies user on a semantic differential scale, e.g., too short, weak, fair, strong, often color-coded (red, orange, yellow, green). Egelman, Sotirakopoulos, Musluhkov, Beznosov, and Herley (2013) devised a model where "... experimental condition framed the password in terms of social pressure by presenting strength relative to all of the users on the system," and found "... that password meters—both traditional and those based on social pressure—can nudge users towards creating stronger passwords. However, nudging users to create stronger passwords may have drawbacks if users cannot remember them or choose to revert to weaker passwords." Another disadvantage is subjectivity: each threshold entropy value must be set manually or based on a real-world data set. Techniques (mutators, PCFG) exist that considerably reduce the time factor involved to reverse engineer a given string despite its very high entropy. They were briefly discussed in chapter 2.4.2, and mutators were demonstrated practically in case study 1 (chapter 5.1).

A password reuse policy must be put in place to prevent users from entering identical strings when deprecated. Chapter 2.2.2 dealt with cryptographic hash functions, the preferred way of storing password fingerprints. Organizations are strongly advised to concatenate the string with pseudo-random or random data (salt), and prefer constructions which deliberately incur performance overhead, e.g., Password-Based Key Derivation Function 2 (Network Working Group, 2000), bcrypt (Provost, 1999), or scrypt (Percival, 1999).

6.3 Model Metrics

The final chapter will present metrics which should help the CISO calculate the potential savings and decide whether the proposed model is a financially viable alternative. Even though effort was taken to include only quantitative indicators, at least one aspect, user satisfaction, is subjective and should be treated as such. Inertial mindset may skew the results if users are polled in early phases of implementation where the organization undergoes transition between the two states, abandoning old ICT security process in favor of a new one. It is recommended users be polled at various points to analyze developments in comfort, work efficiency, internalization of best practices, and impact of SECaaS in production environment. If the system is not accepted

after a predetermined period or the metrics do not indicate improvement, CISO should poll IT personnel and users for suggestions, and present alternatives to the management board.

The ICT model does not claim to fit any organization because many factors influence it apart from those discussed so far: corporate culture, ICT investments, user mindset and approach toward new technologies as well as types of data processed and stored. While most legal entities have sensitive assets in need of protection, some only deal as mediators, e.g., shops with online payments which redirect customers to a third-party portal without retaining any information about the transactions. However, even purchase histories should be considered sensitive and protected accordingly, in particular related to health-related and products and services. The extent to which different records should be protected is debatable when they fall outside of Gramm–Leach–Bliley Act (GLB), Federal Information Security Management Act (FISMA), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), Sarbanes–Oxley Act (SOX), and other industry-specific regulations. It is the author's view organizations are obliged to secure any and all data which, if disclosed to unauthorized third parties, would allow the party to identify or learn about the individual's preferences, commercial, physical, political, social, and religious traits disclosed solely under implicit belief such personally-identifiable, non-public information will be adequately protected and not shared unless an explicit permission was given to do so. An exception are strictly anonymized data which were scrubbed and do not leak any intelligence usable to collate, reconstruct, trace, and profile anyone whose records were included in the batch. This should be clearly indicated before the data in question are imparted. Legal aspects of data sharing is outside the thesis' purview.

Two groups of metrics will be delineated corresponding to the user-side and ICT-side security. Fully loaded cross-industry mean hourly wage rate was selected as a baseline economic measure. The figure will not be specified but can be determined by polling The U.S. Bureau of Labor Statistics or equivalent body publishing workforce-related data. When preparing financial evaluation, it is strongly recommended to use up-to-date sources of information rather than rely on historic developments. Predictions on how hourly wages will likely develop may help justify project expenses in the future.

Economic benefits are counterbalanced by adoption and implementation costs. These are highly dependent on the current state of ICT infrastructure, nature of the industry, number of employees, security policies implemented and other factors, and their calculation will not be attempted. As a tertiary educational institution, Tomas Bata University's ICT development can be classified as above the average due to it being a member of the National Research and Education Network (NREN), and deployment costs would thus not substantially tax its budget. The hypothesis was not evaluated and is based on author's limited knowledge acquired from informal interviews with IT personnel. Costs associated with shifting users' mindset might be considerable, especially in workers with little background and interest in security whose operational routines come from repetition, not experience. This group, highly susceptible to social engineering, should be prioritized during training but not included in pilot usability testing. Once the system has been tailored based on received feedback, decoupling inexperienced users from security need to be given preference. GUI design, intuitive interface, one-click functionality, and prominently visible help and hints which present the recipient with a streamlined experience presenting security as a service, not an obstacle. A questionnaire campaign among the employees, or tracking usage patterns on workstations and BYOD devices may help IT personnel determine which group does the employee fit into. Experienced users should be given the option to customize the service without weakening security, e.g., keep local copies of database files and synchronize them between devices, increase the number of encryption rounds, choose a second authentication factor (one-time passwords, biometric identification), etc. Minimum thresholds should be set for everyone, though.

Quality of the model's economic analysis relies on quality of the data. Care should be taken to ensure the measurements entering the analyses are accurate, objective, reproducible, and checked for anomalies. For example: with users' consent, tracking software installed on workstations can provide accurate data on how long it takes before an authentication string is inputted in a form, the number of failed attempts, typing speed, whether a credentials manager is used, password strength, and number of logins per day. Even though manual tracking is possible, the resources and time expended may not justify it, especially because accuracy and reproducibility can be questioned.

Metrics for user-side security of the proposed model are listed in Tables 23 and 24. The structure is based on a study by O'Connor and Loomis (2010) which quantified economic effects of RBAC deployment.

Tab. 23: *User-side model metrics 1. Time span for the metrics may be set arbitrarily but it is recommended to include a 30-day and one-year period so that the benefits are aggregated over long term.*

Source: own work.

Benefit category	Technical measure	Economic measure	Economic impact metric	Unit scaling factor
Social engineering resilience	Change in administrative time before test emails are classified as fraudulent and deleted (hours)	Fully loaded cross-industry mean hourly wage rate	Change in total time \times loaded hourly wage rate	Number of tested employees
Social engineering resilience	Change in number of consultations with IT department about requests for internal/sensitive/personal data from third parties	Fully loaded cross-industry mean hourly wage rate for users and network system administrators	Quantity \times sum of loaded hourly wage rates	Number of consulting employees
Social engineering resilience	Number of test pages requesting internal/sensitive data classified as fraudulent after training	Cost per lost entry	Number of incorrectly categorized sites \times cost per lost entry	Number of tested employees
Suspicion-based behavior	Change in time before a check is required after a test request is made for internal/sensitive data in person (hours)	Fully loaded cross-industry mean hourly wage rate	Change in time \times loaded hourly wage rate	Number of tested employees
Suspicion-based behavior	Amount of sensitive database records the user imparts to a third party	Cost per lost entry	Number of records \times cost per entry	Number of tested employees
Suspicion-based behavior	Amount of sensitive database records the user hands over to an insider	Cost per lost entry	Number of records \times cost per entry	Number of tested employees

Tab. 24: *User-side model metrics 2. The fully loaded cross-industry mean hourly wage rate impact metric differs across industries and should be updated to reflect the current situation.*

Source: own work.

Benefit category	Technical measure	Economic measure	Economic impact metric	Unit scaling factor
Password management	Decrease in time for managing, revoking, and generating passwords (hours)	Fully loaded cross-industry mean hourly wage rate	Change in time \times loaded hourly wage rate	Number of employees
Password management	Time per login	Fully loaded cross-industry mean hourly wage rate	Time \times loaded hourly wage rate	Number of employees
Password management	Decrease in time for password input to a form field (hours)	Fully loaded cross-industry mean hourly wage rate	Change in time \times loaded hourly wage rate	Number of employees
Password management	Time spent on failed authentication attempts (hours)	Fully loaded cross-industry mean hourly wage rate	Time \times loaded hourly wage rate	Number of employees
Training	Change in score on a self-assessment test before and after training	Training curriculum cost	Training curriculum cost / change in score (cost per point)	Number of trained employees
Training	Change in score on a post-training quiz presented occasionally after system login	Training curriculum cost	Training curriculum cost / change in score (cost per point)	Number of trained employees
Overall satisfaction	Qualitative research on user satisfaction with the security policies and security system	–	–	Number of employees

Benefit categories correspond to the main areas outlined in the ICT model. For anti-social engineering techniques, the model utilizes metrics obtained in testing scenarios because real adversarial campaigns cannot be evaluated for efficiency due to uncontrolled conditions. It is assumed the constructed scenarios are indistinguishable from genuine ones and give the target group no indication a test is being conducted. In case of phishing emails, plausibility can be increased by omitting properties users associate with spam emails (non-standard punctuation, marketing messages in the body) and customize the message via inclusion of first name, changing the sender address to one belonging in the same domain (chapter 5.2.3), and wording the text so that emotional impulses are activated in place of logical reasoning. The outcome is twofold: either a per-user unique link included in the email is clicked and IT personnel logs the campaign's success, or the link is disregarded. A suitable time period should be allowed for the members of the target group to read the email and make a decision. As the email server is hosted on monitored ICT infrastructure, the period between the email is opened and action is taken (click the link, ignore/delete the message) can be quantified and transformed into an economic measure. The preferred outcome is immediate deletion or taking no action after the fraudulent test email was opened.

Over long term, the number of consultations regarding social engineering campaigns initiated with the IT department should be analyzed, especially after controlled social engineering campaigns have been launched, to determine how knowledgeable users subjectively perceive themselves. The figures should be correlated with actions taken and classes established, e.g.:

- users initiated contact and reacted correctly,
- users initiated contact and reacted incorrectly,
- users did not initiate contact and reacted correctly,
- users did not initiate contact and reacted in correctly,

If the performance of the first and the third group are consistent in time, it can be assumed users are capable to discern social engineering with assistance or autonomously. Employees in the second and fourth category should be polled as to why they made the decision, and additional training suggested until an improvement is observed, i.e., initiating correct action with or without help of IT personnel. The preferred outcome is consistent inclusion in either the second or the fourth group over long term. Setting the objective that all users should be able to discern between legitimate and fraudulent attempts on their own is overlystrict.

The last metric in social engineering resilience is a barrage of websites users are presented with after undergoing training curriculum devised in Table 19. Using their browser of choice, the objective is to enter the data required on the site, indicating willingness to impart it, or refusing to do so and specify reasons. Multiplying the cost per lost entry with the number of records imparted on websites incorrectly classified as trustworthy (false positive) may give the testing authority approximate figure on the financial loss should such an attack be executed by a malicious party, and also how skilled users are in differentiating the sites. A more effective approach would be to supply the page when it is requested in place of real one. This would eliminate a situation where the individual simply guesses the answer, but data confidentiality must be assured in case the target is not able to recognize the fraudulent site and inputs their credentials or other sensitive data. The method is as far as the IT personnel can legally converge the scenario with a real-world social engineering campaign. The preferred outcome is consistently correct classification of the test cases.

Suspicion-based behavior's economic metrics also exploit constructed scenarios, however, the target is approached physically which removes the anonymity barrier and introduces psychological effects skewing human behavior. The results can be expected to differ from social engineering

resilience and the tests should be documented for comparison purposes. The first metric, time until a check is made after a request for sensitive/personal data denotes how long it takes for the information classes scheme presented in chapter 6.1 to be invoked, a proxy variable for whether the distinction was internalized. If no check is required, the user does not perceive any difference between internal and sensitive assets security-wise. The preferred outcome is denying access to the data, requesting authentication, or consulting IT personnel after the demand has been made.

The second metric, amount of database records handed over to an insider simulates threat from within the organization. Suspicion-based behavior should encompass requests originating from both authenticated employees and third parties, even though it can be expected the former group will be subjected to fewer checks which opens the attack vector where malicious insider can obtain sensitive data they have neither authorization nor need for, violating the principle of least privilege. Care should be taken when preparing the test to select a correct employee who will pose as the requesting party: unfamiliar to the target but nevertheless sharing their organizational unit as well as knowledgeable of the processes and terminology so that plausible test scenarios can be constructed. Should the target agree to provide the data, willingness to plug in and copy the data onto malicious removable drives can be evaluated as well, giving the insider opportunity to infect the host with malware. The amount of internal/sensitive records provided is directly translatable to financial losses if cost per entry is quantified. The target should be notified of the result and of the hypothetical costs they incurred to the organization afterwards, but the insider should not be identified. This may lead to selective screening should a legitimate request be made by the same employee in the future. The preferred outcome is refusing access to the data, or notifying authoritative third party to advise on the situation.

Password management metrics are divided between time spent managing passwords throughout their life cycle, beginning with generation, storage, handling, securing, and ending with revocation. The period can be eliminated by deploying credentials manager located in a remote data center or hosted locally. The cost therefore represents losses for the organization if the management takes place without such software. The model considers passwords to be an area where improvements not only reduce attack surface porosity, thereby increasing security, but are also economically justifiable because passwords permeate every aspect of day-to-day operations. Streamlining the authentication process may increase user satisfaction (discussed below) by eliminating the need to manage login credentials manually. Decoupling human element from security while providing fast and reliable service on demand can make password revocation and generation a one-click, instantaneous event taking at worst several seconds, and resulting in a string in conformance with ICT policy requirements. It is strongly recommended enterprise password management be deployed first and measurements pertaining to time spent on password management (first metric) and time per login (second metric) used to evaluate LastPass or equivalent solution. Table 20 listed financial calculations with data from a sample of LastPass clients. If the data which reflects current situation is known, the figures will faithfully represent savings the management can expect. Infrastructure utilization being a marginal issue as organizations frequently have overprovisioned hardware resources and cloud computing offers pay-per-use virtual instances, the measure is justifiable both from security and usability standpoints. The third metric, time spent on failed authentication attempts, also represents costs eliminated by password managers. The preferred outcome is reduction in time spent on authentication-related activities.

Assessing training efficiency via comparison of tests conducted before and after the training is concluded (first metric) determines immediate effect of the regimen and should show statistically significant difference in point scores. Dividing the cost per individual by the change denotes a “cost per additional point.” The expected outcome is a positive ratio. However, not reinforcing

the acquired information will result in forgetting, and the organization must strive for continuity in education plans because one-off events do little to instill change, especially when the goal is to modify ingrained patterns of behavior, e.g., in social interactions. The second metric is thus proposed: at various points in time, employees are subjected to a mandatory quiz whose results are compared with pre-training base score and the points achieved afterwards. This way, temporal data are available to determine the point at which the training should be commenced anew. Moreover, the findings on how employees internalized the information may help to direct focus on groups which exhibited lower scores, or change the training methods and curriculum if the results are consistently below the base score. This will also suggest individuals have not transformed the information into knowledge applicable in everyday situations. The preferred outcome is increase in point scores over long term.

Overall satisfaction with the new security measures should be evaluated via interviews and questionnaire research where the target group is polled in discrete time intervals. No economic measure was assigned but the opinions may point out shortcomings and suggestions. Inherently subjective, interviews and questionnaires should be understood as reflecting participants' attitudes toward change, resistance to abandon a working system for an unproven alternative, and personality traits which preclude generalization from small samples to the entire organization. A feedback loop between users and IT personnel with the CISO a facilitator is visualized in the ICT governance model (Figure 113) and should not be neglected.

Metrics for ICT-side security of the proposed model are listed in Tables 25 and 26. They are purposefully structured in the same way as user-side metrics for fast overview and ease of orientation.

Tab. 25: *ICT-side model metrics 1. Time frame for the metrics should coincide with user-side part and include figures per month and year.*
Source: own work.

Benefit category	Technical measure	Economic measure	Economic impact metric	Unit scaling factor
BYOD management	Change in amount of secure connections to internal/sensitive data from mobile devices	Cost per lost entry	Change in amount \times cost per lost entry	Number of employees enrolled in BYOD management program
BYOD management	Change in time spent on BYOD-related activities (hours)	Fully loaded cross-industry mean hourly wage rate for network system administrators	Change in time \times loaded hourly wage rate	Number of IT personnel
Password resilience	Change in zero-order entropy-measured password strength	Estimated cost to reverse engineer	Change in strength \times estimated cost to reverse engineer (cost per additional bit of entropy)	Number of users
Password resilience	Time spent in a password manager GUI between initiating and concluding a session	Fully loaded cross-industry mean hourly wage rate	Time \times loaded hourly wage rate	Number of employees
Password resilience	Time to generate a strong password after issuing a request	Fully loaded cross-industry mean hourly wage rate and ICT infrastructure operating cost	Time \times loaded hourly wage rate and ICT operations cost	Number of employees

Tab. 26: *ICT-side model metrics 2. Data for some economic impact metrics (cost per hour of inactivity) may not be readily available and should be assessed on a per-organization basis.*

Source: own work.

Benefit category	Technical measure	Economic measure	Economic impact metric	Unit scaling factor
Infrastructure and network hardening	Change in number of internal/sensitive records the hosts an insider is on the network segment with can access	Cost per lost entry	Change in number \times cost per lost entry	Number of users
Infrastructure and network hardening	Change in number of internal/sensitive records clients have access to	Cost per lost entry	Change in number \times cost per lost entry	Number of users
Infrastructure and network hardening	Change in amount of privileges granted to host	Cost per privilege by exploitability	Change in amount \times cost per privilege	Number of users
Infrastructure and network hardening	Number of known unpatched vulnerabilities in hosts on a network segment	Cost per vulnerability based on severity and number of internal/sensitive records the host has access to	Change in vulnerabilities \times cost per vulnerability	Number of users
Incident response	Change in downtime in test scenarios (hours)	Cost per hour of inactivity	Change in downtime \times cost per hour	Organization
Incident response	Change in administrative time before test connectivity issues are resolved (hours)	Cost per hour of inactivity	Change in time \times cost per hour	Organization

Aspects of ICT-side security not discussed in previous chapters were left out even though links between them exist, such as between patch management and infrastructure hardening. This makes classification of metrics into categories challenging. Benefit categories correspond to the main areas outlined in the model, those omitted were not included and suitable indicators can be set up by IT personnel in cooperation with the CISO. For BYOD management, the two benefit categories pertain to losses averted using profiles on mobile devices enrolled in the program. Some employees may opt out and refuse presence of restrictive software on the device they own. Objectively and clearly communicating the advantages and disadvantages of the proposed solution may encourage employees to install the profile for a test period. Nevertheless, should the opt-out decision prevail, technical measures limiting or disallowing sensitive data access for this class of devices, or establishing a separate network segment where connection requests are redirected if insecure configuration is detected, are recommended. While passive and active network traffic intercept may still occur, the combination of a segmented network and enterprise password manager restricts the threat domain because the adversary cannot access servers located in other segments. Even though noisy techniques can still be used, they should be immediately detected and incident response initiated. The preferred outcome for the first measure, change in amount of secure connections, is increase as more devices are enrolled, the second (time spent on BYOD-related activities) should decrease after the initial phase where features included in the profiles are programmed, tested, and deployed. The economic interpretation depends on cost per lost entry, though, which is not trivially quantified.

