

Operační systémy - Security

Lucie Galdová

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Ráda bych zde poděkovala vedoucímu mé bakalářské práce Ing. Martinu Syslovi Ph.D. za odborné vedení, za připomínky a čas věnovaný mé práci.

Prohlašuji, že jsem na celé diplomové práci pracovala samostatně a použitou literaturu jsem citovala.

Ve Zlíně, 06. 06. 2005

.....

jméno diplomanta

ABSTRAKT

Abstrakt česky

Práce popisuje základní možnosti ochrany dat a operačního systému. V obecné rovině se zabývá šifrováním dat, prokazováním identity a otázkou přístupových práv. Předkládá problém napadení bezpečnosti a psychologických útoků a nabízí některé možnosti, jak tento problém řešit.

Praktická část se zabývá problémem bezpečnosti hesla v operačním systému Microsoft Windows v závislosti na jeho délce a složitosti. K tomuto účelu byl využit program L0phtcrack4.

Abstrakt ve světovém jazyce

This work discusses basic possibilities of data protection and security of operating system. It deals with cryptography, authentication and access control in a general level. It calls attention to security attacks and psychological attacks and try to find a solution of these problems.

The practical part deals with a problem of password protection in operation system Microsoft Windows in dependence to length and complexity of password. For this purpose it was used the program L0phtcrack4.

OBSAH

ABSTRAKT	3
OBSAH	4
ÚVOD.....	6
I. TEORETICKÁ ČÁST	7
1 KRYPTOGRAFIE	8
1.1 TAJNÝ KLÍČ (SECRET-KEY CRYPTOGRAPHY)	8
1.2 VEŘEJNÝ KLÍČ (PUBLIC-KEY CRYPTOGRAPHY).....	10
2 PROKAZOVÁNÍ IDENTITY	11
2.1 HESLA	11
2.2 JEDNORÁZOVÝ PŘÍSTUP (SINGLE SIGN-ON).....	12
2.3 BIOMETRIE	13
2.3.1 Snímání otisků prstů.....	13
2.3.2 Snímání oční sítnice a duhovky	14
2.3.3 Další techniky.....	14
2.4 ČIPOVÉ KARTY	14
2.5 DIGITÁLNÍ PODPIS	15
3 ŘÍZENÍ PŘÍSTUPU	17
3.1 PŘÍSTUPOVÁ PRÁVA A OCHRANNÉ DOMÉNY	17
3.1.1 Přístupová práva v UNIXu.....	17
3.1.2 Přístupová práva ve Windows XP Professional.....	20
3.2 BEZPEČNOSTNÍ MODELY A POLITIKA	24
4 NAPADENÍ BEZPEČNOSTI.....	26
4.1 KRYPTOANALÝZA	26
4.2 VIRY.....	27
4.3 ČERVI.....	28
4.4 TROJSKÉ KONĚ	28
4.5 ÚTOKY DOS (DENIAL-OF-SERVICE).....	29
4.6 PSYCHOLOGICKÉ ÚTOKY	30
4.6.1 Hoax	30
4.6.2 Spam.....	31
4.7 NEVYŽÁDANÝ SOFTWARE	31
4.7.1 Příznaky nevyžádaného softwaru.....	33
5 PREVENCE ÚTOKU A ŘEŠENÍ BEZPEČNOSTI	34
5.1 ANTIVIROVÝ SOFTWARE.....	34
5.1.1 Vyhledávání v databázi (Signature-scanning virus detection).....	34
5.1.2 Heuristická analýza	34

5.1.3	Kontrola integrity	35
5.1.4	Rezidentní sledování	36
5.2	FIREWALLY	36
5.2.1	Paketové filtry	36
5.2.2	Aplikační brány	37
5.3	BEZPEČNOSTNÍ ZÁPLATY	38
5.3.1	Windows Update	38
5.3.2	Linux Update	39
6	STANDARD BEZPEČNOSTI - ORANGE BOOK.....	42
II.	PRAKTICKÁ ČÁST	44
	L0PHTCRACK.....	45
	ZÁVĚR.....	51
	SEZNAM POUŽITÉ LITERATURY	52
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	55
	SEZNAM OBRÁZKŮ	56
	SEZNAM TABULEK.....	57
	SEZNAM PŘÍLOH.....	58

ÚVOD

V dnešní době je na bezpečnost dat kladen velký důraz. Mnoho organizací vlastní hodnotné informace, jejichž vyzrazením či ztrátou by organizace přišla o miliony dolarů nebo o významné postavení na trhu. Tyto informace jsou proto pečlivě střeženy a na jejich ochranu jsou vynakládány nemalé finanční prostředky.

S důležitostí střežených dat ale narůstá také počet lidí, kteří se pokouší bezpečnost systému podniku narušit. Ať už je důvodem snaha poškodit organizaci nebo snaha ověřit si, že to dokáží.

Cenné informace se mohou ztratit také neopatrným nebo neodborným zacházením se systémem nebo zničením média, na němž jsou informace uchovávány. Větší množství kopií se hůře hlídá a také hůře udržuje aktuální. Je proto nutné najít vhodný kompromis, který možnost znehodnocení a prozrazení informací minimalizuje.

Pak je tu také otázka důvěry. Veškerá zabezpečení jsou zbytečná, pokud osoba, které byl povolen přístup k informacím, tyto informace prozradí. Nebo pokud nedodrží zásady bezpečnosti a případný útočník se k informacím dostane přes její uživatelský účet.

Existuje mnoho možností, které mají pomoci zabezpečit střežené informace. Antivirové programy mají zabránit útoku škodlivého softwaru, šifrování dat má znemožnit jakémukoli neoprávněnému uživateli data přečíst, zabezpečení řízeného přístupu zajišťuje, že s daty nebude manipulovat nikdo, kdo k tomu nemá příslušná přístupová práva a správné nastavení moderních operačních systémů a firewallů má zabránit útokům hackerů.

Protože ani operační systémy nejsou zcela bez chyb, je nutné pro operační systém stahovat nejnovější opravné balíčky a aktualizace.

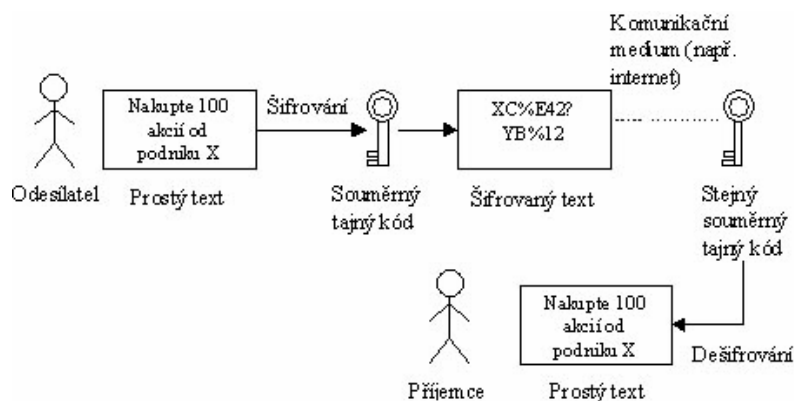
I. TEORETICKÁ ČÁST

1 KRYPTOGRAFIE

Důležitým cílem práce s počítači je snadnější dostupnost obrovského množství dat. Ale přenos elektronických dat, zejména přes veřejné linky, je nezabezpečený. Řešením tohoto problému je vytvoření takových dat, aby byla nečitelná pro neautorizované uživatele. Toho můžeme dosáhnout zašifrováním dat. Šifrování se zaměřuje na kódování a dekódování dat tak, že mohou být přeložena pouze oprávněným příjemcem. Data jsou přeměněna použitím šifer nebo kryptosystémem - matematickým algoritmem pro šifrování zpráv. Klíč, zastoupený řetězcem znaků, je vkládán do šifry. Algoritmus převádí nešifrovaná data, nebo prostý text, na šifrovaná data, nebo šifrovaný text. K tomu používá klíč jako vstup – různé klíče vytvoří různé šifrované texty.

1.1 Tajný klíč (Secret-key cryptography)

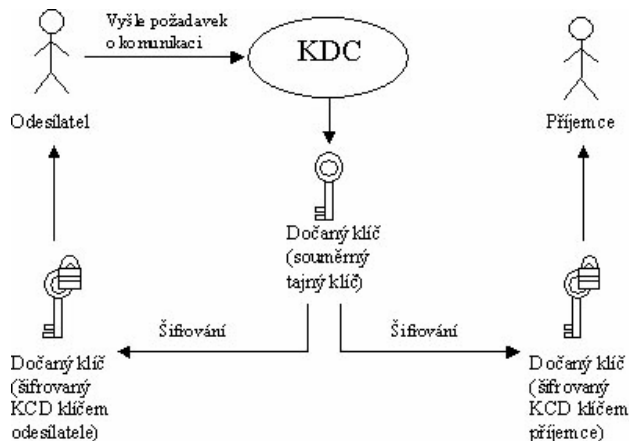
Symetrická kryptografie, také známé jako tajný klíč, používá stejný klíč ke kódování i k dekódování zprávy (Obr. 1). V tomto případě odesílatel zašifruje zprávu použitím tajného klíče, pak pošle zašifrovanou zprávu příjemci, který zprávu rozšifruje použitím stejného klíče. Podstatnou nevýhodou této metody je, že i klíč musí být bezpečně přenesen.



Obr. 1. Tajný klíč

Metoda souměrného šifrování je přijatelná pro komunikaci dvou stran, ale není příliš vhodná pro rozlehlé sítě. K udržení soukromé komunikace odesílatel potřebuje jiný klíč pro každého příjemce. Následkem toho by bezpečná práce ve velkých organizacích vyžadovala velké množství tajných kódů pro každého uživatele. Tento problém částečně řeší vytvoření centrální autority KDC (Key Distribution Center). KDC generuje dočasné

klíče (session keys) pro jednu transakci (Obr. 2). Pak doručí odesílateli a příjemci dočasný klíč, zašifrovaný tajným klíčem, který oba sdílí s KDC. Tím, že pro každou komunikaci vytváří KDC nový klíč, narůstá bezpečnost sítě.