In total, five economic impact metrics are challenging to distill into a single figure: cost per lost entry, estimated cost to reverse engineer a password, exploitability, cost per vulnerability based on its severity, and cost per unit of internal/sensitive data. A fifth measure, cost per hour of inactivity, can be extrapolated from historic balance sheets and cash flow statements. Cost per lost entry, C_{le} , can be supplanted by cost per unit of internal/sensitive data because the two are related. C_{le} aggregates loss of any record regardless of category, and is expected to lie in the $C_{ii} < C_{le} < C_{si}$ interval, where C_{ii} denotes cost per lost unit of internal data, and C_{si} cost per lost unit of sensitive data. This implies C_{le} is arithmetic mean of the two values, although organization may use another suitable location parameter. Calculating C_{ii} and C_{si} should begin by categorizing electronic assets unequivocally as internal or sensitive (public data are freely available and cost per unit is thus 0), and determining their value by summing losses incurred should the data be expropriated. The losses should include audits, purchasing costs of hardware, software, and expertise, penetration testing and vulnerability assessment as well as certification of compliance but also indirect losses from decreased productivity, customer attrition, reputation damage, public relations management as well as trust-rebuilding efforts, particularly if the leak is disclosed publicly. External consultation services may allow for precise economic calculations which should be periodically revised.

Estimated cost to reverse engineer a password correlates with its strength and fluctuates as prices of cloud computing VMs decrease over time. As of December 31, 2013, Amazon Elastic Compute Cloud (EC2) charges 0.650 USD per hour for g2.2xlarge instance which runs a Linux distribution of choice and a dedicated graphics processing unit (GPU). As mentioned in case study 1 (chapter 5.1), GPU is currently capable to brute-force strings of length 8, and the parameter will increase in the future due to advances in circuit design, algorithmic optimization, component integration, and energy efficiency. When the string length is above the minimum character count, exponential nature of brute-force enumeration, mentioned in chapter 2.4.2, causes the attack to run prohibitively long which drives the cost upwards. The phenomenon, titled “exponential wall,” is depicted in Figure 124.

Brute-force enumeration is prohibitively taxing even for shorter passwords unless parallel computations are used. In real-world situations, dictionary attacks are predominant and compara-

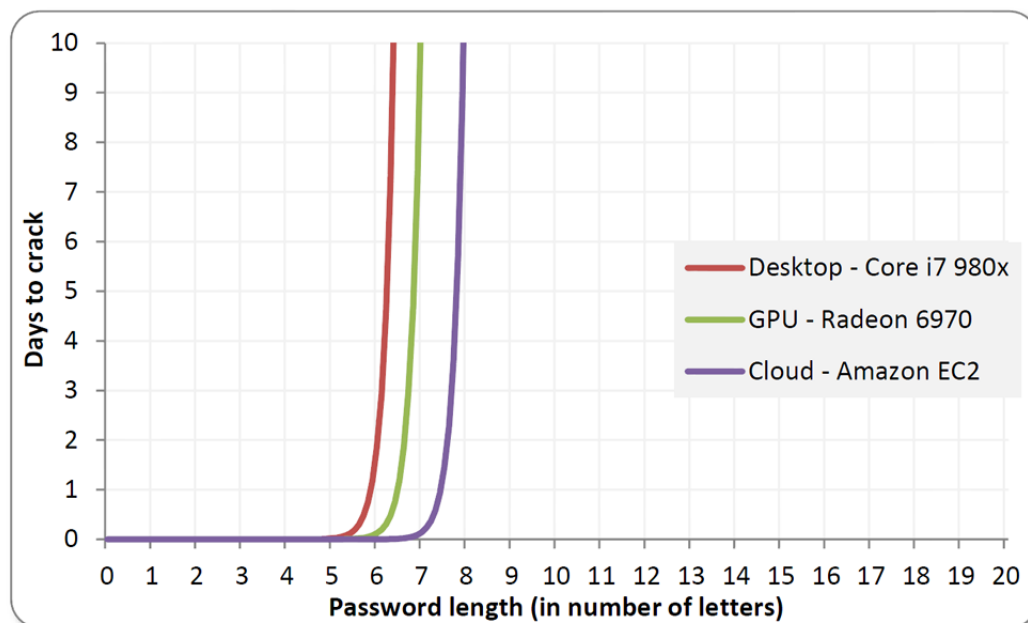


Fig. 124: Password reverse engineering exponential wall. In the future, the curve will shift to the right on the x axis toward longer password lengths.
 Source: Graham (2012).

tively more effective. If a password manager is deployed which generates strings the users do not remember but copy from a secured container utilizing a strong encryption scheme (solutions were mentioned at the end of chapter 6.2.4), even dictionary attack does not result in a breakthrough. IT personnel should enumerate strength of all authentication strings using zero-order entropy (first metric) by mandating domain logins and capturing the passwords as they are passed for evaluation. Alternatively, a pop-up window explaining the analysis and asking users for cooperation can be used. The data must be anonymized and only used for evaluation purposes. The ICT model is based on the assumption database entries are irreversibly encrypted, and sampling passwords from them is not possible. After the statistics are obtained, up-to-date hardware benchmarks should be polled and cost assigned based on prices of commercial hardware and VM instances (EC2 was demonstrated as an example). It is hypothesized zero-order entropy will be low initially, especially for users who remember their passwords. The preferred outcome is statistically-significant increase in strength after enterprise credentials manager has been deployed.

The second and third password resilience metrics pertain to simplicity of password manager's GUI and computational overhead when generating strings, respectively. Neither should exceed seconds, though the algorithm may take slightly more time due to multiple entropy sources employed for meeting the randomness property. Even then, the delay may not even be perceptible from the user's viewpoint. Ideally, the session consists of steps delineated in Figure 125.

Line lengths scale to time: sourcing entropy and waiting between the user selects password requirements and is presented with the result positively correlate with security: the longer the password requested, the longer time it takes to get more random bits. Because the server only returns data from the entropy pool, the critical phase is acquiring sufficient amount of unpredictable data out of which the string is created. IT personnel can poll online sources or acquire physical devices known for high-quality random data. Technical details are beyond the scope of the thesis but a brief overview was provided in chapter 6.2.4. The preferred outcome is for both times to be low.

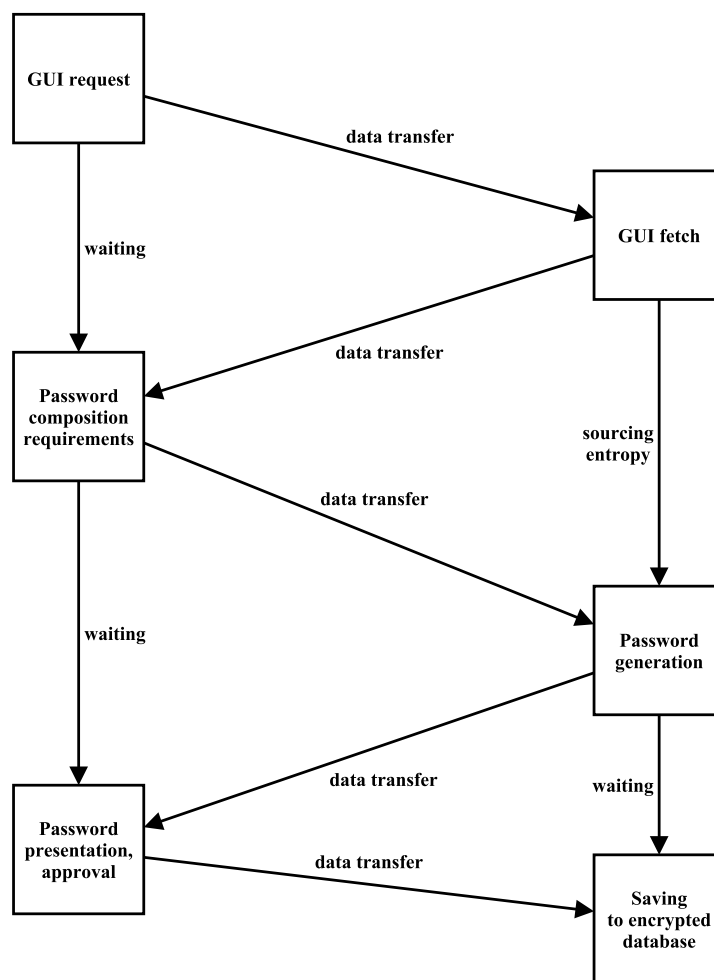


Fig. 125: Password generation session. Left side represents actions taken by the user, right side operations performed on the server.
Source: own work.

Internal network segmentation and ICT infrastructure hardening were aggregated and four economic metrics devised. The first one is a two-step measure of how many internal/sensitive records the adversary can access through other hosts if she gains foothold in the internal network where they are located. The lowest value equals 0 and represents a single host on a subnetwork, another scenario which results in the metric being 0 is no workstation on the network can interact with sensitive electronic assets. In practice, the metric will be non-zero. The indicator is based on the assumption the perpetrator (external or internal) resorts to attacks on the same segment while avoiding noisy, detectable techniques. For larger organizations, a single network may contain dozens of vulnerable hosts, particularly when patch management policy is not enforced. This opens up surplus of opportunities for system penetration and data expropriation. The second metric, number of internal/sensitive records any host can access, can be tracked over time and the principle of least privilege honored. IT personnel may use the following two approaches:

- “shrinking:” all records are initially available; if some are not used over time, access is revoked and the entries become unavailable,
- “expanding:” no records are initially available; if any is accessed, access is automatically granted as per the principle of least privilege,
- combination: expanding with eventual shrinking (recommended for security), shrinking with eventual expanding.

As per the principle of least privilege, granting users the absolute lowest set of entitlements necessary for achieving their task must be preferred. The time frame for evaluating whether the particular record set has been accessed can be set aggressively (day) or permissively (week, month), but longer intervals expand the attack surface. The tracking can be executed server-side. The preferred outcome is a value balancing network management with security (first metric), the number of records for which access was granted should be the smallest subset needed over long term.

The last two metrics utilize exploitability and cost per vulnerability. Exploitability refers to economic evaluation of privileges the user has in case their station is exploited for malicious purposes. This correlates with the number of accessible records and denotes severity of any operation executed on them. For example: if only the “read” privilege is active, the adversary can at best expropriate the sensitive database entry, threatening confidentiality and equaling the cost per lost entry. The “write” permission, however, allows to inject arbitrary data in place of the original entry, threatening availability and integrity as well as endangering hosts who read the value. The “execute” privilege alone can lead to device and network segment compromise as well as data expropriation, and thus constitutes the most potent attack vector which must be designated as economically taxing and granted with utmost consideration. Assigning more privileges does not compound the economic effect: the “highest” one in the read→write→execute sequence is selected. Calculating economic impact of exploitability is challenging and while shortcuts exist (cost per lost entry), organizations are advised to revise user privileges policy assignment. Inflating the metric for reasons such as higher user comfort and less time spent on on-site management by IT personnel is a security weakness, and the practice needs to be discontinued in face of expanding ICT threat landscape and the gap between how complex computing systems work and how users are able to operate them. Convenience should not dominate security. The preferred outcome is reduction in the amount of privileges granted over time to coincide with the principle of least privilege.

Cost per vulnerability presupposes IT personnel follows reputable vulnerability databases which disclose and rate new venues of exploitation. Common Vulnerability and Exposure (CVE) employs a Common Vulnerability Scoring System (CVSS)² which forms a comprehensive framework incorporating factors such as impact on the constituents of the CIA triad (chapter 2.2), propagation vector, remediation level, and collateral damage potential. Being an open standard, CVSS can be utilized for scoring vulnerabilities in industry-specific systems for which full disclosure is not practiced. IT personnel is strongly discouraged from using ad-hoc schemes in place of standardized and documented practices which allow common comparison basis to be established. In CVSS, a single numerical value is calculated, published, and communicated using a semantic scale (low, medium, high). If the affected product is integrated in the ICT infrastructure, the window of opportunity for the adversary is active until the patch is deployed, the risk mitigated, and score reset to 0. Economic impact must correlate with severity: whether the affected software is Internet-facing, reachable on the internal network and to how many hosts (segmentation) or not directly addressable; how many internal/sensitive records are administered or reachable using the program; which privileges does the software have (read, write, execute); and if working exploits are known to exist. Additional factors can be incorporated if their effect is deemed substantial. The proposed model metric quantifies economic impact of unpatched vulnerabilities, i.e., for which vendors have not released fixes or a fix has not been deployed yet. It is tied to segmentation and patch deployment policies, and complements them when window of opportunity’s breadth and depth is estimated. The preferred outcome is maintaining the number as low as possible over time.

²<http://www.first.org/cvss/cvss-guide>

Incident response metrics indicate economic benefits of efficient out-of-bounds situation handling. The first one, change in downtime in test scenarios, refers to a corrective or preventive maintenance discussed in chapter 2.2.3 during which mission-critical systems should be transferred over and run from backup instances. Corrective maintenance can rarely be predicted or simulated on live systems without incurring business inactivity penalties. Therefore, the metric employs test scenarios vetted by management and executed during off-hours. At least two cases should be investigated: before and after the change in incident response methodology using the PDCA logic discussed in chapter 6.2.3. A controlled power outage can be launched locally to assert readiness for such a scenario. The time between test commencement and full data availability constitutes corrective downtime which impedes business continuity and should be minimized. The preferred outcome is reduction in downtime after incident response plans revision.

The second metric, change in administrative time before test connectivity issues are resolved, should be of particular importance for organizations which use remote data centers (cloud computing) for real-time VDI or backup solutions. The objective is seamless transition to a redundant connectivity point without observable consequences to bandwidth and latency, and at least two scenarios should be executed: before and after incident response plans revision. Network parameters and behavior under increased loads should be inspected and corrections made if either quality of service or performance falls below expectations. User response can be also analyzed when the experiment is conducted during business hours. The metric uses identical measure as the previous one, cost per hour of inactivity, despite the organization being functional. Because information and communication technology forms integral part of business operations, a drop in connectivity paralyzes productivity and is essentially equal to loss of revenue from inactivity. The preferred outcome is reduction or elimination of network outages caused by single points of failure.

The overview should not be understood as a definitive source of metrics but as a suggestion to be customized based on industry, total number of employees, type and amount of sensitive data processed and stored, and other factors which influence organizational ICT security process governance. It is the author's opinion data from which suitable impact metrics can be constructed is available, and it is management's prerogative to use it in an objective manner for stakeholder-oriented security decisions.

6.4 Conclusion

The proposed ICT model does not aim to comprehensively map the security landscape of organizations because rapid developments would make any such attempt obsolete quickly. It focuses on areas the author considers critical based on overview of secondary sources as well as from primary research. Some trends, e.g., BYOD and cloud computing, can be expected to proliferate in the future, while others, e.g., password management and patch deployment are commonplace in users and organizations, respectively. Technology blurs the distinction between personal and work spaces, and policies must be modified to accommodate the changes.

It can be argued some economic measures in the model are challenging to measure accurately, e.g., cost per lost entry and cost per privilege by exploitability. The author believes categorizing data into classes as suggested in chapter 6.1 is the first step toward approximate quantification of economic impacts caused by lacking security. The organization may assign a fixed cost per record in each category and multiply it with the number of entries, although this would implicitly assume all data within the same class is equal which may be overly simplistic.

Another concern is that the model is primarily technical, and sidelines economic considerations. The author argues low-level implementation details and theoretical background were kept to a minimum, but explanation of key concepts had to be provided. On the other hand, sidelining the ICT aspect in favor of economy would result in a general framework which does not respect real-world specifics of security engineering. The author argues the model found a balance between accessibility and ways to make practical implementation viable. However, economy cannot claim to be self-sustainable, and perspectives from other disciplines can foster improvements in ICT security when organizations accept their economic performance critically depends on it.

The model unifies results from primary research with sources whose scope was purposefully not limited to academia, but included practical research published without peer review scrutiny. The author is of the opinion this did not devalue the results in any way, and inclusion of information obtained during controlled testing with reproducible results is beneficial regardless of how and where it was published. Penetration testing community in particular prefers alternative means of disseminating important findings, e.g., blogs, conferences, case studies, reports, white papers, workshops, etc. which are more flexible than academic publishing and allow to reach interested parties faster.

Further benefits are of financial metrics of key ICT security aspects. The author believes this could help organizations establish basis for comparison. Chapter 2.1.5 mentioned that some efforts have been made in economic quantification of security, e.g., return on information security investment (ROISI). The ICT governance model does not aim for cross-industry acceptance, but urges the CISO and IT personnel to consider the scope of the attack surface, how it can be reduced, and what measures are economically justifiable to deploy to improve security of users and ICT infrastructure. The majority of businesses need simple and effective solutions and policies which rely on users primarily as recipients of the services hardware and software can provide. Nevertheless, long-term training programs and practical experience are equally important because they ensure employees are involved in security to some extent.

User-side and ICT-side aspects of ICT security governance were introduced. User-side features include anti-social engineering techniques, suspicion-based behavior, password management together with training and real-life demonstrations. ICT-side elements are BYOD management, password composition requirements, logical and physical data separation, patch management and deployment, automated insider threat detection and containment, incident response, internal network segmentation, and ICT infrastructure hardening. Moreover, the following stimuli from outside the organizational boundary were considered: insider and outsider attacks; new software versions, patches; novel attack vectors and threats; malware, social engineering; and third-party electronic data interchange. They constitute avenues which may cause instabilities and disruptions of mission-critical services, or allow the adversary unauthorized access to sensitive data. It was also stated security depends on cooperation, communication, and interplay of users, IT personnel, CISO, management, hardware, software, and processes.

The following chapters summarize results, state main contributions of the thesis for theory and practice, hint at future research directions, and formulate concluding remarks.

7 RESULTS SUMMARY

Chapter 3.1 presented one main and two auxiliary goals. Their overview is as follows:

- main goal: strengthen the organizational sensitive electronic data and ICT security processes by addressing selected cybernetic risks and techniques for unauthorized access tied to mobile and other devices interacting with the ICT infrastructures and accessing electronic assets by devising a model and introducing best practices applicable to real-world, practical conditions
- first auxiliary goal: a collection of best practices and recommendations understandable to users with little to no security background,
- second auxiliary goal: a comprehensive report detailing shortcomings found during the testing which will form a basis for the ICT security governance model.

This chapter briefly summarizes findings from the previous two chapters, whether the goals were successfully achieved as well as caveats found during the research phase. For succinctness, results will be commented on only briefly, and the majority of details omitted. These can be found in chapters 4, 5, and 6 for the first auxiliary, second auxiliary, and main goal, respectively.

Questionnaires polled a cross section of general population for opinions, views, and real-world practices pertaining to various aspects of ICT and mobile security. The results are representative of the dichotomy between how individuals should approach and actually approach technology in everyday situations. An assumption that the same behavior uncovered during the investigation is present in organizations was made, and the author is of the opinion unless policies are in place mandating otherwise, human factor does not differentiate between approach to security in personal and work space. The findings suggest there is much to be improved, but enforcing normative security measures will do little to change ingrained attitudes, and user comfort must always be taken into account when devising ICT policies. Despite being mostly fluent in fundamental terminology, many respondents do not uphold baseline security practices and strongly prefer convenience. Employing a sample size of 784 participants, the outcomes are statistically significant and inferences can be drawn about the whole user population.

Case studies posited two questions: “How effectively can attacker possessing low to moderate ICT security knowledge reverse engineer password digests using trivial methods with pre-defined rules?” and “What relevant and usable data about the target’s ICT infrastructure can an attacker who possesses low to moderate ICT security gain using freely-available software?” A low-skilled adversary with little to no knowledge of the underlying technical concepts can make significant progress and obtain sensitive data about users and target organization of her choosing. Insider threat where a malicious trusted party aims to expropriate sensitive electronic assets or damage the company was considered as well, and by utilizing original research it was concluded employees are inherently treated as harmless, granted excessive permissions, allowed access to resources not based on what they really need but on what they might need, significantly expanding the attack surface.

The first auxiliary goal was **successfully accomplished** by means of questionnaire poll. Solutions were devised in the proposed ICT governance model which can be clearly communicated to users with little ICT background. As was reiterated several times throughout the thesis, automation, decoupling people from security to the highest extent possible, and hiding low-level details is crucial. The concept of security as a service is a promising avenue which aims to reduce security-related incidents by taking away the responsibility from end users and placing it on professionals. A synergy between computer science, ergonomics, and human/machine interaction may result in an acceptable compromise supporting security as a service. The practices are

presented in no particular order; when assembled, they can clearly communicate the findings and suggest improvements for the target group. Therefore, the goal was accomplished.

The second auxiliary goal was **successfully accomplished** by means of case studies. Being the longest part of the thesis, objective and correct test execution along with reproducible procedures were used: given identical data, software, and methods, anyone can verify reliability and validity of the conclusions. The first case study uncovered substandard security habits when it comes to selection of passwords, the most commonly employed means of authentication currently in existence. A real-world database of credentials was selected for analysis and severe omissions were found. The passwords were prone to naïve enumeration techniques which require little sophistication and result in a success rate of over 50% using commercially-available hardware. Furthermore, shortcomings were discovered which allow the adversary to predict most probable sequences and exploit them under realistic assumptions. With widespread reuse, the ripple effect and economic repercussions of a password breach could be substantial not only for the victim, but for organizations whose systems may become susceptible to compromise should the same credentials be reused and no ICT policies are enforced. The second case study demonstrated how malicious insiders can trivially fingerprint resources on internal network, impersonate legitimate user accounts, and collect information exploitable in freely-available software. For example: in case of the Faculty of Management and Economics, current password assignment policy allows to hijack arbitrary account in several hours supposing the default string was not changed. Coupled with administrative set of permissions each user is granted, a single weakness endangers the network while few countermeasures are in place which address the unfavorable predicament. The case studies present comprehensively describe methodology, details of test scenarios, results, and solutions. Therefore, the goal was accomplished.

The main goal was **successfully accomplished** by means of a model conceptualizing the methodology supported by output of the auxiliary goals. Interrelations between actors in ICT process management (users, IT personnel, Chief Information Security Officer, external stimuli) were mapped, and factors influencing users and IT personnel identified. Economic metrics were also presented which help quantify the model's benefits. Questionnaire research established the basis for user-side metrics which emphasized resilience toward psychological manipulation and social engineering, streamlined password management, training and real-world demonstrations. The author considers these areas underdeveloped and in need of improvement. Case studies established the basis for ICT-side metrics which emphasized bring your own device (BYOD) management, infrastructure hardening, incident response, password composition policies, and sensitive data separation. Mobile management in particular was pointed as dynamically developing and necessitating dedicated policies. The proposed ICT model builds on decoupling users from security decisions to the highest extent possible. Instead, accessible front-ends which hide low-level details and are viewed as a service the users invoke on demand need should be preferred. The three key agents in the model (adversaries, ICT personnel, users) were extensively covered, and their interactions visualized and analyzed. Therefore, the goal was accomplished.

8 CONTRIBUTIONS OF THE THESIS

The contributions are divided into theoretical and practical parts. The author is of the opinion improvements to the theoretical background together with practical implications have resulted in several noteworthy contributions which mainly tend toward practice.

8.1 Science and Theory

The main theoretical contributions of the thesis are identification of factors influencing organizational ICT security process governance from the point of view of users, integrating knowledge from seemingly disparate disciplines, i.e., information technology and economics into a cohesive unit, and devising economic indicators linking the two together. ICT-side security was investigated by means of adversarial thinking which brought a novel perspective into infrastructure resilience modeling that should be increasingly focusing on black-box and gray-box testing methodologies to comprehensively map the threat surface. Penetration testing and vulnerability assessment can enrich complexity theory by uncovering relations between systems usually treated separately, and demonstrating how business continuity depends on pitfalls of theoretical concepts put into practice.

The literature review purposefully incorporated sources outside academia and synthesized them (chapter 2.4) which was supposed to show the dynamism information technology offers to organizations. Future research can inspect applicability of theory in practical situations, and its development by parties not subjected to formal peer review process. In the author's opinion, lack of academic scrutiny should not constitute reason for disregarding promising research venues. Acknowledging sources outside of main academic venues was valuable because theoretical aspects of information technology should be closely tied with their practical value and applicability in real-world settings.

Among the scientific contributions of the thesis, integration of concepts from ergonomics, information technology, psychology, and economics when devising policies acceptable to employees opens new avenues of research and enables organizations to modify their approach when selecting and implementing security measures. Original research and primary data which led to verification of statistical hypotheses contributes to understanding of the phenomenon in which individuals cooperate to achieve synergic effects. Furthermore, presenting an up-to-date questionnaire survey results have implications for social science theory as well as media communications: it is postulated respondents were mainly influenced by recommendations from audiovisual, printed, and online sources when devising their passwords. By modifying the message content to better reflect the realities of ICT security, the general population could be influenced to change their behavior indirectly.

Because organizational ICT security processes incorporate human element, another scientific contribution is prediction of response patterns when faced with unknown situations in adversarial settings, e.g., social engineering campaigns. Cognitive psychology may benefit from the results, and further research in the area may lead to a model of how human behavior changes under psychological duress facilitated by technology, and how trust toward ICT shifts with prior negative experiences. This may lead to crisis management plans which address concerns of employees who have become victims of an attack.

In terms of educational benefits, curricula on economic faculties could be modified to include ergonomics, psychology, and how they affect each other in Internet-enabled global societies. Subjects pertaining to information technology could benefit from adversarial thinking, real-world

testing, and system design whose goal is to devise resilient infrastructure impervious to a given attack vector. Experiential learning could improve understanding of complex system relations in practical scenarios confronted with their theoretical foundations.

8.2 Practice

Primary contributions of the thesis lean toward practice due to the data gathered from users about various aspects of ICT which have become ubiquitous in everyday situations. Case studies contributed to the results by faithfully representing abilities of a low-skilled adversary who can launch potent attack scenarios.

Adversarial password security enumeration asserted a tendency of users to prefer convenience, simplicity, and streamlining non-essential activities. Moreover, system administrators tend to treat all parties within the organizational boundary as inherently trusted. The assumption was proven detrimental to security and substantiated by primary data. This should lead to revision of ICT policies which must treat users as unreliable.

The main contribution is the model of ICT security process governance. Synthesizing findings from questionnaire research and case studies, the concept explicitly accentuates insider threat as a legitimate attack vector, and advises IT personnel to harden internal infrastructure so that attempts of unauthenticated parties to access sensitive electronic assets are prevented outright, or detected at the origin. Human factor was identified as the weakest link of security and a curriculum outlined which practically demonstrates preventative techniques against social engineering, a vector the adversary prioritizes when other vulnerabilities are overly taxing to exploit. Long-term training and testing the acquired knowledge in controlled scenarios are important for the psychological change the organizations should strive for in their employees.

Mobile management (BYOD) was discussed and deemed a significant trend in organizations, albeit one which has no backing in ICT policies. Adversarial thinking backed by relevant sources and data suggested attacks can be executed using minimal resources, and centralized device management via profiles was put forward as a solution. However, questionnaire research found sentiment of users regarding profiles is polarized, and open communication together with employee education was accentuated. Mobile devices bring about challenges and opportunities, and the model explicitly accepted and analyzed both.

Continuously stressed in the model was the notion of decoupling users from security to the highest extent possible. Arguments can be raised this relegates them to the position of recipients rather than participants in the ICT security governance process. However, system complexity has been steadily growing and broad expertise is required to actively partake in the ICT design process. Abstracting from implementation details and presenting users with security as a service was corroborated by evidence in case study 1 and is believed to represent a prudent course of action from the security standpoint. Centrally-administered enterprise credentials management and workstation privilege deescalation were specifically mentioned as starting points for improving security posture. Break-even point analysis was presented for a selected organization and economic impact metrics suggested.

9 FUTURE RESEARCH DIRECTIONS

Future research directions were hinted at throughout the thesis are aggregated here. Several promising avenues exist which expand some of the topics and may result in tighter integration of security and economics. The first pertains to security change management and whether its specific nature affects the methods to instill positive attitude in employees. Tied to psychology, the matter was not investigated further due to author's limited knowledge in the area.

A small-scale experiment with alternative means of authentication, e.g., mobile applications and biometric identification tokens may give organizations insights into whether they are preferred over passwords, change in login times and frequency, and their acceptance rates. Penetration testing could uncover exploitable vectors when such means are deployed on the premises which may lead to iterative improvements of a system the organization may decide to eventually deploy on a permanent basis.

Strong data encryption and defense in depth with associated processing overheads and performance degradation under varying loads would be interesting to analyze for mitigating insider threat. The author is of the opinion hardware infrastructure would not be severely taxed for data at rest, but may exhibit spikes when processing frequently-accessed ("live") records. Analyzing the optimum between security and usability could go further: hosting a centralized database of user credentials along with the willingness of individuals to accept longer delays for increased security could be tested using sliders where the extremes would represent speed and security, and the user would select the desired balance. Aggregating the data, a pool of preferences could be created and offered as a default choice in other security-related decisions.

Over long term, more precise economic metrics could be set. With a tendency of some organizations to migrate into the cloud, the risk shifts and the policies of data center operators need to be factored in as well. This brings about novel challenges when calculating cost per lost entry of sensitive data as a breach in a remote location may see the data expropriated without forewarning. Even though a list of factors in the formula which quantifies the economic impact of lost records can be assembled, its construction is a research task which necessitates extensive real-world data and cross-industry cooperation.

10 CONCLUSION

Effective information processing has become the enabler of competitiveness, although people expect information and communication technology to simply work and provide them with a service, much like other public utilities. However, being a complex interconnected system, synergy of technology, people, and processes is needed to support and foster efficient business operations. Features of successful ICT governance are high availability, performance, stability, service orientation, hiding the technical details from users and enable them to benefit from intuitive, simple front-end functionality. To do that, the infrastructure must be administered and maintained in accordance with policies which stem from and respect organizational process map. Security is a pervasive feature of the arrangement, and must be treated as such given the globally-interconnected world has been increasingly dependent on technology and much data is now available exclusively in electronic form.

It is the author's opinion security is disregarded because it is challenging to quantify: unlike in accounting and financial management, there are no widely-accepted indicators, and multiple unstandardized implementations exist. Consensus is yet to be determined with respect to economic metrics for measuring efficiency of security, and while attempts have been made to introduce suitable indicators, organizations are not legally bound to include them in balance sheets and cash flow statements. The thesis did not aim to produce the consensus, but argues that it will involve many aspects from both ICT and economics. Presenting a thorough literature review of sources from within and outside academia, security was demonstrated theoretically and practically as a multifaceted approach which dynamically develops and integrates findings from a broad range of disciplines. While no person, natural or juridical, can claim to have "perfect" or "complete" security, these claims regularly appear in practice which state a "perfect" solution has been found simply because no objective baseline of evaluation has been set. As each individual and organization is unique, no ultimate risk-mitigating panacea exists.

The thesis sometimes used terms "may," "could," and "should" instead of the stronger "will" and "must" because the sources gave few guarantees regarding the effect any recommendation might have outside of controlled conditions. What works for one organization is ineffective for others, and opposing views exist even on fundamental security aspects: external audits, long-term employee education, and deploying black-box solutions. The cost of misconfiguration uncovered and exploited by a malicious party may prove higher than the investment into security, though.

A challenge of how to quantify returns on security investments lies in the cost per lost record, particularly when loss of confidence, customer attrition, and reputation management are involved. Even though the thesis hinted at a remedy, categorizing information into classes should constitute the first step. Economic risks of insider threat were pointed out: treating each employee as potentially malicious and implementing appropriate measures to reduce the threat surface were proposed as a solution. Automated detection logic must give IT personnel early warning signs, assuming the attacker is insufficiently skilled and triggers them. The thesis extensively covered adversarial modeling, too.

Questionnaire research and two case studies were presented to evaluate hypotheses which state security mindset is not prevalent in a representative sample of users, and security engineering is not systematically practiced in organizations, respectively. Building on the results, the ICT security process governance model was devised which synthesizes findings from literature review and original research into a cohesive unit, and supplies the Chief Information Security Officer and managers with a framework addressing selected ICT security aspects. The proposed economic metrics form a link between security and economics, and quantify effects of coordinated efforts

put into long-term employee training, credentials management, infrastructure hardening, incident response, and other factors.

The aim of the thesis was to show organizations need and depend on security economically, and that human factor must be accepted as playing a vital role. Denying these realities will result in negative consequences and costs disproportionate to investments into technology, people, and processes, each of which can become a bottleneck and a weakness if disregarded in favor of the others. Achieving synergy through anticipation of developments, reflecting them in policies, and practicing proactive, positive stance toward the inevitable technological change are thus the differentiating features of modern organizations.

References

Monographs

- Allen, L. (2012). *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide*. Birmingham: Packt Publishing.
- Anderson, R. J. (2008). *Security Engineering: A Guide to Building Dependable Distributed Systems* (2nd ed.). New Jersey: Wiley.
- Andress, J. (2011). *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*. Waltham: Syngress.
- Arnold, R. A. (2008). *Economics* (9th ed.). Mason: Cengage Learning.
- Bagley, P. R. (1968). *Extension of Programming Language Concepts*. Philadelphia: University City Science Center.
- Barlow, R. E. & Proschan, F. (1981). *Statistical Theory of Reliability and Life Testing: Probability Models* (2nd ed.). Silver Spring: To Begin With.
- Bernstein, P. A., Hadzilacos, V., & Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Boston: Addison-Wesley.
- Besanko, D. & Braeutigam, R. (2010). *Microeconomics* (4th ed.). New Jersey: Wiley.
- Boiko, B. (2004). *Content Management Bible* (2nd ed.). New Jersey: Wiley.
- Coronel, C., Morris, S., & Rob, P. (2009). *Database Systems: Design, Implementation and Management* (9th ed.). Mason: Cengage Learning.
- EC-Council. (2010). *Penetration Testing: Procedures & Methodologies*. Mason: Cengage Learning.
- Creswell, J. W. (2002). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (2nd ed.). Los Angeles: SAGE Publications.
- Davenport, T. H. (1992). *Process Innovation: Reengineering Work Through Information Technology*. Boston: Harvard Business Review Press.
- Davenport, T. H. & Prusak, L. (2000). *Working Knowledge: How Organizations Manage What They Know* (2nd ed.). Boston: Harvard Business Review Press.
- Dooley, K. (2001). *Designing Large-Scale LANs*. Sebastopol: O'Reilly Media.
- Drucker, P. (1996). *Landmarks of Tomorrow: A Report on the New Post-Modern World*. New Jersey: Transaction Publishers.
- Engelbreton, P. (2011). *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy* (2nd ed.). Waltham: Syngress.
- Freeman, R. E. (2010). *Strategic Management: A Stakeholder Approach*. Cambridge: Cambridge University Press.
- Friedl, J. E. F. (2002). *Mastering Regular Expressions* (2nd ed.). Sebastopol: O'Reilly.
- Giarratano, J. C. & Riley, G. D. (2004). *Expert Systems: Principles and Programming* (4th ed.). Mason: Course Technology.
- Goldreich, O. (2004). *Volume II: Basic Applications*. The Foundations of Cryptography. Cambridge: Cambridge University Press.
- Gollmann, D. (2011). *Computer Security* (3rd ed.). New Jersey: Wiley.
- Gordon, L. & Loeb, M. (2005). *Managing Cybersecurity Resources: A Cost-Benefit Analysis*. New York: McGraw-Hill.
- Hadnagy, C. (2010). *Social Engineering: The Art of Human Hacking*. New Jersey: Wiley.
- Hammer, M. & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. New York: Harper Business.
- Harris, S. (2012). *CISSP All-in-One Exam Guide* (6th ed.). New York: McGraw-Hill.

- Hayden, L. (2010). *IT Security Metrics: A Practical Framework for Measuring Security & Protecting Data*. New York: McGraw-Hill.
- IEEE. (1990). *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York: The Institute of Electrical and Electronics Engineering.
- Ihara, S. (1993). *Information Theory for Continuous Systems*. Singapore: World Scientific.
- Jaquith, A. (2007). *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Boston: Addison-Wesley.
- Johansson, H. J., McHugh, P., Pendlebury, A. J., & Wheeler, W. A. (1993). *Business Process Reengineering: Breakpoint Strategies for Market Dominance*. New Jersey: Wiley.
- Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*. New Jersey: Wiley.
- Knight, F. H. (1921). *Risk, Uncertainty, and Profit*. Boston: Hart, Schaffner & Marx.
- Lécuyer, C. & Brock, D. C. (2010). *Makers of the Microchip: A Documentary History of Fairchild Semiconductor*. Cambridge: The MIT Press.
- Levitin, A. (2011). *Introduction to the Design and Analysis of Algorithms* (3rd ed.). Boston: Addison-Wesley.
- Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic Theory*. Oxford: Oxford University Press.
- Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. Boca Raton: CRC Press.
- Merriam, S. B. (2009). *Qualitative Research: A Guide to Design and Implementation* (3rd ed.). San Francisco: Jossey-Bass.
- Mirkovic, J., Dietrich, S., Dittrich, D., & Relher, P. (2005). *Internet Denial of Service: Attack and Defense Mechanisms*. New Jersey: Prentice Hall.
- Mitchell, C. (2005). *Trusted Computing*. London: Institution of Engineering and Technology.
- Mitnick, K. D. & Simon, W. L. (2003). *The Art of Deception: Controlling the Human Element of Security*. New Jersey: Wiley.
- NISO. (2004). *Understanding Metadata*. Baltimore: NISO Press.
- Norris, J. R. (1998). *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press.
- Onn, Y., Druckman, Y., Timor, R., Maroun, A., Nachmani, Y., Sicklai, S., . . . Packer, S. (2005). *Privacy in the Digital Environment*. Haifa: Haifa Center of Law and Technology.
- Osborne, M. J. & Rubinstein, A. (1994). *A Course in Game Theory*. Cambridge: MIT Press.
- Oscarson, P. (2007). *Actual and Perceived Information Systems Security* (Doctoral Thesis, Södertörn University, Stockholm, Sweden). Retrieved 2013-06-08, from <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-10215>
- Oxford University Press. (2011). *Oxford Advanced American Dictionary for learners of English*. New York: Oxford University Press.
- Paar, C. & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin: Springer.
- Parker, D. B. (1998). *Fighting Computer Crime: A New Framework for Protecting Information*. New Jersey: Wiley.
- Parker, M. M., Benson, R. J., & Trainor, H. (1988). *Information Economics: Linking Business Performance to Information Technology*. New Jersey: Prentice Hall.
- Porter, M. E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: The Free Press.