Obr. 2. Dočasný klíč s využitím centrální autority
KDC

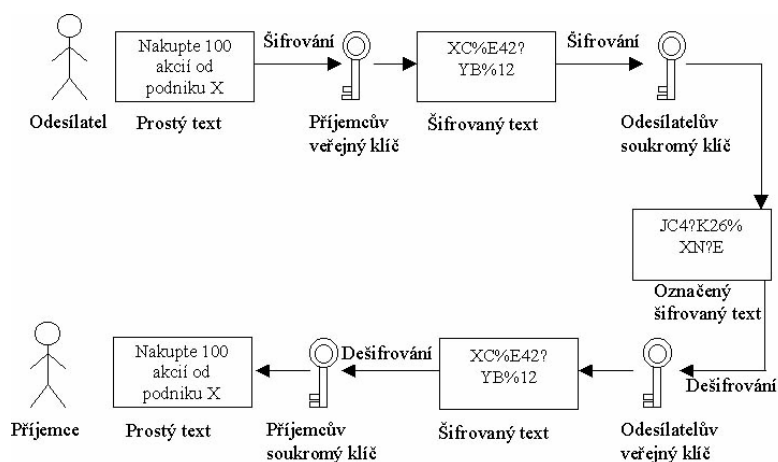
Jedním z nejvíce používaných souměrných šifrovacích algoritmů je DES (Data Encryption Standard) vytvořený v 70. letech. Má klíč dlouhý 56 bitů a šifruje data do 64-bitových bloků. Tento způsob šifrování je znám jako bloková šifra, protože ze zprávy vytváří skupiny bitů a šifrovací algoritmus aplikuje na bloky jako celek. V 90. letech byl vytvořen Triple DES, nebo 3DES – varianta DES, která je v podstatě třemi DES systémy v sérii a je bezpečnější. Každá část má jiný tajný klíč, který pracuje na bloku.

Alternativou k DES algoritmu je Blowfish. Jedná se o symetrickou blokovou šifru s velikostí bloku 64 bitů a délkou klíče nejvýše 448b (tj. 56B). Algoritmus je tvořen dvěma částmi: část expanze klíče a část šifrování dat. Expanze klíče převádí klíč s libovolnou délkou (nejvýše však 56B) na několik polí podklíčů, čítajících dohromady 4168B. Podklíče jsou vypočteny ze zadaného klíče ještě před samotným šifrováním, resp. dešifrováním dat. Šifrování dat je prováděno – po blocích 64 bitů – v šestnácti rundách. Každá runda provádí permutaci závislou na klíči a substituci závislou jak na kódovaných datech tak i klíči. [1]

Dnes je nejbezpečnějším a nejúčinnějším standardem šifrování AES (Advanced Encryption Standard) - Rijndael. Rijndael se může použít pro velikosti klíče a bloku 128, 192 a 256 bitů.

1.2 Veřejný klíč (Public-key cryptography)

Asymetrická kryptografie - kryptografie veřejného klíče používá dva druhy klíčů – veřejný a soukromý. Veřejný klíč slouží k šifrování dat a soukromý pro dešifrování. Oba účastníci vlastní oba druhy klíčů – každý svůj. K bezpečnému přenosu zprávy používá odesílatel příjemcův veřejný klíč. Příjemce rozšifruje zprávu pomocí jedinečného soukromého klíče. Zprávu tak může přečíst jen oprávněný příjemce. Nevýhodou je, že příjemce si nemůže být naprosto jist tím, že zprávu opravdu odeslal předpokládaný odesílatel. Pro autentizaci obou účastníků se zpráva zašifrovaná příjemcovým veřejným klíčem šifruje ještě jednou – odesílatelovým soukromým klíčem - a pak se rozšifruje opačným postupem (Obr. 3.).



Obr. 3. Veřejný klíč

Nejpoužívanější algoritmus asymetrické kryptografie je RSA (Rivest, Shamir, Adleman). Bezpečnost RSA je založena na skutečnosti, že je obtížné rozložit velká čísla (z nichž každé je součinem dvou velkých prvočísel). Většina transakcí elektronického obchodu a komunikace na internetu používá RSA.

Šifrovací systém PGP (Pretty Good Privacy) šifruje e-mailové zprávy a soubory a může také zajistit digitální podpisy, které potvrzují identitu autora e-mailové zprávy.

2 PROKAZOVÁNÍ IDENTITY

Identifikace uživatelů a akcí, které jsou oprávněni vykonávat je zásadní pro udržení bezpečnosti počítačového systému. Uživatel může být identifikován podle:

- Specifické charakteristiky člověka (otisky prstů, hlas, obraz sítnice)
- Vlastnictví předmětu (odznak, identifikační karta, klíč, čipová karta)
- Znalostí uživatele (heslo, PIN, kombinace zámku)

2.1 Hesla

Nejběžnějším způsobem ověření identity je ochrana heslem. Uživatel si vybere heslo, zapamatuje si jej a zadává jej k získání přístupu ke zdroji nebo k síti. Ochrana heslem ale není zcela bezpečná. Uživatelé volí hesla, která si snadno zapamatují (jména blízkých, data narození) a jiní uživatelé je mohou uhádnout. Při užívání složitějších hesel si je uživatelé často poznamenají a tím také ohroží bezpečnost chráněného systému.

Při „crackování hesel“ se používají následující typy útoků:

- **Slovníkový útok** (dictionary attack) – zkouší se hesla ze slovníku (zpravidla několik desítek tisíc hesel). Tento typ útoku nemá přímou souvislost s použitým algoritmem, snaží se využít chyby lidského faktoru při volbě hesla. Je velmi rychlý, ale nepříliš často úspěšný.
- **Útok hrubou silou** (brute-force attack) – zkouší se všechny možné klíče definované abecedy. Rychlost útoku (udává se v počtu odzkoušených hesel za sekundu) závisí na použitém šifrovacím algoritmu. Pro větší délky klíče je neúnosně pomalý.
- **Útok se známým otevřeným textem** (known-plaintext attack) – jde o typ útoku, při kterém známe původní nezašifrovaná data (většinou stačí jeden soubor z celého archivu) a snažíme se zjistit příslušné heslo. Čas potřebný ke zjištění hesla můžeme tímto podstatně zkrátit.

Silná hesla podstatně prodlouží dobu nutnou k dešifrování hesla, nebo dešifrování zcela zabrání.

Pro vytvoření silného hesla existuje několik základních pravidel:

- Nepoužívat slova ze slovníku
- Použití nejméně sedmi znaků
- Používat znaky alespoň ze tří znakových sad (A-Z, a-z, 0-9, zvláštní znaky: @, &, #, ! atd.),
- Používat i netisknutelné znaky za pomoci Alt+hodnota z ASCII tabulky (např. známý nástroj na crackování hesel L0phtcrack s těmito znaky vůbec nepočítá)

Průnik do obrany operačního systému neznamena nutně jednoznačné ohrožení bezpečnosti. Většina operačních systémů šifruje soubor hesel. Pro další maření útoku používají některé systémy tzv. password salting – metodu, která vkládá další znaky na různé pozice v hesle ještě před vlastním šifrováním (Tab. 1.). Obranou proti brute-force crackingu může být omezení počtu pokusů o zadání hesla.

Tab. 1. Metoda password salting

Text	Šifrovaný text
password	cGFzc3dvcmQ=
psasaswlord	cHNhc2Fzd2xvcnRk
newpassword	bmV3cGFzc3dvcmQ=
nsewaplatssewodrd	bnN1d2FwbGF0c3N1d29kcmQ=

2.2 Jednorázový přístup (Single Sign-On)

Systémy Single Sign-On zjednodušují identifikační proces tím, že dovolí uživateli zalogovat se jen jednou, užitím jednoho hesla. Uživatelé se tak mohou dostat k mnoha aplikacím na mnoha počítačích.

Nejjednodušší formou Single Sign-On jsou Workstation login scripts. Uživatel se zalogue na pracovní stanici a vybere si aplikaci z menu. Hesla jsou často ukládána v klientově počítači jako prostý text, takže bezpečnost tohoto typu Single Sign-On je nedostatečná.

Authentication server scripts identifikují uživatele přes centrální server. Ten kontroluje spojení mezi uživatelem a aplikací. Authentication server scripts jsou bezpečnější než workstation login script, protože uživatelské heslo je uloženo na serveru.

Nejpokročilejší Single Sign-On systémy používají tzv. token based authentication. Jakmile je uživatel identifikován, je vytvořen jedinečný token, aby umožnil uživateli přístup ke specifickým aplikacím.

V současné době jsou v čele vývoje technologie Single Sign-On tři společnosti - Liberty Alliance Project, Microsoft a Novell.

2.3 Biometrie

Biometrie využívá specifických charakteristik člověka – otisky prstů, snímky sítnice nebo tváře – k identifikaci uživatele. Tento způsob identifikace je mnohem bezpečnější a uživatelsky pohodlnější. Při zalogování do sítě nebo při přístupu k zabezpečeným informacím se porovnává otisk nebo snímek ze scanu s databází. Některé systémy využívají kombinace hesel a biometrických údajů.

2.3.1 Snímání otisků prstů

Otisky prstů jsou pro svoji spolehlivost již velmi dlouhou dobu používány pro identifikaci osob v kriminalistice a za tuto dobu se rozvinul značný počet metod, které dovolují pracovat i s neúplnými daty, umožňují vytváření vzorů otisků s dostatečnou kompresí dat beze ztráty identifikátorů a podobně. Rovněž existuje množství algoritmizovatelných postupů, automatických a poloautomatických systémů pro jejich ověřování. Na trhu je k dispozici celá řada komerčních produktů, které dovolují používat rozpoznávání otisků prstů pro ověření přístupu k počítačovým systémům. Jde vždy o spojení snímače otisku prstu s kontrolním programem. V poslední době se kromě klávesnic se snímačem otisků začínají objevovat dokonce řady notebooků s již integrovaným snímačem otisků prstů, firma Compaq dokonce nabízí pro řadu svých notebooků externí snímač otisků prstů.

Snímače otisků jsou založeny na principu elektrického, optického, ultrazvukového, tepelného či tlakového snímání otisku. Nejstarší křemíkové snímače využívají změny elektrické kapacity mezi dvěma body snímače, podle toho, je-li mezi nimi papilární čára nebo mezera. Jde o levné snímače, jejichž spolehlivost však může být značně ovlivněna rozdíly ve vlhkosti pokožky.

Dnes nejrozšířenější jsou optické snímače, využívající změny odrazu světla od částí prstu. Jsou poměrně levné, jejich spolehlivost je však silně závislá na čistotě snímací plochy.

Nejnovějším typem jsou ultrazvukové snímače, které jsou značně spolehlivé, pro svou vyšší cenu však relativně málo rozšířené.

Všechny tyto metody snímání se kvůli zvýšení spolehlivosti často kombinují s tepelnými a tlakovými senzory.

Nevýhodou identifikace podle otisků prstů je možnost oklamání snímače pomocí návleku na prstu, který při použití vhodné technologie vykazuje všechny znaky živého prstu, tedy propouští teplo, pulsaci krve a má správnou vlhkost.

2.3.2 Snímání oční sítnice a duhovky

Snímání oční sítnice je technikou, jež vznikla v polovině 80. let na základě lékařských poznatků, že rozložení cév a vlasečnic na pozadí oční sítnice se během života jedince nemění. Jde sice o velmi spolehlivý způsob identifikace, vyžaduje však využití speciálního laserového snímače a nepříliš příjemné metody snímání, což snižuje pravděpodobnost rozsáhlejšího nasazení.