- Rotini, F., Borgianni, Y., & Cascini, G. (2012). *Re-engineering of Products and Processes: How to Achieve Global Success in the Changing Marketplace*. Springer Series in Advanced Manufacturing. London: Springer-Verlag. doi:10.1007/978-1-4471-4017-7_1
- Savage, L. J. (1972). *The Foundations of Statistics*. New York: Dover Publications.
- Schneier, B. (1996). *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd ed.). New Jersey: Wiley.
- Schneier, B. (2000). *Secrets & Lies: Digital Security in a Networked World*. New Jersey: Wiley.
- Shannon, C. E. (1956). *Automata Studies*. New Jersey: Princeton University Press.
- Sheskin, D. J. (2004). *Handbook of Parametric and Nonparametric Statistical Procedures* (3rd ed.). Boca Raton: CRC Press.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2012). *Operating System Concepts* (9th ed.). New Jersey: Wiley.
- Simon, H. A. (1957). *Models of Man: Social and Rational*. New York: Wiley.
- Stevens, M. [M.M.J.]. (2007). *On Collisions For MD5* (Master's Thesis, Eindhoven University of Technology, Eindhoven, Netherlands). Retrieved 2013-08-07, from <http://www.win.tue.nl/hashclash/On%20Collisions%20for%20MD5%20-%20M.M.J.Stevens.pdf>
- Taleb, N. N. (2010). *The Black Swan: The Impact of the Highly Improbable* (2nd ed.). New York: Dark Horse.
- Venn, J. (1866). *The Logic of Chance: An Essay on the Foundations and Province of the Theory of Probability, with Especial Reference to Its Application to Moral and Social Science*. London: Macmillan.
- von Foerster, H. (2003). *Understanding Understanding: Essays on Cybernetics and Cognition*. New York: Springer-Verlag.
- Wang, J. (2009). *Computer Network Security: Theory and Practice*. Beijing: Higher Education Press.
- Weikum, G. & Vossen, G. (2002). *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery*. London: Morgan Kaufmann Publishers.
- Wiener, N. (1965). *Cybernetics: Or Control and Communication in the Animal and the Machine* (2nd ed.). Cambridge: The MIT Press.
- Wu, X. (2007). *Security Architecture for Sensitive Information Systems* (Doctoral Thesis, Monash University, Melbourne, Australia). Retrieved 2013-06-08, from http://www.csse.monash.edu.au/~srini/theses/Ping_Thesis.pdf
- Yin, R. K. (2008). *Case Study Research: Design and Methods* (4th ed.). Los Angeles: SAGE Publications.

Articles, Chapters

- Ackoff, R. L. (1989). From Data to Wisdom. *Journal of Applied Systems Analysis*, 16.
- Adams, A. & Sasse, M. A. (1999). Users Are Not The Enemy. *Communications of the ACM*, 42, 40–46. doi:10.1145/322796.322806
- Aldhaban, F. (2012). Exploring the Adoption of Smartphone Technology: Literature Review. In *2012 Proceedings of PICMET '12: Technology Management for Emerging Technologies* (pp. 2758–2770). IEEE.
- Alipour, M., Kord, B., & Tofighi, E. (2011). A Study of Different Types of Business Risks and Their Effects on Banks' Outsourcing Process (Case Study: Tejarat Bank in Iran). *International Journal of Business and Social Science*, 2, 227–237. doi:10.1201/1078/44432.21.3.20040601/82471.2

- Amico, M. D., Michiardi, P., & Roudier, Y. (2010). Password Strength: An Empirical Analysis. In *Proceedings of the 2010 IEEE INFOCOM* (pp. 1–9). IEEE. doi:10.1109/INFOCOM.2010.5461951
- Anderson, R. (2004). Cryptography and Competition Policy – Issues with ‘Trusted Computing’. *Advances in Information Security*. Economics of Information Security, *12*, 35–52. doi:10.1007/1-4020-8090-5_3
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., . . . Zaharia, M. (2010). A View of Cloud Computing. *Communications of the ACM*, *53*, 50–58. doi:10.1145/1721654.1721672
- Aron, R., Clemons, E. K., & Reddi, S. (2005). Just Right Outsourcing: Understanding and Managing Risk. In *Proceedings of the 38th Hawaii International Conference on System Sciences* (pp. 214–224). doi:10.1109/HICSS.2005.368
- Ballagas, R., Borchers, J., Rohs, M., & Sheridan, J. G. (2006). The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, *5*, 70–77. doi:10.1109/MPRV.2006.18
- Bennett, S. C. (2010). Ethics of “Pretexting” in a Cyber World. *McGeorge Law Review*, *41*, 271–279.
- Bezuidenhout, M., Mouton, F., & Venter, H. (2010). Social Engineering Attack Detection Model: SEADM. In *Proceedings of Information Security for South Africa (ISSA), 2010* (pp. 1–8). IEEE. doi:10.1109/ISSA.2010.5588500
- Biggam, J. (2001). Defining Knowledge: An Epistemological Foundation for Knowledge Management. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (pp. 1–7). doi:10.1109/HICSS.2001.927102
- Bishop, M. (1991). Password Management. In *Digest of Papers COMPCON Spring '91* (pp. 167–169). IEEE. doi:10.1109/COMPCON.1991.128801
- Bretherton, F. P. & Stingley, P. T. (1994). Metadata: A User’s View. In *Proceedings of the 7th International Working Conference on Scientific and Statistical Database Management* (pp. 51–59). 7th International Working Conference on Scientific and Statistical Database Management. doi:10.1109/SSDM.1994.336950
- Brewer, E. A. (2000). Towards Robust Distributed Systems. Association for Computing Machinery. 19th ACM Symposium on Principles of Distributed Computing. Retrieved 2013-06-13, from <http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>
- Buhari, M. I., Habaebi, M. H., & Ali, B. M. (2005). Artificial Neural System for Packet Filtering. *Journal of Computer Science*, *1*, 259–269. doi:10.3844/jcssp.2005.259.269
- Bursztein, E. & Bethard, S. (2009). Decaptcha: Breaking 75% of eBay Audio CAPTCHAs. In *Proceedings of the 3rd USENIX Conference on Offensive Technologies* (pp. 1–7). USENIX Association.
- Burt, B. A. (2001). Definitions of Risk. *Journal of Dental Education*, *65*, 1007–1008.
- Buss, D. M. (1987). Selection, Evocation, and Manipulation. *Journal of Personality and Social Psychology*, *53*, 1214–1221. doi:10.1037/0022-3514.53.6.1214
- Carl, G., Kesidis, G., Brooks, R. R., & Rai, S. (2006). Denial-of-Service Attack-Detection Techniques. *IEEE Internet Computing*, *10*, 82–89. doi:10.1109/MIC.2006.5
- Cash, G. L. & Hatamian, M. (1987). Optical Character Recognition by the Method of Moments. *Computer Vision, Graphics, and Image Processing*, *39*, 291–310. doi:10.1016/S0734-189X(87)80183-4
- Cavusoglu, H., Cavusoglu, H., & Zhang, J. (2008). Security Patch Management: Share the Burden or Share the Damage? *Management Science*, *54*, 657–670. doi:10.1287/mnsc.1070.0794
- Chakrabarti, S. & Singhal, M. (2007). Password-based Authentication: Preventing Dictionary Attacks. *Computer*, *40*, 68–74. doi:10.1109/MC.2007.216

- Chamberlin, R. D. & Boyce, R. F. (1974). Sequel: A Structured English Query Language. In *Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control* (pp. 249–264). Association for Computing Machinery. doi:10.1145/800296.811515
- Chang, R. K. C. (2002). Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. *IEEE Communications Magazine*, 40, 42–51. doi:10.1109/MCOM.2002.1039856
- Chellapilla, K. & Simard, P. Y. (2004). Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). *Advances in Neural Information Processing Systems*, 17, 265–272.
- Chellapilla, K. & Simard, P. Y. (2008). Breaking Audio CAPTCHAs. *Advances in Neural Information Processing Systems*, 21, 1625–1632.
- Chen, E. Y. & Itoh, M. (2010). Virtual Smartphone over IP. In *Proceedings of the 2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks* (pp. 1–6). IEEE. doi:10.1109/WOWMOM.2010.5534992
- Chen, P. M. & Noble, B. D. (2001). When Virtual is Better Than Real. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems* (pp. 133–138). IEEE. doi:10.1109/HOTOS.2001.990073
- Chew, M. & Tygar, J. D. (2004). Image Recognition CAPTCHAs. In *Information Security: Proceedings 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27-29, 2004* (pp. 268–279). Berlin: Springer Verlag. doi:10.1007/978-3-540-30144-8_23
- Chin, E., Felt, A. P., Sekar, V., & Wagner, D. (2012). Measuring User Confidence in Smartphone Security and Privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (pp. 1–16). Association for Computing Machinery. doi:10.1145/2335356.2335358
- Christodorescu, M., Sailer, R., Schales, D. L., Sgandurra, D., & Zamboni, D. (2009). Cloud Security is not (Just) Virtualization Security: A Short Paper. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security* (pp. 97–102). Association of Computing Machinery. doi:10.1145/1655008.1655022
- Cleghorn, L. (2013). Network Defense Methodology: A Comparison of Defense in Depth and Defense in Breadth. *Journal of Information Security*, 4, 144–149. doi:10.4236/jis.2013.43017
- Clemons, E. K., Reddi, S. P., & Row, M. C. (1993). The Impact of Information Technology on the Organization of Economic Activity: The “Move to the Middle” Hypothesis. *Journal of Management Information Systems*, 10, 9–35.
- Coase, R. H. (1937). The Nature of the Firm. *Economica*, 4, 386–405.
- Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13, 377–387. doi:10.1145/362384.362685
- Cohen, F. (1987). A Cryptographic Checksum for Integrity Protection. *Computers & Security*, 6, 505–510. doi:10.1016/0167-4048(87)90031-9
- Constandache, I., Gaonkar, S., Saylor, M., Choudhury, R. R., & Cox, L. (2010). EnLoc: Energy-Efficient Localization for Mobile Phones. In *Proceedings of the 28th IEEE International Conference on Computer Communications Workshops* (pp. 2716–2720). IEEE. doi:10.1109/INFCOM.2009.5062218
- Copeland, R. & Crespi, N. (2012). Analyzing Consumerization – Should Enterprise Business Context Determine Session Policy? In *Proceedings of the 2012 16th International Conference on Intelligence in Next Generation Networks* (pp. 187–193). IEEE. doi:10.1109/ICIN.2012.6376024
- Dagon, D., Martin, T., & Starner, T. (2004). Mobile Phones as Computing Devices: The Viruses are Coming! *IEEE Pervasive Computing*, 3, 11–15. doi:10.1109/MPRV.2004.21

- Dali, A. & Lajtha, C. (2012). ISO 31000 Risk Management – “The Gold Standard”. *EDPACS: The EDP Audit, Control, and Security Newsletter*, 45, 1–8. doi:10.1080/07366981.2012.682494
- Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully Completing Case Study Research: Combining Rigour, Relevance and Pragmatism. *Information Systems Journal*, 8, 273–289. doi:10.1046/j.1365-2575.1998.00040.x
- Datta, R., Li, J., & Wang, J. Z. (2005). Imagination: A Robust Image-Based CAPTCHA Generation System. In *Proceedings of the 13th Annual ACM International Conference on Multimedia* (pp. 331–334). Association of Computing Machinery. doi:10.1145/1101149.1101218
- Denning, D. E. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, SE-13, 222–232. doi:10.1109/TSE.1987.232894
- Denzin, N. K. & Lincoln, Y. S. (2005). Introduction: The Discipline and Practice of Qualitative Research. In N. K. Denzin & Y. S. Lincoln (Eds.), *The SAGE Handbook of Qualitative Research* (3rd ed.). Los Angeles: SAGE Publications.
- Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why Phishing Works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 581–590). Association for Computing Machinery. doi:10.1145/1124772.1124861
- Dhillon, G. & Backhouse, J. (2001). Current Perspectives in IS Security Research: Towards Socio-Organizational Perspectives. *Information Systems Journal*, 11, 127–153. doi:10.1046/j.1365-2575.2001.00099.x
- Dong, H., Hao, Q., Zhang, T., & Zhang, B. (2010). Formal Discussion on Relationship Between Virtualization and Cloud Computing. In *Proceedings of the 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (pp. 448–453). doi:10.1109/PDCAT.2010.41
- Earl, M. & Khan, B. (1994). How New is Business Process Redesign? *European Management Journal*, 12, 20–30. doi:10.1016/0263-2373(94)90043-4
- Egelman, S., Sotirakopoulos, A., Musluhkov, I., Beznosov, K., & Herley, C. (2013). Does My Password Go up to Eleven? The Impact of Password Meters on Password Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2379–2388). Association for Computing Machinery. doi:10.1145/2470654.2481329
- Elerath, J. G. (2000). Specifying Reliability in the Disk Drive Industry: No More MTBF’s. In *Proceedings of the Annual Reliability and Maintainability Symposium* (pp. 194–199). IEEE. doi:10.1109/RAMS.2000.816306
- Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., & Estrin, D. (2010). Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (pp. 179–194). Association for Computing Machinery. doi:10.1145/1814433.1814453
- Al-Fares, M., Loukissas, A., & Vahdat, A. (2008). A Scalable, Commodity Data Center Network Architecture. In *Proceedings of SIGCOMM 2008* (pp. 63–74). doi:10.1145/1402946.1402967
- Feistel, H. (1973). Cryptography and Computer Privacy. *Scientific American*, 228, 15–23.
- Firesmith, D. (2004). Specifying Reuseable Security Requirements. *Journal of Object Technology*, 3, 61–75. Retrieved 2013-06-10, from http://www.jot.fm/issues/issue_2004_01/column6.pdf
- Florêncio, D. & Herley, C. (2007). A Large-Scale Study of Web Password Habits. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 657–666). Association for Computing Machinery. doi:10.1145/1242572.1242661
- Fluhrer, S., Martin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In *Selected Areas in Cryptography: Revised Papers 8th Annual International*

- Workshop, SAC 2001 Toronto, Ontario, Canada, August 16–17, 2001 (pp. 1–24). Berlin: Springer Verlag. doi:10.1007/3-540-45537-X_1
- Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, 12, 219–245. doi:10.1177/1077800405284363
- Forer, B. F. (1949). The Fallacy of Personal Validation: A Classroom Demonstration of Gullibility. *Journal of Abnormal and Social Psychology*, 44, 118–123. doi:10.1037/h0059240
- Friedewald, M. & Raabe, O. (2011). Ubiquitous Computing: An Overview of Technology Impacts. *Telematics and Informatics*, 28, 55–65. doi:10.1016/j.tele.2010.09.001
- Fritsch, C., Netter, M., Reisser, A., & Pernul, G. (2010). Attacking Image Recognition CAPTCHAs: A Naive but Effective Approach. In *Trust, Privacy and Security in Digital Business: Proceedings 7th International Conference, TrustBus 2010, Bilbao, Spain, August 30-31, 2010* (pp. 13–25). Berlin: Springer Verlag. doi:10.1007/978-3-642-15152-1_2
- Geambasu, R., Kohno, T., Levy, A. A., & Levy, H. M. (2009). Vanish: Increasing Data Privacy with Self-Destructing Data. In *Proceedings of the 18th USENIX Security Symposium*. USENIX Association.
- Gessner, D., Girao, J., Karame, G., & Li, W. (2013). Towards a User-Friendly Security-Enhancing BYOD Solution. *NEC Technical Journal*, 7, 113–116.
- Gilbert, S. & Lynch, N. (2002). Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. *ACM SIGACT News*, 33, 51–59.
- Gill, P., Jain, N., & Nagappan, N. (2011). Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications. In *Proceedings of SIGCOMM 2011 & Best Papers of the Co-Located Workshops* (pp. 350–361). doi:10.1145/2043164.2018477
- Gjerde, K. A. P., Slotnick, S. A., & Sobel, M. J. (2002). New Product Innovation with Multiple Features and Technology Constraints. *Management Science*, 48, 1268–1284.
- Goldberg, I., Wagner, D., Thomas, R., & Brewer, E. A. (1996). A Secure Environment for Untrusted Helper Applications: Confining the Wily Hacker. In *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography: Volume 6* (pp. 1–13). USENIX Association.
- Gray, J. (1981). The Transaction Concept: Virtues and Limitations. In *Proceedings of the 7th International Conference on Very Large Databases* (Vol. 7, pp. 144–154). 7th International Conference on Very Large Databases.
- Greenberg, A., Lahiri, P., Maltz, D. A., Patel, P., & Sengupta, S. (2008). Towards a Next Generation Data Center Architecture: Scalability and Commoditization. In *Proceedings of PRESTO’08 – Workshop on Programmable Routers for Extensible Services of Tomorrow* (pp. 57–62). doi:10.1145/1397718.1397732
- Gu, Y., Fu, Y., Prakash, A., Lin, Z., & Yin, H. (2012). Os-sommelier: Memory-Only Operating System Fingerprinting in the Cloud. In *Proceedings of the 3rd ACM Symposium on Cloud Computing* (pp. 1–13). Association of Computing Machinery. doi:10.1145/2391229.2391234
- Guo, C., Wang, H. J., & Zhu, W. (2004). Smart-Phone Attacks and Defenses. In *Proceedings of the Third Workshop on Hot Topics in Networks* (pp. 1–6). Association for Computing Machinery.
- Haerder, T. & Reuter, A. (1983). Principles of Transaction-Oriented Database Recovery. *ACM Computing Surveys*, 15, 287–317. doi:10.1145/289.291
- Halderman, J. A., Schoen, S. D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J. A., ... Felten, E. W. (2008). Lest We Remember: Cold Boot Attacks on Encryption Keys. In *Proceedings of the 17th USENIX Security Symposium* (pp. 45–60).