Podobnou technikou je identifikace podle oční duhovky. Struktura duhovky se během lidského života s výjimkou některých onemocnění nemění.

2.3.3 Další techniky

Existuje ještě celá řada dalších technik, které se snaží určovat identitu za pomoci co nejmenší obtížnosti pro uživatele a co nejmenšího využití speciálních technických zařízení. Příkladem může být například dynamika psaní na klávesnici, tedy měření časových prodlev mezi stisky jednotlivých kláves, nebo metoda založená na období běžného podpisu pomocí počítačové myši. [2]

2.4 Čipové karty

Čipové karty se používají pro potvrzení identity a uchování dat. Nejpopulárnější jsou paměťové karty a mikroprocesorové karty. Paměťové karty slouží k uchovávání a přemísťování dat. Mikroprocesorové karty obsahují počítačové komponenty řízené operačním systémem, který obstarává bezpečnost a paměť. Čipové karty mohou obsahovat soukromé klíče, digitální certifikáty, čísla kreditních karet apod. Každá čipová karta může

být použita pouze v kombinaci s PIN. Některé mikroprocesorové karty smažou uložená data pokud dojde k nedovolené manipulaci s kartou.

Čipové karty se podle svého zpracování dělí na dvě základní skupiny:

- kontaktní - obsahují na svém povrchu kontaktní plochy a pro svou činnost musí být vloženy do čtecího zařízení
- bezkontaktní - nevyžadují žádný fyzický kontakt, stačí jejich pouhá přítomnost v blízkosti čtecího zařízení; k přenosu dat a napájení používají indukční smyčku

Dalším alternativním řešením pro posílení zabezpečených uživatelských přístupů je iKey. Na rozdíl od smart card nevyžaduje zvláštní čtecí mechaniku. Je možno jej zasunout do standardního sériového konektoru typu USB. Z hlediska provozních nákladů je tudíž řešení verifikace uživatelů pomocí iKeys méně nákladné.

2.5 Digitální podpis

Digitální podpis představuje nejúčinnější prostředek pro zajištění integrity odesílaných dat (například zadání transakcí) a bezpečné ověření jejich odesílatele.

Digitální podpis má tyto vlastnosti:

- je autentický - příjemce s určitostí ví, kdo je autorem dokumentu
- je nezfalšovatelný
- je nepopíratelný
- má jednorázové použití - podpis nelze přenést na jiný dokument.
- nezměnitelnost dokumentu - po podpisu už nelze provádět v textu žádné změny

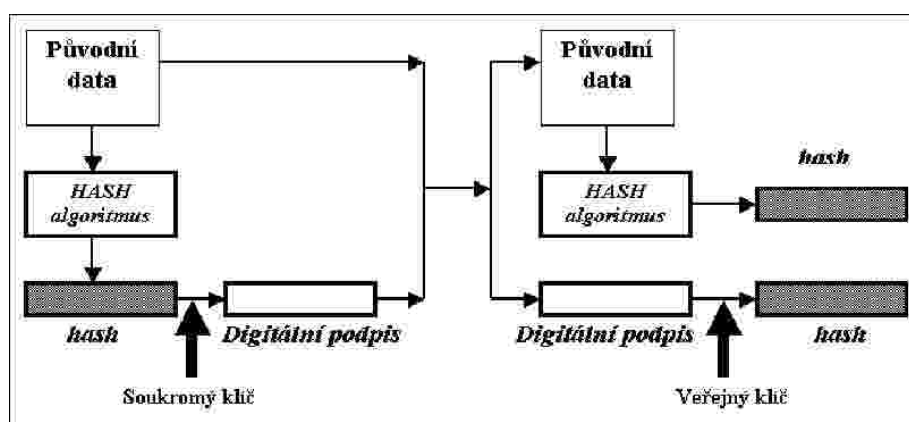
Pro zajištění bezpečnosti digitálního podpisu se využívá hash funkce. Je to jednocestná funkce, tzn. nelze z ní získat zpět původní text nebo data. Vstupem této funkce jsou data libovolné délky a výstupem je datový blok konstantní délky. Jakákoli změna dat změní hodnotu hashe.

Proces podepsání dokumentu představuje výpočet hashe dokumentu jednocestným algoritmem a zašifrování tohoto hashe privátním klíčem odesílatele. Šifra hashe (signature block) se připojí k dokumentu a odešle. Příjemce připojenou šifru dešifruje veřejným klíčem odesílatele, svými prostředky vypočte dohodnutým jednocestným algoritmem hash

dokumentu a srovná jej s hashem, jehož šifra byla k dokumentu připojena (Obr. 4.). Pokud jsou oba hashe analogické, je dokument považován za autentický dokument s neporušeným nemodifikovaným obsahem.

Příjemce má k dispozici pouze veřejný klíč odesílatele. Pouze veřejným klíčem odesílatele lze ověřit jeho digitální podpis. Tento klíč ovšem nemůže být použit pro vytvoření platného digitálního podpisu.

Nejpopulárnější hashovací funkce jsou MD (Message Digest), které vytváří 16-bytový výsledek, a SHA (Secure Hash Algorithm), vytvářející 20-bytový výsledek.



Obr. 4. Princip digitálního podpisu

3 ŘÍZENÍ PŘÍSTUPU

Dnešní operační systémy jsou vytvářeny tak, aby byly schopny ochránit svůj provoz a citlivé informace před uživateli nebo softwary, kteří mají přístup k počítačovým zdrojům. Přístupová práva chrání systémové zdroje a služby před potenciálně nebezpečnými uživateli tím, že zakazuje nebo omezuje akce, které by mohly narušit zdroje.

3.1 Přístupová práva a ochranné domény

Přístupová práva definují jak různé subjekty mohou přistupovat k různým objektům. Subjekty mohou být uživatelé, procesy programy apod. Objekty jsou zdroje jako hardware, software a data. Objekty musí být chráněny před subjekty. Kdyby procesy mohly přistupovat ke každému zdroji v systému, mohl by uživatel ohrozit bezpečnost systému. Aby se dalo předejít takovým událostem, každý subjekt musí mít autorizaci k přístupu k objektům uvnitř systému.

Ochranná doména je soubor přístupových práv. Každé přístupové právo v doméně je reprezentováno jako uspořádaná dvojice s poli pro jméno objektu a jeho práva. Například, pokud má uživatel právo číst a zapisovat do souboru pokus.txt, uspořádaná dvojice pro přístupová práva uživatele bude `<pokus.txt, {read, write}>`. Doména často komunikuje s jednotlivým uživatelem a upozorňuje jej, co může dělat a co ne.

Každý objekt může být ve více doménách současně a mít přitom pro každou doménu jiná přístupová práva.

Každý proces neustále běží na některé ochranné doméně. Jinými slovy, existují soubory objektů, které mohou přistupovat, a pro každý objekt je nastavena nějaká množina práv. Procesy se mohou mezi doménami přepínat i během jejich provádění. Pravidla pro přepínání mezi doménami jsou silně závislá na systému.

3.1.1 Přístupová práva v UNIXu

V UNIXu je doména procesu určena svým UID (identifikace vlastníka) a GID (identifikace skupiny, do které vlastník patří). Zadáním UID a GID kombinací je možné vytvořit kompletní seznam všech objektů (souborů, včetně I/O zařízení reprezentovaných zvláštními soubory, atd.), ke kterým je možné přistoupit, a jestli jsou přístupné pro čtení, psaní nebo spuštění.

drwxr-x--x

Označuje, že se jedná o adresář ('d'), vlastník má všechna práva ('rwx'), skupina má pouze práva 'r' a 'x', ostatní mají pouze právo 'x'.

Jednotlivá práva mají u souboru následující význam:

r - umožňuje čtení souboru (jeho zobrazení, zkopírování, tisk apod.).

w - umožňuje zápis do souboru (například textovým editorem nebo jiným programem)

x - umožňuje spuštění souboru jako programu

Navíc je nutné mít ve všech adresářích od kořene až po adresář, v němž se soubor nachází, právo 'x'.

Práva adresářů (s výjimkou práva 'x') neovlivňují možnost čtení, zápisu a spouštění souboru. Ovlivňují však možnost výpisu obsahu adresáře, vytváření, rušení a přejmenování souboru a podadresářů:

r - umožňuje vypsát jména souboru a podadresářů v adresáři bez jakýchkoli dalších údajů (nelze ani rozlišit, zda je dané jméno soubor nebo podadresář)

w - spolu s právem x umožňuje vytvářet, rušit a přejmenovávat soubory a podadresáře

x - umožňuje vypsát informace o souboru nebo podadresáři se známým jménem, je podmínkou pro všechny operace s obsahem adresáře kromě vypsání jmen

Rozšířená práva mají význam pro programy a adresáře. Jedná se o jedno právo pro každou z kategorií vlastník, skupina a ostatní:

s pro uživatele (setuid)

s pro skupinu (setgid)

t pro ostatní (sticky)

Příkaz 'ls -l' vypisuje u souboru nebo adresářů, které mají nastavena některá z rozšířených práv místo písmene 'x' (právo execute) písmeno 's' (u vlastníka nebo skupiny) nebo 't' (u ostatních). Pokud soubor nebo adresář nemá v příslušné kategorii právo 'x', je rozšířené právo znázorněno velkým písmenem 'S' nebo 'T'.

Program s nastaveným setuid bitem disponuje při běhu právy svého vlastníka. Díky tomu si mohou uživatelé měnit heslo, uložené v souboru '/etc/passwd', do kterého má právo zápisu pouze správce (vlastníkem je 'root' a jeho práva jsou 'rw-r--r--'). Program '/bin/passwd', který umožňuje změnu hesla, vlastní také 'root' a má nastaven setuid bit.

Podobně funguje právo setgid. Program popmail může vybírat dopisy z adresáře '/var/spool/mail', který vlastní 'root', skupina je 'mail' a jehož práva jsou 'rwxrwx-x', díky tomu, že 'popmail' má práva 'rwx--s--x' a skupinu 'mail'.

Změna práv souboru

Ke změně práv souboru slouží příkaz chmod. Jeho základní použití vypadá takto:

```
chmod práva souboru [ soubor ... ]
```

Práva mohou být zadána buď pomocí čísla v osmičkové soustavě nebo symbolickým zápisem.

Např. zápis:

```
chmod 751 pictures
```

je totožný se zápisem:

```
chmod u=rwx,g=rx,o=x pictures
```

a znamená práva rwxr-x--x.

V prvním případě první číslice udává práva vlastníka, druhá práva skupiny, třetí ostatních.

Právo r má hodnotu 4, právo w 2, a právo x 1.

Mají-li být zadána rozšířená práva je třeba použít čtyřmístné číslo.

Pro symbolický zápis je možné použít i nastavení např:

```
chmod a=rx pictures,
```

kde a znamená all (vlastník, skupina i ostatní)

Rovnítko znamená nastavení práv, na jeho místě může být znak +, který znamená přidání práv, nebo znak –, ubrání práv. Za znakem =, + nebo – může být jeden nebo více ze znaku rwxXst s významem:

r - read

w - write

x - execute

X - execute pouze pokud se jedná o adresář nebo pokud už má nastavené právo x pro některou kategorii

s - setuid nebo setgid a execute

t - sticky bit

nebo jeden ze znaků u, g, o, které znamenají nastavit práva podle práv vlastníka, skupiny nebo ostatních.

Příkaz umask zobrazuje nebo nastavuje implicitní masku práv pro nově vytvářené soubory. Bez parametru vypíše aktuální nastavení.

Hodnota se interpretuje jako osmičkové číslo jako v příkazu chmod, ale udává, která práva budou nově vytvářeným souborům nebo adresářům chybět. Přitom vytvářené soubory nikdy nemají nastavena práva execute (pokud mají být nastavena, je nutné to udělat dodatečně příkazem chmod).

3.1.2 Přístupová práva ve Windows XP Professional

Systém Windows XP Professional nabízí velmi jemné odstupňování řízení zabezpečení při přístupu k celé řadě objektů. Místní složka má k dispozici 14 oprávnění. Soubory a složky mají základní a zvláštní oprávnění.

Základní oprávnění k souborům a složkám

Číst – umožňuje vidět soubory a podsložky, zobrazit atributy složky, jeho vlastníka a oprávnění.

Zapisovat – umožňuje vytvářet nové soubory a podsložky, měnit atributy složky, zobrazit vlastníka a oprávnění složky.

Zobrazovat obsah složky – umožňuje zobrazit názvy souborů a podsložek ve složce.

Číst a spouštět – dává uživateli práva udělená oprávněním Číst a Zobrazovat obsah složky a umožňuje procházet složkami.

Měnit – umožňuje odstranit složku a provádět akce povolené oprávněním Zapisovat a Číst a spouštět.

Úplné řízení – umožňuje měnit oprávnění, přebírat vlastnictví, odstraňovat podsložky a soubory a provádět akce, jež umožňují ostatní oprávnění.

Následující oprávnění se týkají souborů na oddílech se systémem souborů NTFS.

Číst – umožňuje číst soubory a zobrazit atributy souboru, jeho vlastníka a oprávnění.

Zapisovat – umožňuje přepisovat soubory, měnit atributy souboru, zobrazit vlastníka a jeho oprávnění.

Číst a spouštět – dává uživateli práva potřebná ke spuštění aplikací a práva udělená oprávněním Číst.

Měnit – umožňuje měnit a odstranit soubor a provádět akce povolené oprávněním Zapisovat a Číst a spouštět.

Úplné řízení – umožňuje měnit oprávnění, přebírat vlastnictví a provádět akce, jež umožňují ostatní oprávnění.

Zvláštní oprávnění k souborům a složkám

Po klepnutí na tlačítko Upřesnit se zobrazí dialog Vlastnosti, v němž jsou k dispozici podrobnější oprávnění. Pro uživatele, skupinu nebo objekt zabezpečení lze nastavit další oprávnění:

Procházet složkou/Spouštět soubory – umožňuje nebo odmítá možnost procházet procházet složkami a dosáhnout tak jiných souborů nebo složek i v případě, že uživatel nemá oprávnění k procházení složek. (Pokud skupina nebo uživatel nemají udělené oprávnění Nepoužívat kontrolu procházení – v modulu snap-in Zásady skupiny).

Oprávnění Spuštění souboru povoluje nebo odmítá možnost spouštět programové soubory.

Zobrazovat obsah složky/Číst data - umožňuje nebo odmítá možnost zobrazit názvy souborů ve složce. Oprávnění Číst data povoluje nebo odpírá možnost zobrazit data v souborech.

Číst atributy - umožňuje nebo odmítá možnost zobrazit atributy souboru nebo složky. Atributy definuje systém souborů NTFS.

Číst rozšířené atributy - umožňuje nebo odmítá možnost zobrazit rozšířené atributy souboru nebo složky. Rozšířené atributy definují programy a mohou se lišit podle jednotlivých programů.

Vytvářet soubory/Zapisovat data - umožňuje nebo odpírá možnost vytvářet soubory ve složce. Oprávnění Zapisovat data umožňuje nebo odpírá možnost provádět změny v souboru a přepisovat stávající obsah souboru.

Vytvářet složky/Připojovat data - umožňuje nebo odpírá možnost vytvářet složky ve složce. Oprávnění Zapisovat data umožňuje nebo odpírá možnost provádět změny na konci souboru, ale neumožňuje mu měnit, odstraňovat ani přepisovat stávající data.

Zapisovat atributy - umožňuje nebo odpírá možnost změnit atributy souboru nebo složky.

Zapisovat rozšířené atributy - umožňuje nebo odpírá možnost změnit rozšířené atributy souboru nebo složky.

Odstranit - umožňuje nebo odpírá možnost odstraňovat soubor nebo složku. Uživatel, který nemá povoleno Odstranit soubor, může jej odstranit, pokud má v nadřazené složce povoleno Odstraňovat podsložky a soubory

Odstraňovat podsložky a soubory - umožňuje nebo odpírá možnost odstraňovat soubory nebo podsložky i v případě, že uživatel nemá uděleno oprávnění Odstranit k podsložce nebo souboru.

Číst oprávnění - umožňuje nebo odpírá možnost číst oprávnění souboru nebo složky.

Měnit oprávnění - umožňuje nebo odpírá možnost měnit oprávnění souboru nebo složky.

Přebírat vlastnictví - umožňuje nebo odpírá možnost přebírat vlastnictví souboru nebo složky. Vlastník souboru nebo složky může vždy měnit jejich oprávnění bez ohledu na existující oprávnění chránící soubor nebo složku.

Řada zvláštních oprávnění se konfiguruje již při výběru určitého základního oprávnění. Proto obecně nebývá nutné ručně konfigurovat zvláštní oprávnění.

Whoami – nástroj k zobrazování uživatelských práv a oprávnění jedince

Tento nástroj spuštěný z příkazového řádku vrací název domény nebo počítače a jméno uživatele, který je právě přihlášen k počítači; dále vrací úplný obsah přístupového tokenu aktuálního uživatele. Zobrazí jméno uživatele a jeho identifikátory zabezpečení (SID), skupiny a jejich identifikátory SID, oprávnění a jejich stav (povolené nebo zakázané), a ID přihlášení. Instalace se provádí z adresáře Support/Tools na CD-ROM s operačním systémem Windows XP Professional.

K nastavení výsledků, jež vrátí nástroj whoami se používají přepínače:

/ALL zobrazí všechny informace v aktuálním přístupovém tokenu

/USER zobrazí uživatele identifikovaného v aktuálním přístupovém tokenu

/GROUPS zobrazí skupiny uvedené v aktuálním přístupovém tokenu

/PRIV zobrazí oprávnění spojená s aktuálním přístupovým tokenem

/LOGONID zobrazí ID přihlášení použité pro aktuální relaci

/SID zobrazí identifikátory SID spojené s aktuální relací (používá se v kombinaci s /USER, /GROUPS, /PRIV, /LOGONID)

/NOVERBOSE zobrazí minimální informace (používá se v kombinaci s /USER, /GROUPS, /PRIV, /LOGONID)

Systém NTFS ve verzi Windows 2000 (dříve NT 5.0) nabízí další zlepšení v oblasti přístupu k datům. Má vestavěnou podporu pro šifrování, která zabrání programům jako NTFSDOS obcházet bezpečnostní nastavení a zobrazit data. NTFS nešifruje přímo, ale je v něm přidán driver, který to zajistí. Tato část komunikuje se systémem zabezpečení NT, aby bylo zajištěno, že se k zašifrovaným datům dostane pouze oprávněný uživatel. Každý soubor složky může být zašifrován a pak není možné, aby ho někdo přečetl jinou instalací NT.

3.2 Bezpečnostní modely a politika

Řízení přístupu může být rozděleno do tří úrovní: modely, politika a mechanismus. Bezpečnostní model definuje systémové subjekty, objekty a práva. Bezpečnostní politika definuje, která práva k objektům jsou subjektům přidělena. Bezpečnostní mechanismus je metoda, kterou systém vykonává bezpečnostní politiku. V mnoho systémech se politika v průběhu času mění podle nastavení systémových zdrojů a změn uživatele, ale bezpečnostní model a mechanismus nevyžadují změny. Proto je bezpečnostní politika oddělena od mechanismu a modelu.

Oblíbený bezpečnostní model organizuje uživatele do tříd. Nevýhoda je, že přístupová práva jsou uložena v každém souboru a specifikují jednotlivé vlastníky a skupiny, takže nanejvýš jedna skupina může přistoupit k určitému souboru. Kromě toho systém může potřebovat změnit svolení skupiny pro několik souborů, když přidělí nová přístupová práva skupině uživatelů.

Role-based access control (RBAC) model

V politice založené na RBAC modelu jsou oprávnění sdružené s rolími a uživatelé jsou rozděleni jako členové určitých rolí.

Základní koncept rolí: přiřadit oprávnění na základě funkčních rolí v dané organizaci a potom zařadit uživatelé do příslušných rolí. V RBAC politice se rozhodování o přístupu ke zdrojům vykonává na základě rolí, ve kterých vystupují jednotliví uživatelé. Role se zakládají pro různé pozice v organizaci a uživatelé se zařazují do rolí na základě jejich pracovního rozdělení – zodpovědnosti a kvalifikaci. Protože role v jedné organizaci jsou relativně stabilní a častěji se mění příslušnost uživatelů k rolím, RBAC výrazně snižuje komplexnost, náklady a výskyt omylů administrátora při správě přístupů ke zdrojům. RBAC umožňuje centralizovat správu přístupu, protože příslušnost uživatelů k rolím i příslušnost oprávnění k rolím je možné ovládat centralizovaně.

Jeden uživatel může být členem více rolí a v každé roli může vystupovat více uživatelů. Jedna role může mít přiřazeno více oprávnění, každé oprávnění může být přiřazeno více rolím.

Oprávnění je přístupové právo ke jednomu nebo více zdrojům v systému. Význam oprávnění závisí na konkrétním systému implementujícím mechanismy pro RBAC.

Hierarchie rolí je přirozený způsob strukturování rolí, který respektuje strukturu organizace. Role nadřazená jiné roli dědí od ní všechny její pravomoci. Uživatel – člen nadřazené role - může aktivovat jakoukoli podmnožinu rolí, od nichž tato role dědí.