- Halfond, W. G., Viegas, J., & Orso, A. (2006). A Classification of SQL Injection Attacks and Countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (pp. 1–11). IEEE.
- Han, S., Qian, C., Leith, D., Mok, A. K., & Lam, S. S. (2011). Hartfi: An Energy-Efficient Localization System. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Green Networking* (pp. 25–30). IEEE. doi:10.1145/2018536.2018543
- Hansen, P. B. (1970). The Nucleus of Multiprogramming System. *Communications of the ACM*, 13, 238–241. doi:10.1145/362258.362278
- Hardin, G. (1968). The Tragedy of the Commons. *Science*, 162, 1243–1248. doi:10.1126/science.162.3859.1243
- Harries-Jones, P. (1988). The Self-Organizing Polity: An Epistemological Analysis of Political Life. *Canadian Journal of Political Science*, 21, 431–433.
- Harris, B. & Hunt, R. (1999). TCP/IP Security Threats and Attack Methods. *Computer Communications*, 22, 885–897. doi:10.1016/S0140-3664(99)00064-X
- Harrison, J. V. (2005). Enhancing Network Security By Preventing User-Initiated Malware Execution. In *Proceedings of the International Conference on Information Technology: Coding and Computing* (pp. 597–602). IEEE. doi:10.1109/ITCC.2005.146
- Henry, N. N. (1974). Knowledge Management: A New Concern for Public Administration. *Public Administration Review*, 34, 189–196.
- Herley, C. & van Orschot, P. C. (2012). A Research Agenda Acknowledging the Persistence of Passwords. *IEEE Security & Privacy*, 10, 28–36. doi:10.1109/MSP.2011.150
- Holton, G. A. (2004). Defining Risk. *Financial Analyst Journal*, 60, 19–25.
- Householder, A., Houle, K., & Dougherty, C. (2002). Computer Attack Trends Challenge Internet Security. *Computer*, 35, 5–7. doi:10.1109/MC.2002.1012422
- Hu, X.-Y., Eleftheriou, E., Haas, R., Iliadis, I., & Pletka, R. (2009). Write Amplification Analysis in Flash-Based Solid State Drives. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference* (pp. 1–9). Association for Computing Machinery. doi:10.1145/1534530.1534544
- Hughey, A. W. & Mussnug, K. J. (1966). Designing Effective Employee Training Programmes. *Training For Quality*, 5, 52–57. doi:10.1108/09684879710167638
- Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a Definition of Mixed Methods Research. *Journal of Mixed Methods Research*, 1, 112–133. doi:10.1177/1558689806298224
- Kam, J. B. & Davida, G. I. (1979). Structured Design of Substitution-Permutation Encryption Networks. *IEEE Transactions on Computers*, 28, 747–753.
- Karlson, A. K., Meyers, B. R., Jacobs, A., Johns, P., & Kane, S. K. (2009). Working Overtime: Patterns of Smartphone and PC Usage in the Day of an Information Worker. In *Proceedings of the 7th International Conference on Pervasive Computing* (pp. 398–405). doi:10.1007/978-3-642-01516-8_27
- Kaufman, L. M. (2009). Data Security in the World of Cloud Computing. *IEEE Security & Privacy*, 7, 61–64. doi:10.1109/MSP.2009.87
- Kelley, P. G., Komanduri, S., Mazurek, M. E., Shay, R., Vidas, T., Bauer, L., . . . López, J. (2012). Guess Again (And Again And Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy* (pp. 523–537). IEEE. doi:10.1109/SP.2012.38
- Kenworthy, G. & Rohatgi, P. (2012). Mobile Device Security: The Case for Side Channel Resistance. In *Proceedings of the Mobile Security Technologies 2012* (pp. 1–4). IEEE.
- Kerckhoffs, A. (1883). La Cryptographie Militaire. *Journal des Sciences Militaires*, 9, 5–38.

- Kim, I.-S. (2012). Keypad Against Brute Force Attacks on Smartphones. *IET Information Security*, 6, 71–76. doi:10.1049/iet-ifs.2010.0212
- King, A. A., Lenox, M., & Barnett, M. A. (2002). Strategic Responses to the Reputation Commons Problem. In A. F. Hoffman & M. J. Ventresca (Eds.), *Organizations, Policy, and the Natural Environment: Institutional and Strategic Perspectives* (pp. 393–406). Stanford: Stanford University Press. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=624201
- King, S. T., Chen, P. M., Wang, Y.-M., Verbowski, C., Wang, H. J., & Lorch, J. R. (2006). SubVirt: Implementing Malware with Virtual Machines. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy* (pp. 314–327). IEEE. doi:10.1109/SP.2006.38
- Klein, D. V. (1990). “Foiling the Cracker”: A Survey of, and Improvements to, Password Security. In *Proceedings of the 2nd USENIX Security Workshop* (pp. 1–17). USENIX Association.
- Klein, H. K. & Myers, M. D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly*, 23, 67–93. doi:10.2307/249410
- Klutke, G.-A., Kiessler, P. C., & Wortman, M. A. (2003). A Critical Look at the Bathtub Curve. *IEEE Transactions on Reliability*, 52, 125–129. doi:10.1109/TR.2002.804492
- Kuhn, M. G. (2004). Electromagnetic Eavesdropping Risks of Flat-Panel Displays. In *Proceedings of the 4th International Workshop on Privacy Enhancing Technologies* (pp. 88–107). University of Toronto. doi:10.1007/11423409_7
- Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L. F., Hong, J., & Nunge, E. (2007). Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 905–914). Association for Computing Machinery. doi:10.1145/1240624.1240760
- Lacity, M. C., Willcocks, L. P., & Feeny, D. E. (1996). The Value of Selective IT Sourcing. *Sloan Management Review*, 37, 13–25.
- Latanicki, J., Massonet, P., Naqvi, S., Rochwerger, B., & Villari, M. (2010). Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks. In *Proceedings of the Future Internet Assembly 2010* (pp. 127–137). IOS Press Ebooks. doi:10.3233/978-1-60750-539-6-127
- Lee, S. (2011). Shared-Nothing Vs. Shared-Disk Cloud Database Architecture. *International Journal of Energy, Information and Communications*, 2, 211–216.
- Liang, X. & Xiao, Y. (2013). Game Theory for Network Security. *IEEE Communications Surveys & Tutorials*, 15, 472–486. doi:10.1109/SURV.2012.062612.00056
- Liang, Y., Poor, H. V., & (Shitz), S. S. (2009). Information Theoretic Security. *Foundations and Trends in Communications and Information Theory*, 5, 355–580. doi:10.1561/01000000036
- Likert, R. (1932). A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22, 5–55.
- Littlewood, B., Brocklehurst, S., Fenton, N., Mellor, P., Page, S., Wright, D., ... Gollmann, D. (1993). Towards Operational Measures of Computer Security. *Journal of Computer Security*, 2, 211–229. doi:10.3233/JCS-1993-22-308
- Ma, L., Ofoghi, B., Watters, P., & Brown, S. (2009). Detecting Phishing Emails Using Hybrid Features. In *Proceedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2009* (pp. 493–497). IEEE. doi:10.1109/UIC-ATC.2009.103
- Mallery, J. (2009). Building a Secure Organization. In J. R. Vacca (Ed.), *Organizations, Policy, and the Natural Environment: Computer and Information Security Handbook* (pp. 3–21). Amsterdam: Elsevier.

- Malone, D. & Maher, K. (2012). Investigating the Distribution of Password Choices. In *Proceedings of the 21st International Conference on World Wide Web* (pp. 301–310). Association of Computing Machinery. doi:10.1145/2187836.2187878
- Man, C. L. T. & Kayashime, M. (2011). Virtual Machine Placement Algorithm for Virtualized Desktop Infrastructure. In *Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)* (pp. 333–337). IEEE. doi:10.1109/CCIS.2011.6045085
- Manadhata, P. K. & Wing, J. M. (2010). An Attack Surface Metric. *IEEE Transactions on Software Engineering*, 37, 371–386. doi:10.1109/TSE.2010.60
- Manikandan, S. (2011). Measures of Central Tendency: The Mean. *Journal of Pharmacology & Pharmacotherapeutics*, 2, 140–142. doi:10.4103/0976-500X.81920
- Marechal, S. (2008). Advances in Password Cracking. *Journal in Computer Virology*, 4, 73–81. doi:10.1007/s11416-007-0064-y
- Matsumoto, T., Matsumoto, H., Yamada, K., & Hoshino, S. (2002). Impact of Artificial “Gummy” Fingers on Fingerprint Systems. In *SPIE Proceedings 4677: Optical Security and Counterfeit Deterrence Techniques IV* (pp. 275–289). The International Society for Optics and Photonics. doi:10.1117/12.462719
- Melão, N. & Pidd, M. (2000). A Conceptual Framework for Understanding Business Processes and Business Process Modelling. *Information Systems Journal*, 10, 105–129. doi:10.1046/j.1365-2575.2000.00075.x
- Miller, K. W., Voas, J., & Hurlburt, G. F. (2012). BYOD: Security and Privacy Considerations. *IT Professional*, 14, 53–55. doi:10.1109/MITP.2012.93
- Moore, A. (2008). Defining Privacy. *Journal of Social Philosophy*, 39, 411–428. doi:10.1111/j.1467-9833.2008.00433.x
- Moore, G. E. (1965). Cramming More Components Onto Integrated Circuits. *Electronics Magazine*, 38, 1–4.
- Mori, G. & Malik, J. (2003). Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 134–141). IEEE.
- Morris, R. & Thompson, K. (1979). Password Security: A Case History. *Communications of the ACM*, 22, 594–597. doi:10.1145/359168.359172
- Neumann, P. G. (2003). Computer Security. *Issues in Science and Technology*, Summer, 50–54. Retrieved 2013-06-06, from <http://issues.org/19.4/updated/neumann.pdf>
- Nicholson, A. J., Chawathe, Y., Chen, M. Y., Noble, B. D., & Wetherall, D. (2006). Improved Access Point Selection. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services* (pp. 233–245). Association for Computing Machinery. doi:10.1145/1134680.1134705
- Nikiforakis, N., Younan, Y., & Joosen, W. (2010). HProxy: Client-Side Detection of SSL Stripping Attacks. In *Detection of Intrusions and Malware, and Vulnerability Assessment: Proceedings 7th International Conference, DIMVA 2010, Bonn, Germany, July 8-9, 2010* (pp. 200–218). Berlin: Springer Verlag. doi:10.1007/978-3-642-14215-4_12
- Orlandi, E. (1991). The Cost of Security. In *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology* (pp. 192–196). doi:10.1109/CCST.1991.202214
- Pang, R., Yegneswaran, V., Barford, P., Paxson, V., & Peterson, L. (2004). Characteristics of Internet Background Radiation. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement* (pp. 27–40). doi:10.1145/1028788.1028794
- Parent, W. A. (1983). Privacy, Morality, and the Law. *Philosophy & Public Affairs*, 12, 269–288. Retrieved from <http://www.jstor.org/stable/2265374>

- Pask, G. (1996). Heinz von Foerster's Self-Organisation, the Progenitor of Conversation and Interaction Theories. *Systems Research*, 13, 349–362.
- Perçin, S. (1993). Fuzzy Multi-Criteria Risk-Benefit Analysis of Business Process Outsourcing (BPO). *Information Management & Computer Security*, 16, 213–234. doi:10.1108/09685220810893180
- Percival, C. (1999). Stronger Key Derivation Via Sequential Memory-Hard Functions. In *Proceedings of BSDCan 2009* (pp. 1–16). Retrieved from <http://www.bsdcn.org/2009/schedule/events/147.en.html>
- Pinheiro, E., Weber, W.-D., & Barroso, L. A. (2006). Failure Trends in a Large Disk Drive Population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies* (pp. 17–29). USENIX.
- Pinkas, B. & Sander, T. (2002). Securing Passwords Against Dictionary Attacks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 161–170). Association of Computing Machinery. doi:10.1145/586110.586133
- Popek, G. J. & Goldberg, R. P. (1974). Formal Requirements for Virtualizable Third Generation Architectures. *Communications of the ACM*, 17, 412–421. doi:10.1145/361011.361073
- Pritchett, D. (2008). BASE: An Acid Alternative. *Object-Relational Mapping*, 6, 51–59.
- Provost, N. (1999). A Future-Adaptable Password Scheme. In *Proceedings of the 1999 USENIX Annual Technical Conference* (pp. 81–92). USENIX.
- Puppy, R. F. (1998). NT Web Technology Vulnerabilities. *Phrack Magazine*, 8. Retrieved from <http://www.phrack.org/issues.html?issue=54&id=8#article>
- Quing-hai, B. & Ying, Z. (2011). Study on the Access Control Model in Information Security. In *Proceedings of the Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)* (Vol. 1, pp. 830–834). doi:10.1109/CSQRWC.2011.6037079
- Rahumed, A., Chen, H. C. H., Tang, Y., Lee, P. P. C., & Lui, J. C. S. (2011). A Secure Cloud Backup System with Assured Deletion and Version Control. In *Proceedings of the 2011 40th International Conference on Parallel Processing Workshops* (pp. 160–167). doi:10.1109/ICPPW.2011.17
- Ramzan, Z. (2010). Phishing Attacks and Countermeasures. In *Handbook of Information and Communication Security* (pp. 443–448). New York: Springer.
- Rhee, M. & Valdez, M. E. (2009). Contextual Factors Surrounding Reputation Damage with Potential Implications for Reputation Repair. *Academy of Management Review*, 34, 146–168. doi:10.5465/AMR.2009.35713324
- Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009). Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (pp. 199–212). Association of Computing Machinery. doi:10.1145/1653662.1653687
- Roscoe, T., Ephinstone, K., & Heiser, G. (2007). Hype and Virtue. In *Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems* (pp. 1–6). USENIX Association.
- Rosenblum, M. (2004). The Reincarnation of Virtual Machines. *Virtual Machines*, 2, 34–40.
- Rowley, J. (2002). Using Case Studies in Research. *Management Research News*, 25, 16–27. doi:10.1108/01409170210782990
- Rowley, J. (2007). The Wisdom Hierarchy: Representations of the DIKW Hierarchy. *Journal of Information Science*, 33, 163–180.
- Runeson, P. & Höst, M. (2009). Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering*, 14, 131–164. doi:10.1007/s10664-008-9102-8
- Runyan, W. M. (1982). In Defense of the Case Study Method. *American Journal of Orthopsychiatry*, 52, 440–446. doi:10.1111/j.1939-0025.1982.tb01430.x

- Saltzer, J. H. (1974). Protection and the Control of Information Sharing in Multics. *Communications of the ACM*, 17, 388–402.
- Scarfò, A. (2012). New Security Perspectives Around BYOD. In *Proceedings of the 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications* (pp. 446–451). IEEE. doi:10.1109/BWCCA.2012.79
- Schmidt, A. (2010). Ubiquitous Computing: Are We There Yet? *Computer*, 43, 95–97. doi:10.1109/MC.2010.54
- Schroeder, B. & Gibson, G. A. (2007). Understanding Failures in Petascale Computers. *Journal of Physics: Conference Series*, 78, 1–11. doi:10.1088/1742-6596/78/1/012022
- Schwartz, E. (2006). Lessons Learned from HP. *InfoWorld*, 28, 6–6.
- Seki, Ī. (2008). The Importance of ICT for the Knowledge Economy: A Total Factor Productivity Analysis for Selected OECD Countries. In O. Esen & A. Ogun (Eds.), *Proceedings of the Conference on Emerging Economic Issues in a Globalizing World*.
- Seshadri, A., Luk, M., Qu, N., & Perrig, A. (2007). SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSES. In *Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles* (pp. 335–350). Association of Computing Machinery. doi:10.1145/1294261.1294294
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 379–423.
- Shannon, C. E. (1949). Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28, 656–715. doi:10.1561/01000000036
- Sheng, S., Kumaraguru, P., Acquisti, A., Cranor, L., & Hong, J. (2009). Improving Phishing Countermeasures: An Analysis of Expert Interviews. In *Proceedings of the eCrime Researchers Summit, 2009* (pp. 1–15). IEEE. doi:10.1109/ECRIME.2009.5342608
- Shroeder, B. & Gibson, G. A. (2007). Disk Failures in the Real World: What Does an MTTf of 1,000,000 Hours Mean to You? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies* (pp. 1–16). USENIX Association.
- Siegel, M. & Matnick, S. E. (1991). A Metadata Approach to Resolving Semantic Conflicts. In *Proceedings of the 17th International Conference on Very Large Data Bases* (pp. 133–145).
- Slovic, P. & Weber, E. U. (1987). Perception of Risk. *Science*, 236, 280–285. doi:10.1126/science.3563507
- Smith, A. J. (1982). Cache Memories. *Computing Surveys*, 14, 473–530. doi:10.1145/356887.356892
- Stevens, M. [Marc], Lenstra, A. K., & de Veger, B. (2012). Chosen-Prefix Collisions for MD5 and Applications. *International Journal of Applied Cryptography*, 2, 322–359. doi:10.1504/IJACT.2012.048084
- Stolfo, S., Bellovin, S. M., & Evans, D. (2011). Measuring Security. *IEEE Security & Privacy*, 9, 60–65. doi:10.1109/MSP.2011.56
- Stone, J. & Merrion, S. (2004). Instant Messaging or Instant Headache. *Queue – Search Engines*, 2, 72–80. doi:10.1145/988392.988410
- Stubblefield, A., Ioannidis, J., & Rubin, A. D. (2004). A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP). *ACM Transactions on Information and System Security*, 7, 319–332. doi:10.1145/996943.996948
- Szefer, J., Keller, E., Lee, R. B., & Rexford, J. (2011). Eliminating the Hypervisor Attack Surface for a More Secure Cloud. In *Proceedings of the 18th ACM Conference on Computer and Communications Security* (pp. 401–412). Association of Computing Machinery. doi:10.1145/2046707.2046754

- Tellis, W. (1997). Application of a Case Study Methodology. *The Qualitative Report*, 3. Retrieved from <http://www.nova.edu/ssss/QR/QR3-3/tellis2.html>
- Tews, E. & Beck, M. (2007). Practical Attacks Against WEP and WPA. In *Proceedings of the 2nd ACM Conference on Wireless Network Security* (pp. 79–86). Association for Computing Machinery. doi:10.1145/1514274.1514286
- Tews, E., Weinmann, R.-P., & Pyshkin, A. (2007). Breaking 104 Bit WEP in Less Than 60 Seconds. In *Information Security Applications: 8th International Workshop, WISA 2007, Jeju Island, Korea, August 27-29, 2007 Revised Selected Papers* (pp. 188–202). Berlin: Springer Verlag. doi:10.1007/978-3-540-77535-5_14
- Thompson, H. H. (2005). Application Penetration Testing. *IEEE Security & Privacy*, 3. doi:10.1109/MSP.2005.3
- Thorndike, E. L. (1920). A Constant Error in Psychological Ratings. *Journal of Applied Psychology*, 4, 25–29.
- Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, 230–265. doi:10.1112/plms/s2-42.1.230
- Turing, A. M. (1938). On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction. *Proceedings of the London Mathematical Society*, 43, 544–546. doi:10.1112/plms/s2-43.6.544
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 59, 433–460.
- Underbrink, A., Potter, A., Jaenisch, H., & Reifer, D. J. (2012). Application Stress Testing: Achieving Cyber Security by Testing Cyber Attacks. In *Proceedings of the 2012 IEEE Conference on Technologies for Homeland Security* (pp. 556–561). IEEE. doi:10.1109/THS.2012.6459909
- Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M. L., . . . Cranor, L. F. (2012). How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation. In *Proceedings of the 21st USENIX Security Symposium* (pp. 5–20). USENIX.
- van Eck, W. (1985). Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computers & Security*, 4, 269–286. doi:10.1016/0167-4048(85)90046-X
- Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009). A Break in the Clouds: Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, 39, 50–55. doi:10.1145/1496091.1496100
- von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). CAPTCHA: Using Hard AI Problems For Security. In *Advances in Cryptology – EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings* (pp. 294–311). Berlin: Springer Verlag. doi:10.1007/3-540-39200-9_18
- von Neumann, J. (1981). The Principles of Large-Scale Computing Machines. *Annals of the History of Computing*, 3, 263–273. (Reprinted)
- Walsh, T. (2012). 10 Security Domains (Updated). *Journal of AHIMA*, 83, 48–52.
- Walsham, G. (1995). Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, 4, 74–81. doi:10.1057/ejis.1995.9
- Walster, E. (1966). Assignment of Responsibility for an Accident. *Journal of Personality and Social Psychology*, 3, 73–79.
- Wang, Y. [Yingxu], Kinsner, W., & Zhang, D. (2009). Contemporary Cybernetics and Its Facets of Cognitive Informatics and Computational Intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 39 Part B: Cybernetics, 823–833. doi:10.1109/TSMCB.2009.2013721

- Wang, Z. & Jiang, X. (2010). HyperSafe: A Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy* (pp. 380–395). IEEE. doi:10.1109/SP.2010.30
- Wang, Z., Jiang, X., Cui, W., & Ning, P. (2009). Countering Kernel Rootkits with Lightweight Hook Protection. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (pp. 545–554). Association of Computing Machinery. doi:10.1145/1653662.1653728
- Warren, S. D. & Brandeis, L. D. (1890). The Right to Privacy. *Harvard Law Review*, 4.
- Webster, A. F. & Tavares, S. E. (1986). On The Design of S-Boxes. In *Proceedings Advances in Cryptology – CRYPTO’85* (pp. 523–534). Lecture Notes in Computer Science. doi:10.1007/3-540-39799-X_41
- Weir, M., Aggarwal, S., de Medeiros, B., & Glodek, B. (2009). Password Cracking Using Probabilistic Context-Free Grammars. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy* (pp. 391–405). IEEE. doi:10.1109/SP.2009.8
- Weiser, M. (1991). The Computer for the 21st Century. *Scientific American*, 265, 94–104.
- Weske, M., van der Aalst, W., & Verbeek, H. (2004). Advances in Business Process Management. *Data & Knowledge Engineering*, 50, 1–8. doi:10.1016/j.datak.2004.01.001
- West, M. (2009). Building a Secure Organization. In J. R. Vacca (Ed.), *Organizations, Policy, and the Natural Environment: Computer and Information Security Handbook* (pp. 39–51). Amsterdam: Elsevier.
- Weyuker, E. J. (1988). Evaluating Software Complexity Measures. *IEEE Transactions on Software Engineering*, 14, 1357–1365. doi:10.1109/32.6178
- Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-Scale Automatic Classification of Phishing pages. In *Proceedings of the 17th Annual Network and Distributed System Security Symposium* (pp. 1–14). Internet Society. Retrieved from <http://research.google.com/pubs/archive/35580.pdf>
- Willcocks, L., Hindle, J., Feeny, D., & Lacity, M. (2004). IT and Business Process Outsourcing: The Knowledge Potential. *Information Systems Management*, 21, 7–15. doi:10.1201/1078/44432.21.3.20040601/82471.2
- Xie, Y., Yu, F., & Abadi, M. (2009). De-anonymizing the Internet Using Unreliable IDs. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication* (pp. 75–86). Association of Computing Machinery. doi:10.1145/1592568.1592579
- Yampolskiy, R. V. (2006). Analyzing User Password Selection Behavior for Reduction of Password Space. In *Proceedings of the 2006 40th Annual IEEE International Carnahan Conferences Security Technology* (pp. 109–115). IEEE. doi:10.1109/CCST.2006.313438
- Yan, J., Blackwell, A., Anderson, R., & Grant, A. (2004). Password Memorability and Security: Empirical Results. *IEEE Security & Privacy*, 2, 25–31. doi:10.1109/MSP.2004.81
- Zauberman, G. (2003). The Intertemporal Dynamics of Consumer Lock-In. *Journal of Consumer Research*, 30, 405–419. doi:10.1086/378617
- Zhan, J. & Rajamani, V. (2008). The Economics of Privacy. *International Journal of Security and Its Applications*, 2, 101–108.
- Zhang, J. & Wang, Y. [Yonghao]. (2012). A Real-time Automatic Detection of Phishing URLs. In *Proceedings of the 2012 2nd International Conference on Computer Science and Network Technology* (pp. 1212–1216). IEEE. doi:10.1109/ICCSNT.2012.6526142
- Zhang, Y. [Yingian], Juels, A., Reiter, M. K., & Ristenpart, T. (2012). Cross-VM Side Channels and Their Use to Extract Private Keys. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (pp. 305–316). Association of Computing Machinery. doi:10.1145/2382196.2382230

- Zhang, Y. [Yue], Hong, J. I., & Cranor, L. F. (2007). CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 639–648). Association for Computing Machinery. doi:10.1145/1242572.1242659
- Zviran, M. & Haga, W. J. (1999). Password Security: An Empirical Study. *Journal of Management Information Systems*, 15, 161–185.

Online

- ACM. (2013). ACM Code of Ethics and Professional Conduct. Retrieved 2013-10-10, from <http://www.acm.org/about/code-of-ethics>
- Aitel, D. (2012). Why You Shouldn't Train Employees for Security Awareness. Retrieved 2013-08-12, from <http://www.csoonline.com/article/711412/why-you-shouldn-t-train-employees-for-security-awareness>
- Aleshunas, J. (2009). Firewalls. Retrieved 2013-10-30, from <http://mercury.webster.edu/aleshunus/COSC%205130/Chapter-22.pdf>
- Allar, J. (2011). Vulnerability Note VU#723755: Wifi Protected Setup (WPS) PIN Brute Force Vulnerability. Retrieved 2013-08-10, from <http://www.kb.cert.org/vuls/id/723755>
- American Society for Cybernetics. (2008). Summary: The Macy Conferences. Retrieved 2013-05-27, from <http://www.asc-cybernetics.org/foundations/history/MacySummary.htm>
- Anderson, R. (2001). Why Information Security is Hard: An Economic Perspective. Retrieved 2013-06-04, from <http://www.acsac.org/2001/papers/110.pdf>
- APA. (2013). How Does the APA Define “Psychology”? Retrieved 2013-09-11, from <http://www.apa.org/support/about/apa/psychology.aspx#answer>
- Atwood, J. (2007). You're Probably Storing Passwords Incorrectly. Retrieved 2013-06-15, from <http://www.codinghorror.com/blog/2007/09/youre-probably-storing-passwords-incorrectly.html>
- Bernstein, D. J. (2002). SYN Cookies. Retrieved 2013-08-08, from <http://cr.yp.to/syncookies.html>
- Bernstein, D. J. (2010). High-Speed High-Security Cryptography: Encrypting and Authenticating the Whole Internet. Retrieved 2013-08-08, from http://events.ccc.de/congress/2010/Fahrplan/attachments/1773_slides.pdf
- Boritz, J. E. (2003). Core Concepts of Information Integrity: A Survey of Practitioners. Retrieved 2013-06-08, from http://accounting.uwaterloo.ca/uwcisa/symposiums/symposium_2003/papers/Revised%20Papers/Boritz-%20Paper.doc
- Boswell, D. (2012). Storing User Password Securely: Hashing, Salting, and Bcrypt. Retrieved 2013-06-15, from <http://dustwell.com/how-to-handle-passwords-bcrypt.html>
- Carter, R. G. (2003). Authentication vs. Authorization. Retrieved 2013-06-11, from <http://people.duke.edu/~rob/kerberos/authvauth.html>
- Chan, J. (2004). Essentials of Patch Management Policy and Practice. Retrieved 2013-08-01, from <http://www.patchmanagement.org/pmessentials.asp>
- Chaney, A. J. B. (2012). Brute Force Attack. Retrieved 2013-08-07, from https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Brute_force_attack.html
- Cohen, S. M. (2004). Glossary of Logical Terms. Retrieved 2013-10-07, from <http://faculty.washington.edu/smcohen/120/Glossary.pdf>
- Connor, D. (2007). Q&A: Diligent CTO Demystifies Data Deduplication. Retrieved 2013-06-13, from <https://www.networkworld.com/news/2007/050407-qa-diligent.html>

- Dalgaard, P. (2012). Power Calculations for Two-Sample Test for Proportions. Retrieved 2013-10-24, from <http://stat.ethz.ch/R-manual/R-patched/library/stats/html/power.prop.test.html>
- Dougherty, C. R. (2009). Vulnerability Note VU#836068: MD5 Vulnerable to Collision Attacks. Retrieved 2013-08-07, from <http://www.kb.cert.org/vuls/id/836068>
- Dougherty, D. (2001). LAMP: The Open Source Web Platform. Retrieved 2013-08-14, from <http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>
- Dürmuth, M., Chaabane, A., Perito, D., & Castelluccia, C. (2013). When Privacy Meets Security: Leveraging Personal Information for Password Cracking. Retrieved 2013-08-27, from <http://arxiv.org/abs/1304.6584>
- Endres, P. (2012). Do Reverse Proxies Provide Real Security? Retrieved 2013-08-15, from http://www.pabloendres.com/wp-content/uploads/2012/09/Reverse_proxy_report_v1.01.pdf
- EPA Cloud. (2011). Exchange: Hosted Vs On-Premise. Retrieved 2013-06-19, from http://www.epacloud.com/downloads/pdf/exchange_whitepaper_lr_feb2011.pdf
- European Network and Information Security Agency. (2013). Glossary. Retrieved 2013-06-02, from <http://www.enisa.europa.eu/activities/risk-management/current-risk/risk-management-inventory/glossary>
- Evron, S. (2012). Storing Passwords the Right Way. Retrieved 2013-06-15, from <http://arr.gr/blog/2012/01/storing-passwords-the-right-way/>
- Wi-Fi Alliance. (2013a). Glossary. Retrieved 2013-08-10, from www.wi-fi.org/knowledge-center/glossary/w
- Wi-Fi Alliance. (2013b). Wi-fi Protected Setup®. Retrieved 2013-08-10, from <http://www.wi-fi.org/knowledge-center/articles/wi-fi-protected-setup%C3%A2%C2%84%C2%A2>
- Fuchs, P. P. (2009). Scalable Permutations: Quickperm & Metaperm Algorithm Examples Without Recursion. Retrieved 2013-10-23, from <http://permuteweb.tchs.info/>
- Galbally, J., Ross, A., Gomez-Barrero, M., Fierrez, J., & Ortega-Garcia, J. (2012). From the Iriscode to the Iris: A New Vulnerability of Iris Recognition Systems. Retrieved 2013-12-05, from https://media.blackhat.com/bh-us-12/Briefings/Galbally/BH_US_12_Galbally_Iris_Reconstruction_WP.pdf
- Gantz, J. & Reinsel, D. (2011). Extracting Value from Chaos. Retrieved 2013-05-26, from <http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>
- Graham, R. (2012). Common Misconceptions of Password Cracking. Retrieved 2013-12-15, from <http://blog.erratasec.com/2012/08/common-misconceptions-of-password.html>
- Hogan, M. (2009). A Primer on Database Clustering. Retrieved 2013-06-19, from <http://www.scaledb.com/pdfs/ArchitecturePrimer.pdf>
- Hogan, M. (2012). Shared-Disk Vs. Shared-Nothing: Comparing Architectures for Clustered Databases. Retrieved 2013-06-19, from http://www.scaledb.com/pdfs/WP_SDvSN.pdf
- HSM. (2012). New Times, New Storage Media, New Standards. Retrieved 2013-06-14, from <http://en.hsm.eu/bt-en/document-shredder/din-66399-en>
- Hunt, T. (2011a). A Brief Sony Password Analysis. Retrieved 2013-10-24, from <http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html>
- Hunt, T. (2011b). Who's Who of Bad Password Practices — Banks, Airlines, and More. Retrieved 2013-12-13, from <http://www.troyhunt.com/2011/01/whos-who-of-bad-password-practices.html>
- IBM. (2007). Implementing the Business Process. Retrieved 2013-06-02, from <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wbit.sample.appl.3.doc/loanapplication/topics/mod-good.html>
- IEEE. (2013). IEEE Code of Ethics. Retrieved 2013-10-10, from <http://www.ieee.org/about/corporate/governance/p7-8.html>

- Irish, V. (2003). Disclosing Confidential Information. Retrieved 2013-06-08, from http://www.wipo.int/sme/en/documents/disclosing_inf.htm
- Komorowski, M. (2009). A History of Storage Cost. Retrieved 2013-06-20, from <http://www.mkomo.com/cost-per-gigabyte>
- Korolkova, K. (2009). Smart Password Mutations Explained. Retrieved 2013-08-07, from <http://blog.crackpassword.com/2009/04/smart-password-mutations-explained/>
- Kozubek, A. (2013). Database Modeling Tip: How to Store Passwords in a Database. Retrieved 2013-06-15, from <http://onewebsql.com/blog/how-to-store-passwords>
- Liebert Corporation. (2003). High-Availability Power Systems, Part II: Redundancy Options. Retrieved 2013-06-17, from http://www.emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White%20Papers/High-Availability%20Power%20Systems,%20Part%20II_Redundancy%20Options.pdf
- Malone, D. & Maher, K. (2011). Investigating the Distribution of Password Choices. doi:10.1145/2187836.2187878
- Martin, R. (2007). Wall Street's Quest To Process Data At The Speed Of Light. Retrieved 2013-06-17, from <https://www.informationweek.com/wall-streets-quest-to-process-data-at-th/199200297>
- McDowell, M. (2009). Security Tip (ST04-015): Understanding Denial-of-Service Attacks. Retrieved 2013-08-08, from <https://www.us-cert.gov/ncas/tips/st04-015>
- McGraw, G. & Miguez, S. (2012). Data Supports Need for Security Awareness Training Despite Naysayers. Retrieved 2013-08-12, from <http://searchsecurity.techtarget.com/news/2240162630/Data-supports-need-for-awareness-training-despite-naysayers>
- Merler, M. & Jacob, J. (2009). Final report. Retrieved 2013-08-04, from <http://www.cs.columbia.edu/~mmerler/project/Final%20Report.pdf>
- Microsoft. (2005). System Integrity: Ensuring Integrity. Retrieved 2013-06-16, from <http://technet.microsoft.com/en-us/library/cc700839.aspx>
- Microsoft. (2007). Kernel Patch Protection: Frequently Asked Questions. Retrieved 2013-06-30, from <http://msdn.microsoft.com/en-us/windows/hardware/gg487353.aspx>
- Microsoft. (2013). SQL Injection. Retrieved 2013-08-14, from [http://technet.microsoft.com/en-us/library/ms161953\(v=sql.100\).aspx](http://technet.microsoft.com/en-us/library/ms161953(v=sql.100).aspx)
- Mizzi, A. (2005). Return on Information Security Investment: Are you spending enough? Are you spending too much? Retrieved 2013-06-04, from <http://www.adrianmizzi.com/ROISI-Paper.pdf>
- Moertel, T. (2006). Never Store Passwords in a Database! Retrieved 2013-06-15, from <http://blog.moertel.com/posts/2006-12-15-never-store-passwords-in-a-database.html>
- Moschella, D., Neal, D., Opperman, P., & Taylor, J. (2004). The 'Consumerization' of Information Technology [Position Paper]. Retrieved 2013-06-25, from <http://www.smaele.nl/edocs/Taylor-Consumerization-2004.pdf>
- Mueller, N. (2012). Server Backup Decision Point: On-Premise Vs. Off-Site Vs. Cloud. Retrieved 2013-06-19, from <http://www.zetta.net/blog/server-backup-decision-point-onpremise-offsite-cloud/>
- Naor, M. (1996). Verification of a Human in the Loop or Identification via the Turing Test. Retrieved 2013-08-04, from <http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.pdf>
- NASA. (1998). Mean Time to Repair Predictions. Retrieved 2013-06-21, from <http://engineer.jpl.nasa.gov/practices/at2.pdf>
- Nielsen, P. M. (2012). Storing Passwords Securely. Retrieved 2013-06-15, from <http://throwingfire.com/storing-passwords-securely/>

- Noor, E. (2011). The Problem with Cyber Terrorism. Retrieved 2013-08-01, from http://www.isis.org.my/attachments/740_Elina_SEARCCT_V2_25Feb11.pdf
- NSA. (2010). Defense in Depth: A Practical Strategy for Achieving Information Assurance in Today's Highly Networked Environments. Retrieved 2013-12-06, from http://www.nsa.gov/ia/_files/support/defenseindepth.pdf
- Opsahl, K. & Reitman, R. (2013). The Disconcerting Details: How Facebook Teams Up With Data Brokers to Show You Targeted Ads. Retrieved 2013-06-11, from <https://www.eff.org/deeplinks/2013/04/disconcerting-details-how-facebook-teams-data-brokers-show-you-targeted-ads>
- OWASP. (2013). OWASP Top 10 – 2013: The Ten Most Critical Web Application Security Risks. Retrieved 2013-08-14, from <http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf>
- Perrin, C. (2008). The CIA Triad. Retrieved 2013-06-08, from <http://www.techrepublic.com/blog/security/the-cia-triad/488>
- Perrin, C. (2009). The Importance of Privilege Separation. Retrieved 2013-06-10, from <http://sec.apotheon.org/articles/the-importance-of-privilege-separation>
- Petitcolas, F. (2013). Cryptographie Militaire. Retrieved 2013-06-15, from <http://www.petitcolas.net/fabien/kerckhoffs/>
- Ponemon, L. (2013). 2013 Cost of Data Breach Study: Global Analysis. Retrieved 2013-12-02, from https://www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf
- Radack, S. (2006). Domain Name System Services: NIST Recommendations for Secure Deployment. Retrieved 2013-11-08, from <http://www.itl.nist.gov/lab/bulletns/bltnjun06.htm>
- Reichl, D. (2013). Password Quality Estimation. Retrieved 2013-12-13, from http://keepass.info/help/kb/pw_quality_est.html
- Rekouche, K. (2011). Early Phishing. Retrieved 2013-08-12, from <http://arxiv.org/abs/1106.4692>
- Riofrio, M. (2013). Hacking Back: Digital Revenge is Sweet but Risky. Retrieved 2013-06-06, from <http://www.pcworld.com/article/2038226/hacking-back-digital-revenge-is-sweet-but-risky.html>
- Ritchie, D. M. (1979). On the Security of UNIX. Retrieved 2013-06-30, from <ftp.cerias.purdue.edu/pub/doc/misc/d.ritchie-on.security.of.unix.ps.Z>
- Rosser, P. (2006). Thoughts on Software Complexity. Retrieved 2013-06-28, from <https://blogs.msdn.com/b/peterrosser/archive/2006/06/02/softwarecomplexity.aspx>
- Sanders, C. (2010). Understanding Man-in-the-Middle Attacks: ARP Cache Poisoning (Part 1). Retrieved 2013-08-10, from http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html
- Schneier, B. (2007). Schneier: Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'. Retrieved 2013-08-14, from <https://www.schneier.com/essay-146.html>
- Schneier, B. (2011). Whitelisting vs. Blacklisting. Retrieved 2013-07-05, from https://www.schneier.com/blog/archives/2011/01/whitelisting_vs.html
- Schreiber, S. (2010). Concept of a Professional Code of Ethics for Penetration Testers. Retrieved 2013-10-21, from https://www.syss.de/fileadmin/ressources/040_veroeffentlichungen/dokumente/Code_of_Ethics_Penetration_Testers.pdf
- Shah, S. (2002). Top Ten Web Attacks. Retrieved 2013-08-15, from <https://www.blackhat.com/presentations/bh-asia-02/bh-asia-02-shah.pdf>
- Sheppard, J. (2003). What is Computer Science? Retrieved 2013-09-02, from <http://www.cs.bu.edu/AboutCS/WhatIsCS.pdf>