Discretionary access control (DAC)

Většina systémů založených na bázi UNIXu následuje model DAC, kde jednotlivým uživatelům či jejich skupinám přiděluje oprávněný uživatel potřebná přístupová práva (privilegia) k jednotlivým objektům. Nevýhodou DAC je to, že každý subjekt, který získá přístup k nějakému objektu, může tento objekt zpřístupnit komukoli jinému (např. udělat veřejnou kopii tajného souboru).

Mandatory Access Control (MAC)

Objekty jsou členěny do předem daných bezpečnostních úrovní a jednotliví uživatelé jsou zařazeni do skupin s příslušnými přístupovými právy. Jedná se o využití principu víceúrovňové bezpečnosti (multilevel security) v databázovém prostředí. S poskytnutím přístupu je zároveň vynucováno dodržování stanovených omezení (např. jakákoli kopie tajného souboru bude zase tajná). [5]

Bell-LaPadula model

Formální model bezpečnostní politiky, který popisuje nastavení práv řízení přístupu.

V tomto formálním modelu, jsou entity v počítačovém systému rozděleny do abstraktních souborů předmětů a objektů. Každá změna stavu zachovává bezpečnost tak, že se přesune z bezpečného stavu do bezpečného stavu, proto systém zůstává bezpečný.

Stav systému je definován jako “bezpečný” pouze pokud dovolené přístupové režimy subjektů k objektům odpovídají specifické bezpečnostní politice. Aby se stanovilo, zda je dovolen přístupový režim nebo ne, je přístupové právo subjektu přirovnáno k nastavení objektu a rozhoduje se podle toho zda je subjekt oprávněn pro přístup. [7]

4 NAPADENÍ BEZPEČNOSTI

4.1 Kryptoanalýza

Kryptoanalytické útoky se pokouší rozšifrovat šifrovaný text bez vlastnictví dešifrovacího klíče. Nejběžnější formou kryptoanalytického útoku jsou ty, v nichž je šifrovací algoritmus analyzován, aby byl nalezen vztah mezi bity v šifrovacím klíči a šifrovaným textem. Cílem takového útoku je zjistit ze šifrovaného textu klíč.

Slabý statistický trend mezi šifrovaným textem a klíčem může být využit k získání znalostí o klíči. Vhodná správa klíče a určení doby zániku klíče může omezit náchylnost k kryptoanalytickému útoku. Čím delší šifrovací klíč použijeme, tím víc zašifrovaného textu může útočník využít k odvození klíče.

Metody kryptoanalýzy:

V kryptoanalýze vždy předpokládáme, že kryptoanalytik zná celý šifrovací algoritmus. Existují čtyři základní typy kryptoanalytických útoků:

- **využití pouze šifrovaného textu** (ciphertext-only attack). Jedinou informací, kterou kryptoanalytik disponuje, je znalost šifrovaného textu různých zpráv zašifrovaných tímž algoritmem (a tímž klíčem).
- **využití dvojic otevřený text a šifrový text** (known-plaintext attack). Kryptoanalytik zde zná nejen příslušné šifrové texty, ale i jim odpovídající otevřené texty.
- **využití volitelných otevřených textů** (chosen-plaintext attack). Oproti předešlému typu útoku má kryptoanalytik navíc možnost vybírat si jaké otevřené texty budou zašifrovány (vůči použití tohoto typu útoku je například zranitelný algoritmus RSA).
- **využití volitelných šifrových textů** (chosen-ciphertext attack). Kryptoanalytik si může vybírat šifrové texty, které budou posléze dešifrovány a má přístup k takto dešifrovaným otevřeným textům.

4.2 Viry

Virus je spustitelný kód – často posílán jako příloha k e-mailové zprávě nebo skrytý v souborech jako audio klipy, video klipy a hry – který se připojuje k jiným souborům nebo je přepisuje, aby se mohl množit. Viry mohou narušit soubory, ovládnout aplikace nebo i smazat harddisk.

Viry lze klasifikovat z různých hledisek. Jednou z možností jak viry klasifikovat je tato:

- **Boot viry** – infikuje bootovací sektor harddisku počítače. Při restartu počítače, který má povoleno zavádění systému z disketové mechaniky a v mechanice je disketa s boot virem, vir se spustí a napadne systémové oblasti pevného disku. Při dalším spuštění počítače se boot vir inicializuje z pevného disku a napadá diskety, které uživatel použije. Virus má samozřejmě velký zájem o to, aby nebyl odhalen například antivirovým programem. Některé typy boot virů si proto původní obsah boot sektoru pevného disku zálohují a na vyžádání jej předkládají k nahlédnutí. Ochrana je jednoduchá: nezapomínat diskety v mechanice při startu nebo restartu počítače (nebo si stáhnout program, který vyjmutí diskety připomene).
- **Souborové viry** - hostitelem těchto virů jsou spustitelné soubory, nejčastěji spustitelné binární (COM, EXE,...), dávkové soubory BAT a soubory ovladačů SYS. Cílem viru je rozmnožení škodlivého kódu ihned po spuštění hostitelského souboru. Podle metody infekce je lze rozdělit do následujících skupin:
 - ♦ **přepisující (overwriting) viry** - přepíše původní soubor tělem viru. Původní data jsou tak nenávratně ztracena a spuštěním souboru dojde pouze k aktivaci samotného viru snažícího se o replikaci. Nejsou paměťově rezidentní a označujeme je za viry přímé akce.
 - ♦ **parazitické (parasitic) viry** - připojují se k proveditelnému hostiteli, aniž by ho trvale poškodily. Původní soubor je vždy upraven tak, aby nejdříve došlo k aktivaci viru a následně pak původního programu. Nejčastěji se připojují na konec původního souboru (append metota).
- **Doprovodné (companion) viry** - tyto viry vytváří nový soubor se stejným názvem jako některý spustitelný soubor, avšak např. s příponou COM,

případně zamění přípony EXE za COM, tak aby došlo při volání souboru k jejich spuštění. Po jejich spuštění někdy uvolní prostor i původně žádanému programu.

- **Makroviry** - vyvinuly se z masivního rozšíření kancelářských aplikací MS Office a jejich možnosti vytvářet makra. Makro, případně soubor maker, které je schopno samo sebe zkopírovat z jednoho dokumentu do druhého je označováno jako makrovirus. Díky rozšíření MS Office na různé platformy (PC, Macintosh) lze makroviry považovat za multi-platformní.

Princip činnosti makroviru: Pokud je dokument, obsahující makrovirus, danou aplikací otevřen, může být za jistých podmínek spuštěn (např. pomocí automakra, předdefinovaných kláves, položky menu, stisk tlačítka na stavové liště). Virus pak může sebe sama zkopírovat do nějakého globálního dokumentu (např. šablona NORMAL.DOT) a je tak spuštěn vždy při použití dané aplikace a může napadat všechny dokumenty této aplikace.

Druhů virů je samozřejmě více, zde jsou uvedeny pouze ty nejrozšířenější a nejznámější.

4.3 Červi

Červ je spustitelný kód, který se šíří díky infikovaným souborům sítí. Červi zřídka vyžadují akci uživatele, aby se mohli množit, a nepotřebují být připojeny k jiným souborům, aby se mohli šířit. Šíří se v podobě síťových paketů. Úspěšně infikovaný systém pak směřuje tyto pakety na další systémy v síti Internet. Podstata šíření červů je postavena na zneužívání konkrétních bezpečnostních děr operačního systému či softwaru, úspěšnost pak od rozšíření konkrétního programového vybavení. Červy nelze detekovat klasickou formou antivirového softwaru. Vedlejším efektem činnosti červů může být kompletní zahlcení sítě.

4.4 Trojské koně

Na rozdíl od virů nedochází k sebe-replikaci a infekci souborů. Trojský kůň je nejčastěji spustitelným EXE souborem, který neobsahuje nic jiného (užitečného), než samotné "tělo" trojského koně. Společně se skutečností, že není připojen k žádnému hostiteli z toho plyne, že jedinou formou dezinfekce je odmazání dotyčného souboru.

Někdy se trojský kůň skrývá v důvěryhodných programech nebo simulacích legitimních programů. Ve skutečnosti způsobuje zničení počítače nebo sítě, zatímco program běží. Takového trojského koně může být těžké detekovat, protože se jeví jako legitimní aplikace.

V současnosti se setkáváme s trojany v těchto formách:

- **Password-stealing trojani (PSW)** - obvykle sledují jednotlivé stisky kláves, ukládají je a následně odesílají na dané e-mailové adresy (zpravidla autorů viru). Majitelé těchto adres tak mohou získat i velice důležitá hesla.
- **Destruktivní trojani** - klasická forma. Pokud dojde ke spuštění, jsou např. likvidovány soubory na disku, disk je zformátován atd. Tuto formu má většina tzv. BAT trojanů.
- **Dropper** - "vypouštěč". Nese ve svém těle jiný škodlivý kód (např. virus), který je při aktivaci trojanu vypuštěn do systému.
- **Backdoor** - speciální forma trojského koně. Jedná se o aplikace klient-server sloužící pro vzdálenou správu PC. Klientská část, umístěná na straně útočníka, vysílá požadavky serverové části. Ta tyto požadavky plní, případně výsledky odesílá zpět ke klientu. [8]

Skupina trojských koní je méně běžná než viry. Velkým prostředníkem v šíření trojských koní jsou dnes samotné viry, které je během svého šíření vypouštějí do systému.

4.5 Útoky DoS (Denial-of-Service)

DoS útoky zabraňují systému, aby vyřizoval legitimní žádosti. Při mnoha DoS útocích neautorizovaný provoz nasytí zdroje sítě a omezuje přístup pro oprávněné uživatele. Například zaplaví servery packety s daty. Denial-of-Service útoky mohou způsobit, že se počítače v síti zhroutí nebo odpojí, naruší službu na Webu nebo i zneprístupní kritické systémy jako telekomunikace nebo centra letové kontroly.

Jiný typ DoS útoku míří na směrovací tabule (routing tables). Směrovací tabule jsou používány routery k určení toho, kam poslat data. Tento typ útoku mění směrovací tabule tak, aby přesměroval síťové aktivity – například všechna data mohou být posílána na jednu

adresu v síti. Podobný útok, DNS útok, může modifikovat adresy na které je veškerý provoz posílán.

Dalším druhem útoku je distributed denial-of-service. Zahlcení packety přichází z více počítačů. Je těžké ho zastavit, protože není snadné rozlišit, který síťový požadavek je od legitimního uživatele a který je součástí útoku.

4.6 Psychologické útoky

4.6.1 Hoax

V počítačovém světě slovem HOAX nejčastěji označujeme poplašnou zprávu, která šíří klamavou poplašnou zprávu.

Text poplašné zprávy obsahuje většinou tyto body:

- **Popis nebezpečí (viru)** - smyšlené nebezpečí (vir) bývá stručně popsáno, v případě viru bývá uváděný i způsob šíření.
- **Ničivé účinky viru**
- **Důvěryhodné zdroje varují** - ve většině případů se pisatel poplašné zprávy snaží přesvědčit, že varování přišlo od důvěryhodných zdrojů ("IBM a FBI varují" nebo "Microsoft upozorňuje" atd...)
- **Výzva k dalšímu rozeslání** - tento bod HOAX vždy obsahuje

Nebezpečnost:

- **Zbytečné zatěžování linek a serverů**
- **Šíření poplašných zpráv**
- **Šířením můžete vyzradit důvěrné informace**

Jestliže se HOAX preposílá, dává se k dispozici obrovský seznam e-mailových adres náhodným příjemcům. Díky lavinovitému šíření zprávy nemůžete vědět, komu v dalších úrovních bude e-mail doručen. Seznam adres je rájem pro spamery, kteří pak mohou na získané adresy posílat nevyžádané e-maily.

Jestliže zpráva obsahuje výzvu k hromadnému rozeslání na další adresy, je to s největší pravděpodobností HOAX.

4.6.2 Spam

Spamming je rozesílání nevyžádaných emailů ve snaze vnutit je lidem, kteří by jinak o ně neměli vůbec zájem. Většinou se jedná o obchodně zaměřené nabídky většinou pochybných produktů, poloilegálních služeb nebo návodů na rychlé zbohatnutí. [10]

Tyto zprávy jsou škodlivé jednak tím, že zbytečně zahlcují síť a brzdí provoz, jednak tím, že zahlcují adresáty, kteří je musí mazat a ztrácí tím čas. Obrana proti SPAMu je dvojí. Za prvé prevence, za druhé filtrování.

Prevence spočívá v utajení emailové adresy před spammery. Pokud se adresa objeví někde na internetu, je vysoce pravděpodobné, že na ni bude zasílán spam. Existují roboty, které automaticky procházejí webové stránky, diskuzní skupiny a konference a hledají e-mailové adresy a poskytují je spammerům. Proti tomu se lze bránit dvěma způsoby: Neuvádět nikde svoji adresu v podobě čitelné pro roboty. Tzn. místo odkazu na email vložit na webové stránky např. obrázek, který obsahuje vaši adresu. Nebo adresy psát ve tvaru „ja zavinac moje@firma tecka cz“.

Druhý způsob, jak spammeři často získají emailové adresy, že prostě posílají maily na statisíce strojově generovaných adres na tomto serveru. Např. víme, že Volný je velký poskytovatel připojení. Takže si prostě vezmeme seznam typických českých jmen a budeme generovat adresy novak@volny.cz, jan.novak@volny.cz, petr.novak@volny.cz...

Existují různé programy, které umějí odfiltrovat spam z vaší pošty. U některých poskytovatelů mailů si lze tuto službu zaplatit, jindy se musí příslušný program instalovat. Filtry spamu fungují na principu statistického filtrování: prostě se analyzuje obsah mailu, to, kolikrát a v jaké souvislosti se v něm vyskytují jaká slova, a to se porovnává se známými vzorky spamu. Na základě toho se určí jisté skóre a pokud toto skóre překročí stanovenou hranici, je mail označen za spam. Je nutné si uvědomit, že je to statistická metoda - není stoprocentní. Filtr může (a bude) některý spam nechávat projít a jindy označí jako spam i zprávu, která ve skutečnosti spam není. [11]

4.7 Nevyžádaný software

Spyware je software, který shromažďuje osobní informace, aniž by upozornil, co dělá, a neumožní uživateli se rozhodnout, zda-li si tuto činnost přeje nebo nikoliv. Informace,

kteřé spyware shromažďuje, mohou být seznamy navštěvovaných webových stránek, nebo citlivější informace, jako jsou uživatelská jména a hesla.

Spyware je často spojen se softwarem, který zobrazuje reklamy, kterému se říká adware. Někteří zadavatelé reklamy mohou skrytě instalovat adware na počítač a vygenerovat tok nevyžádaných reklam, které mohou zaneść pracovní plochu a ovlivnit produktivitu uživatele. Nadbytečné zpracování dat pro sledování pohybu uživatele nebo zobrazování reklam může zbavit počítač výkonu a snížit tak jeho funkčnost.

Spyware a neoprávněný adware jsou dvěma příklady „nevyžádaného“ softwaru. Nevyžádaný software zahrnuje programy, které změní úvodní stránku nebo stránku vyhledávání, aniž by k tomu měly svolení uživatele. Existuje mnoho způsobů, jak se nevyžádaný software může do systému dostat. Běžným trikem je skrytá instalace softwaru během instalace jiného programu, například programu pro sdílení hudby nebo videa.

Uživatel se stane snadným cílem spywaru, pokud si stáhne hudbu pomocí programu pro sdílení souborů, hry zdarma z nedůvěryhodných stránek, nebo jiné softwarové programy pocházející z neznámého zdroje.

Kdykoliv se instaluje nějaký program do počítače, doporučuje se pečlivě si prostudovat všechny údaje a sdělení, včetně licenční smlouvy a prohlášení o nakládání se soukromými informacemi. Někdy je instalace adwaru popsána v instalační proceduře, ale může se objevit například až na samém konci licenční smlouvy nebo prohlášení o nakládání se soukromými informacemi.

Někdy se nevyžádaný software instaluje zcela skrytě, aniž by o tom byl uživatel jakkoliv vyrozuměn. Může k tomu dojít, pokud je v prohlížeči (Internet Explorer) nastavena nejnižší úroveň zabezpečení. Proto je bezpečnější tento parametr nastavit na střední nebo vyšší úroveň zabezpečení. Takto je možné kontrolovat programy instalované na počítač.

Tvůrci nevyžádaného softwaru často používají různé triky k nahrání jejich softwaru do počítače. Například opakované žádosti o stažení programu i přesto, že uživatel odpověděl „ne“. V tomto zabrání instalaci nevyžádaného softwaru uzavření webové stránky, která poprvé vyzvala ke stažení. Alternativně uzavření Internet Exploreru.

4.7.1 Příznaky nevyžádaného softwaru

Nevyžádaný software je naprogramován tak, aby pracoval bez vědomí uživatele, ale existuje několik způsobů, jak zjistit potenciální infikování systému.

- **Při spuštění internetového prohlížeče se otevírá neznámá stránka.** Některý nevyžádaný software mění tato nastavení bez vašeho vědomí.
- **Dojde k náhlému nárůstu reklam na stránkách, kde předtím nebyly.** Nevýžádaný software počítač někdy cíleně zahlcuje reklamními okny bez ohledu na to, jaké stránky jsou prohlíženy.
- **Zpomalení počítače.** Nevýžádaný software nemusí být naprogramován tak, aby využíval zdroje účinným způsobem. Zdroje využitě pro sledování činnosti na počítači a zasílání reklamy mohou zpomalit počítač a chyby v softwaru mohou zapříčinit havárii systému.

Ačkoliv je velmi důležité používat bránu firewall, antivirový program a aktualizovat software, žádné z těchto opatření neposkytne zaručenou ochranu před stažením nevyžádaného softwaru. K dispozici je však speciální detekční a opravný software, který pomůže při hledání a odstraňování nežádoucího softwaru z počítače. Nejznámějšími jsou Lavasoft Ad Aware a Spybot Search & Destroy (S&D). [12]

5 PREVENCE ÚTOKU A ŘEŠENÍ BEZPEČNOSTI

5.1 Antivirový software

Viry a červi se stali hrozbou pro podnikání i pro domácí uživatele a jejich činnost stála ekonomiku miliardy dolarů. Navíc počet útoků virů neustále roste. Antivirový systém se pokouší počítač od virů ochránit a/nebo identifikovat a odstranit viry, které již na počítači jsou. Existují různé způsoby, kterými může antivirový software viry najít a odstranit ze systému. Nicméně žádná z nich nemůže nabídnout kompletní ochranu.

5.1.1 Vyhledávání v databázi (Signature-scanning virus detection)

Vyhledávání v databázi závisí na znalosti struktury počítačového kódu viru. Většina antivirových programů udržuje seznam známých virů a jejich kódu. Všechny viry obsahují část zvanou podpis viru, která se při šíření viru nemění. Ve skutečnosti mnoho seznamu viru obsahuje také seznam virových podpisů. Antivirový program skenuje počítač a porovnává data s kódem viru.

Seznam virů se musí neustále aktualizovat. Nevýhodou této metody je také to, že virus musí být nejdříve identifikován poskytovatelem softwaru, aby mohl být seznam doplněn. Metoda je neúčinná proti polymorfním virům, které dokáží svůj kód modifikovat.

5.1.2 Heuristická analýza

Heuristická analýza je rozbor kódu hledající postupy pro činnost virů typické nebo nějak podezřelé. Tímto způsobem lze odhalit i dosud neznámé viry.

Pasivní heuristika prohledává soubory a hledala v nich typické příznaky (sekvence znaků) pro virus. Pokud bylo takových příznaků (často označovány jako flags) nalezeno dostatečné množství, byl takový soubor považován za napadený. Příznakem mohlo být např. volání nějaké služby INT 21h, zápis do souboru apod. Nevýhodou bylo, že pasivní heuristika nedokázala proniknout pod "povrch" kódovaných či polymorfních virů a tak složitější nedokázala detekovat. Neaktivní heuristiku mohly viry snadno oklamat tím, že určitou činnost vyvolaly jinou sekvencí příkazů než heuristika využívala.

V dnešní době je ve většině případů využita „aktivní“ heuristická analýza. Základem je emulátor kódu a s ním spojená existence virtuálního prostředí počítače. Emulátor kódu

dokáže spustit soubor a jeho část od-emulovat podobně, jako by ho spustil sám uživatel. Emulátor kódu ovšem veškerou činnost provádí ve virtuálním prostředí a skutečný počítač uživatele tak v případě „spuštění“ infikovaného souboru nemůže ohrozit. Pokud by byl zpracováván soubor infikován, emulátor v podstatě vykoná i činnost viru, od-emuluje dekodovací smyčku (dekryptor) a dostane se tak přímo do „vnitřnosti“ viru. V tomto stavu může již skener vyhledávat podle sekvencí. Pokud jsou během emulace sbírány informace o aktivitách programu (např. proces přesměrování vektorů přerušení...), může být do akce zapojena i aktivní heuristická analýza, která na základě získaných informací vyhodnotí, zda se ne/jedná o virus. Jelikož emulace programu probíhá pomaleji než při skutečném spuštění programu, má emulátor nastaven tzv. timeout - tj. čas (či počet instrukcí), po kterém se chod emulátoru na aktuálním souboru zastaví. Tohoto časového limitu některé viry dokážou využít pro svůj prospěch.

Nevýhodou heuristické analýzy je zvýšený výskyt planých poplachů, kdy program označí za virus i zcela nezávadné aplikace. Dnešní heuristická analýza však bývá často natolik propracovaná, že výskyt falešného poplachu je spíše náhodou.