- Singh, J. (2009). Understanding Data Deduplication. Retrieved 2013-06-13, from <http://www.druva.com/blog/understanding-data-deduplication/>
- Small, P. E. (2012). Defense in Depth: An Impractical Strategy for a Cyber World. Retrieved 2013-12-06, from <https://www.sans.org/reading-room/whitepapers/warfare/defense-depth-impractical-strategy-cyber-world-33896>
- Sotirov, A., Stevens, M., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D. A., & de Weger, B. (2008). MD5 Considered Harmful Today: Creating a Rogue CA Certificate. Retrieved 2013-08-07, from <http://www.win.tue.nl/hashclash/rogue-ca/>
- Stajano, F. (2003). Security for Whom? The Shifting Security Assumptions of Pervasive Computing. Retrieved 2013-06-06, from <http://www.cl.cam.ac.uk/~fms27/papers/2003-stajano-shifting.pdf>
- Stanley, S. (2011). MTBF, MTTR, MTTF & FIT: Explanation of Terms. Retrieved 2013-06-21, from <http://www.imcnetworks.com/Assets/DocSupport/WP-MTBF-0311.pdf>
- Texas Instruments. (2013). Handset: Smartphone. Retrieved 2013-06-27, from http://www.ti.com/solution/handset_smartphone
- The Standish Group. (1999). Five “T”s” of Database Availability. Retrieved 2013-06-17, from <http://www.sybase.com/content/1020238/StandishFiveTsHA.pdf>
- Tobias, P. (2012). E-Handbook of Statistical Methods [8.2.1.4 “Bathtub” Curve]. Retrieved 2013-06-21, from <http://www.itl.nist.gov/div898/handbook/apr/section1/apr124.htm>
- Toponce, A. (2011). Strong Passwords NEED Entropy. Retrieved 2013-07-03, from <http://pthree.org/2011/03/07/strong-passwords-need-entropy/>
- Townsend, K. (2011). Does it Matter if it’s Black or White(listing)? Retrieved 2013-07-05, from <http://www.infosecurity-magazine.com/view/20083/does-it-matter-if-its-black-or-whitelisting->
- Varian, H. R. (2001). System Reliability and Free Riding Problem. Retrieved 2013-06-04, from <http://www2.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/econws/49.pdf>
- Viehböck, S. (2011). Brute Forcing Wi-Fi Protected Setup: When Poor Design Meets Poor Implementation. Retrieved 2013-08-10, from http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf
- Walker-Morgan, D. (2011). Wi-Fi Protected Setup Made Easier to Brute Force – Update. Retrieved 2013-08-10, from <http://www.h-online.com/open/news/item/Wi-Fi-Protected-Setup-made-easier-to-brute-force-Update-1401822.html>
- Wheeler, D. (2012). zxcvbn: Realistic Password Strength Estimation. Retrieved 2013-11-20, from <https://tech.dropbox.com/2012/04/zxcvbn-realistic-password-strength-estimation/>
- Yeend, H. (2005). Breaking CAPTCHA Without OCR. Retrieved 2013-08-04, from http://www.puremango.co.uk/2005/11/breaking_captcha_115/
- Zdrnja, B. (2012). Is It Time to Get Rid of NetBIOS? Retrieved 2013-11-25, from <https://isc.sans.edu/diary/Is+it+time+to+get+rid+of+NetBIOS?/12454>
- Zimand, M. (2013). Cryptography: Hash Functions and Message Authentication Codes. Retrieved 2013-08-05, from <http://triton.towson.edu/~mzimand/cryptostuff/N7-Hash.pdf>

Reports, Standards, White Papers

- Abliz, M. (2011). *Internet Denial of Service Attacks and Defense Mechanisms*. University of Pittsburgh. Retrieved 2013-08-08, from <http://people.cs.pitt.edu/~mehmud/docs/abliz11-TR-11-178.pdf>
- Bar-El, H. (2003). *Introduction to Side Channel Attacks*. Retrieved 2013-08-01, from <http://gauss.eecs.uc.edu/Courses/c653/lectures/SideC/intro.pdf>

- Bennett, S. H. (2001). *The NSA: A Brief Examination of the “No Such Agency”*. Retrieved 2013-08-01, from https://www.sans.org/reading_room/whitepapers/standards/nsa-examination-no-agency_544
- Benson, T. (2006). *TIA-942: Data Center Standards Overview*. ADC. Retrieved 2013-06-21, from <http://www.adc.com/us/en/Library/Literature/102264AE.pdf>
- Bindé, J. (2005). *Towards Knowledge Societies: UNESCO World Report*. UNESCO. Retrieved 2013-05-26, from <http://www.unesco.org/new/en/communication-and-information/resources/publications-and-communication-materials/publications/full-list/towards-knowledge-societies-unesco-world-report/>
- BSI. (2009). *BSI Standard 100-2: It-Grundschutz Methodology*. Bundesamt für Sicherheit in der Informationstechnik. Retrieved 2013-08-16, from https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/BSIStandards/standard_100-2_e_pdf.pdf?__blob=publicationFile
- COBIT. (2012). *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. ISACA. Retrieved 2013-06-16, from <http://www.isaca.org/COBIT/Pages/COBIT-5-Framework-product-page.aspx>
- Committee on National Security Systems. (2010). *National Information Assurance (IA) Glossary*. Committee on National Security Systems. Retrieved 2013-06-02, from http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf
- Common Criteria. (2012). *Common Criteria for Information Technology Security Evaluation: Part 1: Introduction and General Model*. Retrieved 2013-08-16, from <http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf>
- Computer Security Division. (2006). *FIPS 200: Minimum Security Requirements for Federal Information and Information Systems*. National Institute of Standards and Technology. Retrieved 2013-06-02, from <http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf>
- Computer Security Division. (2012). *NIST Special Publication 800-30 Revision 1: Guide for Conducting Risk Assessments*. National Institute of Standards and Technology. Retrieved 2013-06-02, from http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf
- Dacey, R. F. (2003). *Effective Patch Management is Critical to Mitigating Software Vulnerabilities*. Retrieved 2013-08-02, from <http://www.gao.gov/assets/120/110329.pdf>
- DoD. (1981). *MIL-STD-721C: Definitions of Terms for Reliability and Maintainability*. Department of Defense. Retrieved 2013-06-16, from http://www.everyspec.com/MIL-STD/MIL-STD-0700-0799/MIL-STD-721C_1040/
- FFIEC. (2005). *Authentication in an Internet Banking Environment*. Retrieved 2013-07-03, from http://www.ffiec.gov/pdf/authentication_guidance.pdf
- FOIS. (2006). *Pervasive Computing: Trends and Impacts*. Federal Office for Information Security. Retrieved 2013-06-26, from https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Percenta/Percenta_eacc_pdf.pdf?__blob=publicationFile
- Graff, M. G. (2001). *Secure Coding: The State of Practice*. Retrieved 2013-06-28, from http://markgraff.com/mg_writings/SC_2001_public.pdf
- Guttman, B. & Roback, E. (1995). *Introduction to Computer Security: The NIST Handbook*. National Institute of Standards and Technology. Retrieved 2013-06-11, from <http://csrc.nist.gov/publications/nistpubs/800-12/>
- Herzog, P. (2010). *OSSTMM 3: The Open Source Security Testing Methodology Manual*. Institute for Security and Open Methodologies. Retrieved 2013-06-02, from <http://www.isecom.org/mirror/OSSTMM.3.pdf>
- HKSAR. (2008). *VPN Security*. Retrieved 2013-07-03, from <http://www.infosec.gov.hk/english/technical/files/vpn.pdf>

- IBM. (2011). *Security and High Availability in Cloud Computing Environments*. IBM. Retrieved 2013-06-21, from http://www-935.ibm.com/services/za/gts/cloud/Security_and_high_availability_in_cloud_computing_environments.pdf
- Information Sciences Institute. (1981). *RFC 793 – Transmission Control Protocol: DARPA Internet Program Protocol Specification*. Internet Engineering Task Force. Retrieved 2013-07-01, from <https://tools.ietf.org/html/rfc793>
- ISACA. (2009). *The Risk IT Framework*. ISACA.
- ISO. (2002). *ISO/IEC Guide 73 - Risk management – Vocabulary – Guidelines for use in standards*. International Organization for Standardization.
- ISO. (2008a). *ISO/IEC 27005:2008 – Information Technology – Security Techniques – Information Security Risk Management*. International Organization for Standardization.
- ISO. (2008b). *Quality Management Systems – Requirements*. International Organization for Standardization.
- ISO. (2009). *ISO Guide 73:2009 - Risk Management – Vocabulary*. International Organization for Standardization.
- ISO. (2012a). *ISO 22301:2012: Societal Security – Business Continuity Management Systems – Requirements*. International Organization for Standardization.
- ISO. (2012b). *ISO/IEC 11179-1:2004 Metadata Registries (MDR) - Part 1: Framework*. International Organization for Standardization.
- ITIL. (2011). *ITIL® Glossary and Abbreviations: English*. Information Technology Infrastructure Library. Retrieved 2013-06-16, from http://www.itil-officialsite.com/InternationalActivities/ITILGlossaries_2.aspx
- Leonard, M. (2006). *Pretexting Prevention: Minimizing Inbound and Outbound Risks*. Axentis. Retrieved 2013-08-12, from <http://osint.pbworks.com/f/Pretexting%20Prevention.pdf>
- Liu, X., Pitoura, E., & Bhargava, B. (1995). *Adapting Distributed Database Systems for High Availability*. Purdue University. Retrieved 2013-06-17, from <http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2190&context=cstech>
- Mandiant. (2013). *M-Trends 2013: Attack the Security Gap*. Mandiant. Retrieved 2013-06-07, from <https://www.mandiant.com/resources/m-trends/>
- McCallister, E., Grance, T., & Scarfone, K. (2010). *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*. National Institute of Standards and Technology. Retrieved 2013-06-08, from <http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf>
- Mead, N. R., Hough, E. D., & Stehney II, T. R. (2005). *Security Quality Requirements Engineering (SQUARE) Methodology*. Carnegie Mellon University. Retrieved 2013-08-26, from <http://www.cert.org/archive/pdf/05tr009.pdf>
- Mogul, J. C. (1989). *Simple and Flexible Datagram Access Controls for Unix-Based Gateways*. Retrieved 2013-07-01, from <http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-89-4.pdf>
- Network Working Group. (2000). *PKCS #5: Password-Based Cryptography Specification Version 2.0*. Internet Engineering Task Force. Retrieved 2013-07-04, from <https://tools.ietf.org/html/rfc2898>
- Network Working Group. (2006). *RFC 4271: A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force. Retrieved 2013-05-27, from <https://tools.ietf.org/html/rfc4271>
- Nieh, J., Yang, S. J., & Novik, N. (2000). *A Comparison of Thin-Client Computing Architectures*. Network Computing Laboratory, Columbia University. Retrieved 2013-06-26, from <http://www.nomachine.com/documentation/pdf/cucs-022-00.pdf>

- O'Connor, A. C. & Loomis, R. J. (2010). *2010 Economic Analysis of Role-Based Access Control: Final Report*. National Institute of Standards and Technology. Retrieved 2013-12-14, from http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf
- Open Information Systems Security Group. (2006). *Information Systems Security Assessment Framework (ISSAF): Draft 0.2.1A*. Committee on National Security Systems. Retrieved 2013-06-02, from <http://www.oisssg.org/files/issaf0.2.1A.pdf>
- Postel, J. (1980). *RFC 798 – User Datagram Protocol*. Internet Engineering Task Force. Retrieved 2013-07-01, from <https://tools.ietf.org/html/rfc768>
- Randazzo, M. R. & Cappelli, D. (2005). *Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector*. Carnegie Mellon University. Retrieved 2013-08-15, from <http://www.dtic.mil/dtic/tr/fulltext/u2/a441249.pdf>
- Rivest, R. (1992). *RFC 1321: The MD5 Message-Digest Algorithm*. Internet Engineering Task Force. Retrieved 2013-08-07, from <https://tools.ietf.org/html/rfc1321>
- Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). *Technical Guide to Information Security Testing and Assessment*. National Institute of Standards and Technology. Retrieved 2013-08-15, from http://www.nist.gov/customcf/get_pdf.cfm?pub_id=152164
- Shackleford, D. (2009). *Application Whitelisting: Enhancing Host Security*. Retrieved 2013-07-05, from https://www.sans.org/reading_room/analysts_program/McAfee_09_App_Whitelisting.pdf
- Shirey, R. W. (2007). *RFC 4949: Internet Security Glossary, Version 2*. Internet Engineering Task Force. Retrieved 2013-08-07, from <https://tools.ietf.org/html/rfc4949>
- SNIA. (2012). *Cloud Data Management Interface (CDMI®): Version 1.0.2*. Storage Networking Industry Association. Retrieved 2013-06-20, from <http://snia.org/sites/default/files/CDMI%20v1.0.2.pdf>
- The Commission on the Theft of American Intellectual Property. (2013). *The IP Commission Report*. The Commission on the Theft of American Intellectual Property. Retrieved 2013-06-06, from http://www.ipcommission.org/report/IP_Commission_Report_052213.pdf
- The Office of Legal Education. (2010). *Prosecuting Computer Crimes*. Computer Crime and Intellectual Property Section. Retrieved 2013-06-06, from <http://www.justice.gov/criminal/cybercrime/docs/ccmanual.pdf>
- The Office of the Australian Information Commissioner. (2013). *Guide to Information Security: 'Reasonable Steps' to protect personal information*. Australian Government. Retrieved 2013-06-07, from <http://www.oaic.gov.au/privacy/privacy-resources/privacy-guides/guide-to-information-security>
- Turing, A. M. (1948). *Intelligent Machinery*. National Physical Laboratory.
- Turner, W. P., Seader, J. H., & Renaud, V. E. (2010). *Data Center Site Infrastructure Tier Standard: Topology*. Uptime Institute. Retrieved 2013-06-21, from http://www.uptimeinstitute.com/component/docman/doc_download/5-tiers-standard-topology

LIST OF PUBLICATIONS

- Sarga, L. (2012a). Cloud Computing: An Overview. *Journal of Systems Integration*, 3, 3–14.
- Sarga, L. (2012b). Current Issues of Resource Sharing and Provisioning Paradigm. In *Sborník 12. mezinárodní konference doktorandů, studentů a mladých vědeckých pracovníků IMEA 2012* (pp. 96–101).
- Sarga, L. & Jašek, R. (2011). Distributed Denial of Service Attacks as Threat Vectors to Economic Infrastructure: Motives, Estimated Losses and Defense Against the HTTP/1.1 GET and SYN Floods Nightmares. In R. Ottis (Ed.), *Proceedings of the 10th European Conference on Cyber Warfare and Security ECIW-2011* (pp. 228–236).
- Sarga, L. & Jašek, R. (2012). User-Side Password Authentication: A Study. In E. Filiol & R. Erra (Eds.), *Proceedings of the 11th European Conference on Cyber Warfare and Security ECIW-2012* (pp. 237–243).
- Sarga, L. & Jašek, R. (2013). Mobile Cyberwarfare Threats and Mitigations: An Overview. In R. Kuusisto & E. Kurkinen (Eds.), *Proceedings of the 12th European Conference on Cyber Warfare and Security ECCWS-2013* (pp. 243–251).
- Sarga, L. & Jašek, R. (2014). Human Factor: The Weakest Link of Security? In A. Liaropoulos & G. Tsihrintzis (Eds.), *Proceedings of the 13th European Conference on Cyber Warfare and Security ECCWS-2014*.
- Sarga, L., Jašek, R., & Benda, R. (2013). Security Assessment of SHA-1 and MD5 Cryptographic Hash Algorithms. In A. Zak & A. Slaby (Eds.), *Proceedings of the 14th WSEAS International Conference on Automation & Information (ICAI '13)* (pp. 19–24).
- Sarga, L., Jašek, R., Vala, R., & Benda, R. (2011). Launching Distributed Denial of Service Attacks by Network Protocol Exploitation. In N. Mastorakis, M. Demiralp, & N. A. Baykara (Eds.), *Proceedings of the 2nd International Conference on Applied Informatics and Computing Theory (AICT '11)* (pp. 210–216).
- Sarga, L., Kovářík, M., & Klímek, P. (2014). Usage Of Control Charts For Time Series Analysis In Financial Management. *Journal of Business Economics and Management*, 1–20. doi:10.3846/16111699.2012.732106
- Sarga, L. & Vala, R. (2010). Softwarový audit. In *Optimalization and Simulation Methods* (pp. 1–7).
- Sarga, L., Vala, R., & Benda, R. (2013). Security Reverse Engineering of Mobile Operating Systems: A Summary. In O. Nakov, P. Borovska, A. Antonio, V. Mladenov, L. Zinchenko, & A. Fuentes-Penna (Eds.), *Proceedings of the 17th WSEAS International Conference on Computers (COMPUTERS '13)* (pp. 112–117).

APPENDICES

Appendix A: Questionnaire Template (Czech Version)

In the appendix, the Czech version of a questionnaire survey whose results are reported in chapter 4, is included. It was produced in L^AT_EX using the SDAPS¹ framework. To cover a representative sample of respondents who do not use the Internet on a regular basis, the questionnaire was distributed in paper form.

¹<http://sdaps.org/>

Dobrý den.

Vítejte při (dalším) dotazníku, který vám někdo podstrčil k vyplnění. Snad vám nezabere mnoho času a bude intuitivní; o to jsem se při jeho tvorbě snažil, a také o to, abyste neusínali nudou, proto omluvte méně formální tón:) Když jsem zkoušel cvičně vyplňovat, nezabral mi déle než 15 minut. Do 30 minut tedy nebudete mít co dělat a můžete se vrátit k příjemnějším věcem vědíce (mimochodem přechodník), že jste mi pomohli k získání titulu Ph.D.

Křížkujte standardně, kdybyste chtěli odpověď přehodnotit, začerněte zcela původní políčko a zaškrtněte nové. Pokud není uvedeno jinak, vyberte vždy pouze jednu odpověď, která je úplně nebo nejbližší tomu, co byste si představovali.

1 Obecný přehled o IT

Nejdříve uděláme exkurz do IT.

1.1 Jak byste se zhodnotil/a jako uživatel (odborně se tomu říká počítačová gramotnost)?

- Bez zkušeností: nevíte, kde je tlačítko pro zapínání
- Začátečník: prohlížení Facebooku; psaní dopisů a vytváření tabulek, ale nic složitého, základní věci; poslech hudby; sledování filmů
- Středně pokročilý: umíte si nastavit emailového klienta; styly ve Wordu; vzorce v Excelu; znáte význam chybových hlášek
- Vysoce pokročilý: makra; znalost sítí (nastavení proxy nebo VPN); umíte nainstalovat a zprovoznit operační systém
- Guru: programování; datová bezpečnost; simulace; opravy hardware; Linux je pro vás standard; používáte vim nebo Bash
- Ajták: 01001110 01100101 01101110 11101101 00100000 01110000 01110010 01101111 01100010 01101100 11101001 01101101

1.2 Mimochodem, který prohlížeč preferujete?

- Internet Explorer (modré éčko)
- Mozilla Firefox (liška a zeměkoule)
- Chrome (červená, modrá, žlutá, a zelená kulička)
- Safari (kompas)
- Opera (červené óčko)
- Jiný, který?

Kombinaci:

1.3 A kdy naposledy jste ho aktualizovali?

- U mě automaticky, nestarám se
- Tak tohle vůbec nevím
- V posledním měsíci
- V posledních 3 měsících
- V posledních 6 měsících
- Maximálně rok zpátky
- Před dávnými časy, v jedné předaleké galaxii...