5.1.3 Kontrola integrity

Kontrola integrity je sledování změn v systému, kdy si program po svém prvním spuštění vytvoří databázi souborů na disku – po dalším spuštění kontroluje aktuální stav souborů a využívá faktu, že uložením viru do některého souboru dojde ke změně souboru, kterou je možné detekovat – program však při nalezení změny nedokáže určit, zda-li ji způsobil vir, což může být poměrně velkou nevýhodou. O tom zda je změna způsobena uživatelem nebo systémem musí rozhodnout sám uživatel. Stejně jako v případě heuristické analýzy, i tato metoda dokáže odhalit doposud neanalyzované a neznámé viry. Předpokladem pro úspěšné nasazení je provedení instalace kontroly integrity v době, kdy počítač není infikován. Kontrola integrity by v opačném případě vycházela z uložených informací, které by byly již ve stavu infekce. Dalším negativem je, že kontrola integrity dokáže zachytit virus až ve chvíli, kdy je aktivní. Klasická forma kontroly integrity postupně mizí, místo ní lze najít u některých antivirů kombinaci s jiným způsobem kontroly.

5.1.4 Rezidentní sledování

Rezidentní sledování je proces antivirového programu, který průběžně kontroluje prováděné operace na PC a je vždy funkční na pozadí systému. Antivir tak dokáže důsledně upozornit uživatele na spuštění souboru, který je infikován virem a tento proces včas přerušit. Rezidentní štít některých antivirů sleduje také pokusy o zápis do bootovací tabulky PC, podezřelé operace přeruší či dá uživateli na výběr, zda-li zápis do boot sektoru povolí. Velmi důležité je nastavení rezidentního sledování. Příliš jemné nastavení způsobuje časté poplachy. Je-li naopak nastavení příliš hrubé, program nebude detekovat téměř nic (tedy ani některé viry).

5.2 Firewally

Firewall chrání LAN před útoky zvnějšku. Kontroluje všechnu příchozí a odchozí komunikaci. Může zakázat veškerý přenos dat, který není výslovně povolen, nebo naopak povolit přenos dat, který není výslovně zakázán. Každá LAN může být do internetu připojena přes gateway, která obvykle obsahuje firewall.

Komunikace v síti probíhá na základě přenosu síťových paketů. Každý paket nese IP datagram, který obsahuje příchozí i odchozí IP adresy, porty na které má paket přijít a další informace.

Dále pak už obsahuje samotný obsah, jež je tvořen právě tou informací, která má být přenesena.

Firewall pracuje ve dvou základních metodách, ale v každé z nich nějakým způsobem vyhodnocuje přenášené pakety. Buď vyhodnocuje IP datagram a to zda vyhovuje nastaveným podmínkám - pak se jedná o paketový filtr a jde o síťovou vrstvu a nebo se zabývá obsahem jednotlivých paketů - jde o tzv. aplikační bránu a tudíž se hovoří o aplikační vrstvě.

Avšak není výjimkou, že firewall pro svou spolehlivou funkci využívá obě dvě zmíněné možnosti zároveň.

5.2.1 Paketové filtry

Pracují tak, že vyhodnocují jednotlivé pakety podle hlavičky IP datagramu, jež obsahuje IP adresu odesílatele i příjemce, zdrojový i cílový port a další informace. Paketový filtr má

daná pravidla, podle kterých pak ze získaných informací z IP datagramu rozhoduje, zda paket projde a nebo bude zahozen. Pracují však pouze na úrovni uvedených vlastností paketů a nedokáží vyhodnotit jejich obsah. Nedokáží například rozpoznat autorizovaného uživatele. Samotné paketové filtry jsou velice rychlé a nenáročné na systémové zdroje. Hlavní výhodou je, že nepotřebuje žádnou podpůrnou aplikaci a celý proces se odehrává přímo v jádře. V dnešní době se paketové filtry vyskytují už ve všech nových operačních systémech jako jsou UNIX, BSD, Windows XP, Mac OS a i Linux.

5.2.2 Aplikační brány

Označují se také jako aplikační gateway, Proxy, gateway, aplikační Proxy. Je to vlastně software běžící na firewallu. Existují dva nepatrně odlišné způsoby fungování.

První z nich je podobný paketovému filtru jen s tím rozdílem, že rozhodování se odehrává s využitím informací jednotlivých aplikačních vrstev (kontroluje obsah paketu). Chová se vlastně navenek jako směrovač, neboť pracuje pouze v síťové vrstvě, avšak interně zasahuje až na aplikační vrstvu. Což znamená, že nevyhodnocuje jen odkud kam paket jde, ale sleduje i jeho obsah.

Častěji se však používá řešení druhé. Uživatel zadá do svého internetového prohlížeče (browseru) adresu, ale svůj požadavek nemíří přímo na server (jak by tomu bylo bez proxy), ale odesílá se skrze prostředníka jimž je právě proxy. Brána požadavek přijme, sama vygeneruje požadavek svým jménem a odešle jej příslušnému serveru. Když pak přijme zpět výsledek, tak jej pošle zájemci.

Hlavní výhodou firewallu používajícího proxy je ta, že již nemůže zůstat pro koncového uživatele transparentní. To znamená, že při kliknutí na odkaz se informace neodešle přímo na cílový server, ale na proxy bránu, která teprve zajistí vše, co je třeba. Jelikož tedy data procházejí skrze proxy gateway, je možné je zaznamenávat, monitorovat či různě blokovat.

Jednou z mnoha dalších funkcí, které může proxy brána plnit, je funkce cache. Tzn. proxy si může uchovávat již jednou navštívené WWW stránky, aby při dalším požadavku na jejich zobrazení je nebylo nutné stahovat z internetu a tím zatěžovat linku, ale stačilo je pouze poslat ze své vnitřní cache paměti. [13]

5.3 Bezpečnostní záplaty

Operační systémy a ostatní software často obsahuje bezpečnostní díry, které jsou objeveny až poté, co si je nainstaluje velké množství uživatelů. Ke včasnému poskytnutí bezpečnostní záplaty musí softwaroví vývojáři nejdříve najít neznámé bezpečnostní díry, rychle vytvořit záplaty (patches) a umožnit uživatelům si záplaty stáhnout.

Často není pouhé spuštění patche dostatečnou ochranou. Například červ Slammer worm zneužíval bezpečnostní díru, pro kterou už byl patch vytvořen.

Společnost Microsoft vydává na nalezené bezpečnostní problémy opravné balíčky.

Prakticky existují zhruba dva druhy oprav:

Záplata (hotfix) – opravy jednotlivých nalezených problémů. Microsoft každému zaregistrovanému bezpečnostnímu problému přiřadí evidenční číslo, a pokud to je možné, vytvoří k němu i záplatu – hotfix. Tu poté společně s Microsoft Security Bulletinem, který danou chybu dokumentuje, zveřejní na svých internetových stránkách.

Service Pack – Celkovým výsledkem snažení je Service Pack - komplexní kumulativní opravný balíček pro konkrétní jazyk a verzi operačního systému. Jedná se o sesazené záplaty v jednom instalačním balíčku, který je vydáván zhruba jednou až dvakrát do roka. Service Pack je kumulativní, což znamená, že v sobě obsahuje všechny předchozí opravy až do jeho vydání a můžeme tedy instalovat vždy jen jeho poslední verzi.

Informace o jednotlivých vydaných záplatách jsou k dispozici v Microsoft Security Bulletinech, které jsou zveřejněny na adrese <http://www.microsoft.com/security>.

Service Packy a opravné záplaty jsou samozřejmě dostupné z webu <http://www.windowsupdate.com/> společnosti Microsoft a často jsou i součástí doprovodných CD počítačových časopisů.

5.3.1 Windows Update

Windows Update je pojem, který je spjat s operačním systémem Windows již od jeho verze 98. Je určen pro zautomatizování přístupu k systému bezpečnostních záplat. Windows 95 touto technologií nedisponují – aktualizace se musí provádět pomocí jiného řešení – ručního vyhledání a instalace záplat.

Windows Update je systém webových stránek, který je provázán se systémem, ze kterého se na něj uživatel připojí. Jeho síla je v automatické detekci verzí programů a operačního systému, což ve výsledku uživateli zobrazí pouze aktualizace, které má smysl na konkrétní operační systém instalovat. Ze zobrazení jsou také vynechány již nainstalované aktualizace.

Pomocí Windows Update je možné stáhnout jednak bezpečnostní záplaty operačního systému, nové verze a opravy programů dodávaných s operačním systémem (Windows Messenger, Windows Media Player, Outlook Express, Internet Explorer, ...) a aktualizace ovladačů zařízení. [14]

5.3.2 Linux Update

Pro update programů v systému Linux se používají tzv. balíčky. Pro každou distribuci je typický jiný balíčkovací systém. Systém balíčků spočívá v tom, že instalovaný soubor je začleněn do databáze již instalovaných programů, data jsou zkopírována na "správné" místo a v databázi jsou následně uloženy informace o tom, že tento proces proběhl.

Zde vzniká problém závislostí balíčků. Určité balíčky mají vyznačenu závislost na jiném balíčku. Uživatel musí mnohdy ručně hledat požadované balíčky, a to i přesnou verzi. Obranou proti tomu by mělo být správné pořadí při instalaci, případně použití inteligentnějšího nástroje - nadstavby, která zvládne řešení problémů se závislostmi za uživatele.

Pro vyhledávání a stahování balíčků, na nichž je požadovaný balíček závislý, existují různé nástroje. Záleží na distribuci:

- Debian apt
- Gentoo emerge
- Mandrakelinux urpmi
- Red Hat, Fedora Core a jejich klony yum
- Slackware swaret
- SUSE Yast

Každý program používá konfigurační soubory, kde se musí zadat adresy repozitářů na Internetu, odkud se balíčky budou stahovat. [15]

Debian

Pro update distribucí odvozených od Debianu se používá program apt-get. Jedná se o řádkovou metodu instalace balíčků. V konfiguračním souboru `/etc/apt/sources.list` specifikujeme adresu vhodného mirroru a oddílů a provedeme

```
apt-get update
apt-get dist-upgrade
```

Utilita apt-get kontroluje vzájemné závislosti balíčků a v případě potřeby sama nainstaluje i všechny ostatní balíčky, na nichž daný balíček závisí (ovšem pouze závislé balíčky, nikoli již ty, co jsou pouze doporučeny či navrženy).

Jestliže je k dispozici balík novější než instalovaný apt-get jej vezme a upgraduje balík starý. Nikdy nejsou odstraňovány instalované balíky, nikdy není do systému přidán balík, který by v něm předtím nebyl. Nové verze starých, instalovaných balíčků, které nemohou být instalovány bez porušení instalačního statutu jiného balíku budou ponechány v jejich současné verzi. Proto bude nutné použít `dpkg` nebo `dselect` k odstranění nebo přeinstalování některých poškozených balíčků nebo závislostí.