1.4 Když už jsme u toho, co operační systém?

- Microsoft Windows
- Mac OS X
- Linux, jaké distro?
- Jiný, který?

Kombinaci:

1.5 Víte, co znamená zkratka HTTPS (ne význam písmen, ale co vám říká?)

- Znamená, že webová stránka je na Internetu
- To je, že webová stránka je určitě zabezpečená (nebo něco takového)
- Že stránka může být zabezpečená
- Netuším a ani jsem to nikdy nikde neviděl/a
- Někde jsem možná zahlédl/a, ale nezajímám se, co zkratka znamená

1.6 Myslete usilovně na heslo, které používáte ke svému nejdůležitějšímu účtu (ale ne PIN pro bankovníctví).
Kolik má celkem znaků?

- 6 a méně
- 12–16
- 22 a více, protože jsem paranoidní:)
- 7–11
- 17–21

1.7 Kdy naposledy jste si ho změnili?

- V posledních 30 dnech
- Ne déle než 6 měsíců
- Ještě nikdy, je moje první
- Něco mezi 1–3 měsíci
- Rok, asi rok
- Něco jiného?

1.8 Kolik obsahuje...

malých písmen:
velkých písmen:
speciálních znaků (mřížka, podrtržítka, zavináč, atd.):
mezer:

1.9 Když vám teď za vaše heslo a přihlašovací údaje nabídnou řekněme 100 Kč, dáte mi je? Hned se taky přesvědčím, jestli jsou správné nebo ne.

- Dám, bez problémů
- Zbláznil jste se? Ne!

Nabídněte tuhle částku (české koruny) a pláceme si, ale rozumně:

1.10 Jak jste si své heslo vybrali?

- Jméno zvířátka, manžela/manželky, dítěte, města, datum narození, oblíbený film, písnička, hláška, slovo ze slovníku atd. tak jak leží a běží
- První odpověď, ale něco jste přidali před nebo za (čísla, písmena, symboly)
- První odpověď, ale některé znaky jste zadali dvakrát (Aniccka) nebo jste je nahradili něčím, co vypadá podobně (Anicka na 4n1ck4)
- Prostě jste na klávesnici naťukal/a nějaké znaky

Vygenerovali jste ho na Internetu nebo v nějakém programu, kde nebo v jakém?

Jinak:

1.11 Poslední otázka, slibuju: jak ho uchováváte? Můžete označit více než jednu odpověď (příště budu značit jako 1+).

- Pamatuji si ho a používám na různých stránkách
- Pamatuji si ho a nepoužívám stejné na různých stránkách
- Mám ho napsané a lísteček nosím pořád u sebe
- Mám ho napsané někde, kde je mi vždycky na očích

Používám speciální program, mám ho zašifované. Který to je?

Jiný způsob?



NONE



2013UTBFAMEMUSKMSARGACZE



3420821876 0002

2 O mobilních telefonech

Tady se vám zeptám, jak a co používáte na svém mobilu. Sice bych se rád zaměřil na uživatele smartphonů a tabletů, ale není problém, jestli ani jeden nemáte. Odpovídejte tak, jako byste měli.

2.1 Takže první otázka: máte smartphone/tablet nebo si plánujete pořídit?

- Ano
- Ne

2.2 Jaký operační systém preferujete?

- Žádný, je mi to jedno
- iOS (iPad, iPhone)
- Android (Samsung, Huawei, HTC, atd.)
- Windows Phone (Nokia, Surface, atd.)

Jiný:

2.3 A proč? (1+)

- Vypadá dobře
- Značka
- Měl/a jsem už jeden dříve
- Dostal/a jsem ho
- Dobré aplikace
- Rodina/přátelé ho mají taky
- Měl/a jsem jiný a chci změnu

Jiný důvod:

2.4 Co na smartphonu/tabletu používáte (budete používat) nejčasteji? (1+)

- Email, internet na wi-fi nebo mobilním připojení
- Základní věci: SMS, volání, MMS, žádný internet
- Aplikace
- Hudba, sledování a natáčení filmů

Jiné:

2.5 Použili jste (byste) smartphone/tablet pro internetové bankovníctví nebo placení přes Internet?

- Jasně, nevidím v tom problém
- Ne, nevím, ale radši používám PC/notebook

Občas ano, ale jen když nemám k dispozici PC/notebook. Kolikrát tak měsíčně?

2.6 A co funkce ve smartphonech/tabletech, jsou srovnatelné s běžnými PC (kromě toho, že ty si nemůžete dát do kapsy)?

- Umí toho přibližně stejně
- Mobily toho umí více
- Mobilům některé funkce chybí

2.7 Máte v mobilu uložená hesla?

- Ano, v poznámce nebo upomínce
- Ne, to vůbec

Ano, zašifrované. Jak?

Ano, jinak:

2.8 Vyžaduje váš mobil kromě PINu k SIM kartě heslo nebo kód pro odemknutí?

- Ne, nemám ho nastavený
- Ano, čísla a písmena
- Ano, vyžaduje, jen čísla
- Ano, čísla a písmena a speciální znaky včetně mezery

2.9 Vadilo by vám pro pracovní věci ve vašem telefonu používat profily (program, který mu řekne, co může a co nesmí)?

- Ano, vadilo. Vždyť je to můj telefon!
- Ne, v pohodě
- Ne, ale jenom pokud by se dal po práci nebo kdykoliv jindy vypnout
- Nevím, asi kdyby to firma chtěla, tak bych musel/a

2.10 Víím, že tenhle typ otázek není moc oblíbený, protože je těžké vybrat tu nejlepší odpověď (jaký je třeba rozdíl mezi “silně preferuji” a “preferuji”?), ale zkusíme. Bude to poprvé a naposledy. Takže: u každé vlastnosti zaškrtněte, jak moc ji u telefonu preferujete

Bezpečnost	Preferuji silně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nepreferuji vůbec
Cena	Preferuji silně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nepreferuji vůbec
Funkce/aplikace	Preferuji silně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nepreferuji vůbec
Vzhled	Preferuji silně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nepreferuji vůbec
Značka	Preferuji silně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nepreferuji vůbec

2.11 A poslední: tipněte, kolik zranitelností (ty můžou skončit tím, že se vám něco špatného stane s počítačem nebo mobilem) bylo podle zprávy společnosti Symantec, známého to výrobce antivirového software, odhaleno v roce 2012? Buďte v klidu, víím, že budete hádat, jde mi o to, jestli hádáte moc, málo, nebo tak akorát. Malou nápovědu? V tisících.

Dobře, tady to je, ale jste si doufám vědom, že střílím od boku:

3 A ještě...

Mám tady pár věcí, které se nehodily nikde jinde. Nebojte, za chvíli už bude hotovo.

3.1 Dostali jste někdy email s prezentací nebo odkazem, který když jste otevřeli, stalo se něco špatného (virus atd.)?

- Ne, nikdy takové emaily neotvírám
- Párkrát se to stalo, ale nejde odolat a otevírám je pořád, všechny
- Ano, a teď už jsem mnohem opatrnější
- Stalo se mi to, a teď otevírám emaily jenom od lidí, kterým důvěřuji

3.2 Vííte, co je spam a phishing?

- Nevím ani jedno, kolik těch otázek ještě bude?!

No jasně, spam je:

Ehm, phishing je:

3.3 Hypoteticky: co kdybych vám řekl, že můžete na Internetu velmi snadno dělat nelegální věci (ne, nemyslím stahování hudby a filmů, horší). Lákalo by vás začít je dělat? (Tato otázka může být problematická na představivost, ale zkuste.)

- Ne, i kdybych věděl/a jak, nešel/la bych do toho protože je to špatné (a měl/a bych strach)
- Ano, ale jenom pod podmínkou, že by mě nechytli
- Ano, ale muselo by z toho něco kápnout (peníze nebo tak)

Mám jiný názor:



NONE



2013UTBFAMEMUSKMSARGACZE



3420821876 0004

4 Informace o vás

Abych byl schopen provést něco s informacemi, které jste mi (snad) poskytli, potřebuji se vás na závěr zeptat na pár otázek, abych vaše odpovědi mohl rozřadit do kategorií podle:

4.1 Pohlaví:

- Žena
- Muž

4.2 Věku:

- 18 let a méně
- 19–25 let
- 26–35 let
- 36–45 let
- 46–55 let
- 56–65 let
- 66 let a více

4.3 Ekonomické aktivity:

- Studující
- Pracující student
- Pracující
- Nezaměstnaný/á
- V důchodu
- V důchodu pracující

4.4 Měsíčního příjmu:

- 0 (nezaměstnaný, student)
- 15 000 Kč a méně
- 15 001–25 000 Kč
- 25 001–35 000 Kč
- 35 001–45 000 Kč
- 45 001–55 000 Kč
- 55 001 Kč a více

Díky moc za trpělivost, vím, že toho bylo hodně, ale když jsem jednou dostal příležitost na něco se zeptat, nemohl jsem odolat:) Ještě, prosím, pod tento text uveďte své celé jméno a podpis. Pokud si přejete být informováni o výsledcích, připojte také svou emailovou adresu (pro jistotu hůlkovým písmem). Dostanete je koncem tohoto nebo začátkem příštího roku v PDF, pokud se poštěstí, tak i s komentářem. Jsem vám k dispozici na emailové adrese a telefonu, oba najdete níže.

A poslední věc: všechny informace, které jsem se od vás dozvěděl (a za které ještě jednou děkuji) nebudu za milióny nebo dovolené v Karibiku nikomu prodávat, ani se s nimi chlubit na veřejnosti, maximálně ve formě: “Přibližně polovina respondentů se domnívá, že...,” ale nic ve smyslu: “Honza Novák má nový smarthonpe a používá ten a ten prohlížeč.” Dotazníky budu zpracovávat pouze já, nikdo jiný, takže se ani nemusíte bát, že bych se nad nimi s někým dalším bavil.

Přeji pěkný den.

Ing. Libor Sarga, Ústav statistiky a kvantitativních metod,
Fakulta managementu a ekonomiky, Univerzita Tomáše Bati ve Zlíně,
Mostní 5139, 760 01 Zlín
sarga[at]fame.utb.cz, +420 576 032 817 (pevná linka)

Vaše jméno a příjmení, podpis, volitelně emailová adresa:

4.5 (Dobrovolné) Prostor pro připomínky, poznámky, kritiku, chválu, komplimenty, apod.



NONE



2013UTBFAMEMUSKMSARGACZE



3420821876 0006

Appendix B: Truth Table for Third-Party Software Module Dependency Violation Problem

The following table pertains to the problem devised in chapter 5.1. The scenario presents five interdependent third-party modules which need to be updated while keeping the system fully functional. System stability evaluates to “Yes” if the dependent module is updated before or at the same time as the authoritative one; “No” otherwise.

A	B	C	D	E	System stable?
0	0	0	0	0	Yes
0	0	0	0	1	Yes
0	0	0	1	0	Yes
0	0	0	1	1	Yes
0	0	1	0	0	No
0	0	1	0	1	No
0	0	1	1	0	No
0	0	1	1	1	Yes
0	1	0	0	0	No
0	1	0	0	1	No
0	1	0	1	0	No
0	1	0	1	1	No
0	1	1	0	0	No
0	1	1	0	1	No
0	1	1	1	0	No
0	1	1	1	1	Yes
1	0	0	0	0	No
1	0	0	0	1	No
1	0	0	1	0	No
1	0	0	1	1	No
1	0	1	0	0	No
1	0	1	0	1	No
1	0	1	1	0	No
1	0	1	1	1	No
1	1	0	0	0	No
1	1	0	0	1	No
1	1	0	1	0	No
1	1	0	1	1	No
1	1	1	0	0	No
1	1	1	0	1	No
1	1	1	1	0	No
1	1	1	1	1	Yes

Appendix C: Source Code for Graphs

The following source code was used to generate graphs in the R Programming Language environment.

Figure 74:

```
library(ggplot2)
datax<-c("01","02","03","04","05","06","07","08","09","10")
datay<-c(80.61,67.24,40.50,38.81,16.02,15.12,12.03,5.73,1.19,0.32)

qplot(datax,datay,xlab="Regular expression",ylab="Percentage",geom="
  histogram",stat="identity")
```

Figure 76:

```
data<-read.csv("in.txt")
out<-apply(data,2,nchar)

hist(out,xlab="Password length",ylab="Frequency",main="",col="gray26")

obj1<-hist(out,xlab="Password length",ylab="Frequency",main="",col="
  gray26")
obj1
```

Figure 78:

```
data2<-read.csv("in2.txt")
out2<-apply(data2,2,nchar)

hist(out2,xlab="Password length",ylab="Frequency",main="",col="gray26")

obj2<-hist(out2,xlab="Password length",ylab="Frequency",main="",col="
  gray26")
obj2
```

Figure 80:

```
data3<-read.csv("in3.txt")
out3<-apply(data3,2,nchar)

hist(out3,xlab="Password length",ylab="Frequency",main="",col="gray26")

obj3<-hist(out3,xlab="Password length",ylab="Frequency",main="",col="
  gray26")
obj3
```

Figure 81:

```
data4<-read.csv("in4.txt")
out4<-apply(data4,2,nchar)
breaks<-seq(5,9,1)
```

```
hist(out4,xlab="Password length",ylab="Frequency",main="",breaks=breaks,
     col="gray26")
```

```
obj4<-hist(out4,xlab="Password length",ylab="Frequency",main="",breaks=
  breaks,col="gray26")
obj4
```

Figure 82:

```
hps<-sort(c(0.5895,257.502,0.1513,0.3425,0.0723))
sph<-sort(c(1.6963,0.0039,6.6076,2.9198,13.8267))
labhps<-c("PRM","RB","TC","BF","STR")
labsph<-c("STR","BF","TC","RB","PRM")

barplot(hps,axes=TRUE,names.arg=labhps,col="gray26",xlab="Mode",ylab="
  Hashes per second",main="")
barplot(sph,axes=TRUE,names.arg=labsph,col="gray26",xlab="Mode",ylab="
  Seconds per hash",main="")
```

Figure 83:

```
abs<-sort(c(12957,65836,119996,144347,15261,190289))
rel<-sort(c(11.9988,21.8697,26.3078,2.3611,2.7814,34.6809))
lab<-c("TC","PRM","BF","STR","RB","Remaining")

barplot(abs,axes=TRUE,names.arg=lab,col="gray26",xlab="Mode",ylab="
  Absolute frequency",main="")
barplot(rel,axes=TRUE,names.arg=lab,col="gray26",xlab="Mode",ylab="
  Relative frequency",main="")
```

Appendix D: Questionnaire Password Composition Frequency Tables

The following frequency tables were constructed from Q1.8 in the questionnaire survey in chapter 4, and pertain to bar charts depicted in chapter 4.2 and are presented separately due to their size. The first table quantifies distribution of lowercase characters the respondents reported in a password which protects their most important account which they were not asked to specify.

q1.8I

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	26	3,3	3,7	3,7
	1	15	1,9	2,2	5,9
	2	31	4,0	4,4	10,3
	3	40	5,1	5,7	16,1
	4	84	10,7	12,1	28,1
	5	104	13,3	14,9	43,0
	6	136	17,3	19,5	62,6
	7	71	9,1	10,2	72,7
	8	67	8,5	9,6	82,4
	9	35	4,5	5,0	87,4
	10	41	5,2	5,9	93,3
	11	11	1,4	1,6	94,8
	12	9	1,1	1,3	96,1
	13	6	,8	,9	97,0
	14	5	,6	,7	97,7
	15	6	,8	,9	98,6
	16	4	,5	,6	99,1
	17	2	,3	,3	99,4
	18	1	,1	,1	99,6
	19	1	,1	,1	99,7
20	2	,3	,3	100,0	
	Total	697	88,9	100,0	
Missing	999	87	11,1		
Total		784	100,0		

The second table quantifies distribution of uppercase characters in the same password.

q1.8u

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	239	30,5	34,2	34,2
	1	226	28,8	32,4	66,6
	2	104	13,3	14,9	81,5
	3	40	5,1	5,7	87,2
	4	26	3,3	3,7	91,0
	5	22	2,8	3,2	94,1
	6	20	2,6	2,9	97,0
	7	7	,9	1,0	98,0
	8	3	,4	,4	98,4
	9	5	,6	,7	99,1
	10	3	,4	,4	99,6
	11	1	,1	,1	99,7
	12	2	,3	,3	100,0
	Total	698	89,0	100,0	
Missing	999	86	11,0		
Total		784	100,0		

The third table quantifies distribution of special characters (!, #, \$, %, ^, &, *, _, {, }, \, etc.) in the same password.

q1.8spec

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	377	48,1	54,0	54,0
	1	94	12,0	13,5	67,5
	2	90	11,5	12,9	80,4
	3	46	5,9	6,6	87,0
	4	53	6,8	7,6	94,6
	5	10	1,3	1,4	96,0
	6	13	1,7	1,9	97,9
	7	3	,4	,4	98,3
	8	3	,4	,4	98,7
	9	2	,3	,3	99,0
	10	5	,6	,7	99,7
	11	1	,1	,1	99,9
	12	1	,1	,1	100,0
	Total	698	89,0	100,0	
Missing	999	86	11,0		
Total		784	100,0		

The fourth table quantifies distribution of spaces in the same password.

q1.8spc

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	666	84,9	95,3	95,3
	1	27	3,4	3,9	99,1
	2	4	,5	,6	99,7
	3	2	,3	,3	100,0
	Total	699	89,2	100,0	
Missing	999	85	10,8		
Total		784	100,0		

CURRICULUM VITAE

Personal Information

Full name: Libor Sarga

Date of birth: April 11, 1986

Residence: Heřmanická 34, 710 00 Ostrava, Czech Republic

Email: sarga@fame.utb.cz

Marital status: Single

Education History

2010-07 – 2014-06 (exp.): Tomas Bata University in Zlín; degree: Ph.D.

2008-10 – 2010-05: Tomas Bata University in Zlín; degree: Ing.

2005-09 – 2008-06: Philosophical Faculty, Palacký University; degree: Bc.

Exchange Programmes

2013-09 – 2014-01: Center for Massive Data Algorithmics (MADALGO), Aarhus University, Aarhus, Denmark (The Erasmus Programme)

Employment History

2013-02 – 2014-08 (exp.): Tomas Bata University in Zlín, Faculty of Management and Economics, Department of Statistics and Quantitative Methods; position: department secretary

Teaching History

2010 – 2014:	Applied Statistics (Czech)
2014:	Applied Marketing Research (Czech)
2014:	Managerial Decision-Making and Risk Management (English)
2011:	Quantitative Decision-Making Methods (Czech, English)

Language Skills

Czech:	Native speaker
English:	Fluent

Research Projects Participation

2010–2012:	MEB051024 Information logistics of transport, production and storing systems
2010–2012:	Competency Based e-portal of Security and Safety Engineering – eSEC, 502092-LLP-1-2009-1-SK-ERASMUS-EMHE
2012:	CreaClust – A Cross-Border Cluster Initiative for the Development of Creative Industry, 22410420020
2012–2014 (exp.):	Centre for Security, Information and Advanced Technologies (CEBIA-Tech), CZ.1.05/2.1.00/03.0089

Zlín, February 28, 2014

“I think that technologies are morally neutral until we apply them. It’s only when we use them for good or evil that they become good or evil.”

– William Gibson