Gentoo

Distribuce Gentoo se od jiných liší tím, že stahuje zdrojové kódy a kompilace probíhá až na samotném počítači. To zabere poměrně dlouhou dobu. Balíčkovací systém v Gentoo se jmenuje Portage a umožňuje jednoduchou, ale komplexní správu systému, upgrade i odinstalaci software. Závislosti mezi balíčky při instalaci si řeší systém automaticky, a tak je nemusí řešit a dohledávat uživatel (root).

Systém se udržuje v aktualizovaném stavu velmi snadno, kombinací příkazů:

```
#emerge sync // synchronizuje Portage systém
#emerge -u world // upgrade systému (všechny nainstalované komponenty)
```

Mandrake (Mandrivalinux)

Distribuce Mandrake (v roce 2005 přejmenovaný na Mandriva) vychází původně z distribuce Red Hat. S tím souvisí používání balíčků typu RPM. Ovšem kromě standardního nástroje pro práci s balíčky (*rpm*) nabízí i vlastní - **urpmi** (podobný nástroj jako *apt* u

Debianu). Balíčky RPM kromě vlastního programu obsahují informace o závislostech na jiných balíčcích, ať jde o celé programy nebo jen o knihovny.

```
urpmi.update -a stáhne nejnovější seznamy balíčků ze všech nevyjímatelych zdrojů
```

```
urpmi --auto --auto-select upgrade systému dle staženého seznamu balíčků
```

```
urpmi <jmeno_balicku> nainstaluje balíček daného jména. [18]
```

Instalovat, odstraňovat a aktualizovat balíčky je možné taky pomocí grafického nástroje RPMDrake. Aktualizace balíčků se provádí pomocí programu MandrakeUpdate, který je součástí RPMDrake.

Automatickou aktualizaci lze provádět také pomocí služby Mandrakeonline. Mandrakeonline poskytuje automatizovanou aktualizaci software včetně bezpečnostních záplat, oprav chyb a oznámení o chybách. Mandrakeonline informuje o dostupnosti aktualizací, přičemž si můžete naplánovat jejich instalaci na pro vás nejvhodnější čas. Je to však placená služba, lze ji objednat na adrese <http://www.mandriva.com/mandrakeonline>.

RPM balíčkovací systém využívají také jiné distribuce založené na RedHatu, tzn. RedHat, Fedora Core, a SuSe. RPM neřeší problém závislosti balíčků a uživatel si musí vyhledat balíčky sám. Příkazy pro instalaci a update jsou následující:

```
rpm -ivh mc-4.6.1-0.8-i386.rpm //Instalace
rpm -Uvh <balíček.rpm> //Update
```

Fedora Core 3

Fedora Core 3 obsahuje dva nástroje pro automatické udržování systému v aktuálním stavu, řešení závislostí balíčků, a doinstalaci dalších balíků, a to `yum` a `up2date`. Obsahuje i nástroj třetí, `apt-get`, ten si můžete doinstalovat později pomocí `yum install apt`, ale je to zcela duplicitní věc.

Vlastní aktualizace systému se provede pomocí:

```
yum check-update
yum update
```

6 STANDARD BEZPEČNOSTI - ORANGE BOOK

(TCSEC -Trusted Computer System Evaluation Criteria)

Standard americké vlády určující bezpečnost počítačových systémů. Má čtyři stupně, od A po D.

D je systém bez jakékoli bezpečnosti, C (dělí se na C1 a C2) vyžaduje přihlášení heslem a určitý auditovací mechanismus, B (B1-B3) vynucuje různé úrovně ověření věrohodnosti uživatele, A1 pak vyžaduje prokázaný matematický model pro ověření totožnosti a bezpečnosti.

třída D - žádná ochrana

třída C1 - volná ochrana

Oddělení uživatelů od dat, musí existovat metody umožňující uživatelům chránit vlastní data před ostatními, uživatel zvolí, zda tyto mechanismy bude používat

třída C2 - kontrolovaný přístup

Systém stále provádí volnou ochranu zdrojů, granularita však musí být až na úroveň jednotlivých uživatelů, musí být veden access log. Navíc ochrana proti residuům - obsahy paměti, registrů, ... poté, co proces přestane tyto používat. Residua nesmí být zpřístupněna někomu jinému.

třída B1- značkováná ochrana

Každý kontrolovaný subjekt a objekt musí mít přiřazen stupeň utajení a musí být tímto stupněm označen, každý přístup musí být ověřován dle Bell-LaPadula modelu, musí existovat popis implementovaného formálního modelu, systém je podrobován testování

třída B2 - strukturovaná ochrana

Musí být k dispozici verifikovatelný globální návrh systému, systém musí být rozdělen do dobře definovaných nezávislých modulů, návrh musí zohledňovat princip nejmenších možných oprávnění, bezpečnostní mechanismy musí být uplatňovány vůči všem subjektům a objektům včetně všech zařízení, musí existovat analýza možných skrytých kanálů

vlastní systém musí běžet v rámci své bezpečnostní domény a provádět kontroly své integrity

třída B3 - bezpečnostní domény

Systém musí být podrobitelný extenzivnímu testování, musí existovat úplný popis celkové struktury návrhu systému, musí být konceptuálně jednoduchý

musí existovat ochranné mechanismy na úrovni jednotlivých objektů, každý přístup musí být testován, kontrola na úrovni provádění jednotlivých typů přístupu daného subjektu

Systém musí být vysoce odolný vůči průnikům. Zařízení provádějící audit log musí umět odhadnout hrozící nebezpečí.

třída A1 - verifikovaný návrh

Návrh systému musí být formálně verifikován, existuje formální model bezpečnostního mechanismu s důkazem konzistentnosti, formální specifikace systému s ověřením, že odpovídá formálnímu modelu, ověření, že implementace není odchylná od formální specifikace, formální analýza skrytých kanálů.[21]

II. PRAKTICKÁ ČÁST

L0PHTCRACK

Nástroj L0phtcrack umožňuje získat hesla z registry, repair disku či pásky. Aby se uživatel dostal k registry nebo backup informacím v počítači, musí mít práva administrátora. Pouze administrátor může L0phtcrack nainstalovat.

L0phtcrack zjišťuje hesla klasickým způsobem. Hesla nelze odchytnout v použitelné podobě a jediné co lze získat jsou "hash" formy. L0phtcrack porovnává pouze hashe hesel získaných ze souboru hesel s hashi hádaných hesel. Nejdříve porovná hesla se slovníkem často používaných hesel. Pak vyzkouší metodu "Hybrid", která připojuje za běžná slova ze slovníku písmena a čísla. Nakonec je použito klasické "Brute force" metody, tzn. zkoušejí se všechny kombinace písmen a jiných nastavených znaků.

L0phtcrack využívá snahy Microsoftu o zpětnou kompatibilitu se staršími Windows 9x. Kromě dokonalejší metody hashování NTLM je v systému povoleno přijímání autentizace starších hashů LanMan algoritmu. Heslo je tedy rozděleno na části o sedmi znacích a každá část hesla se zkouší zvlášť. Proto heslo o 8 a více znacích trvá rozluštit stejně dlouho jako heslo o sedmi znacích.

Byl proveden test jak rychle dokáže program L0phtcrack4 uhodnout přihlašovací hesla uživatelů. V okamžiku testování bylo na počítači definováno 91 uživatelů. Každý uživatel měl zadáno heslo jiné délky a složitosti.

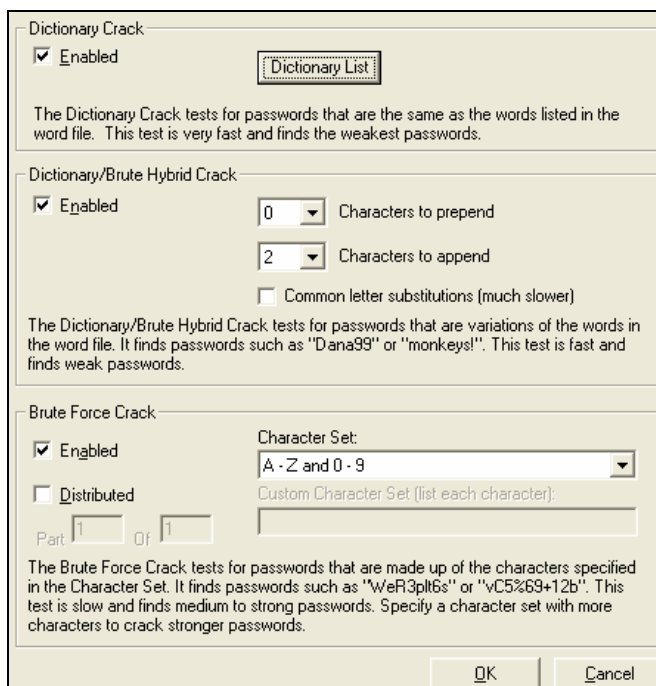
Do programu nebyl importován český slovník, je použit defaultní anglický slovník čítající 29156 slov.

Test byl proveden na operačním systému Windows XP Professional na této počítačové sestavě:

- CPU Name: AMD Athlon, 1245MHz
- Current FSB Clock: 99.6MHz (199.3MHz DDR)
- L1 Cache: Instruction: 64KBytes, Data: 64KBytes
- L2 Cache: Integrated: 512KBytes
- Memory type: DDR SDRAM
- Total memory size: 511MBytes
- Motherboard model: KV7(VIA KT600-8237)
- Motherboard chipset: VIA VT8368 KT400

Test 1

Pro první test bylo zvoleno hádání hesel pouze ze znakové sady A-Z, 0-9 a mezera. Odhadovaný čas pro metodu Brute force byl 7h15min. Po této době byl test ukončen. Hodnoty hesla byly voleny tak, aby zahrnuly většinu možností zadání alfanumerického hesla a bylo možné porovnat časy, v nich se podařilo heslo prolomit. Tabulky jsou vytvořeny pro jednotlivé délky hesel 1-7 znaků. Poslední tabulka znázorňuje časy pro delší hesla. Bylo tu bráno v úvahu také porovnávání časů pro podobná hesla různých délek.



Obr. 5. Nastavení programu pro test s alfanumerickými znaky

Výsledky jsou zobrazeny v tabulkách v příloze P I.

Všechny časy zobrazené v tabulkách jsou počítány od spuštění programu. Tedy pokud bylo heslo uhodnuto metodou Brute force, je do času započítána i doba, po kterou byla porovnávána slova ze slovníku a doba nutná pro metodu „Hybrid“

Program L0phtcrack vyzkouší hesla ze slovníku a metodu "Hybrid" během 55s.

Hesla z jednoho písmene jsou prolomena ihned. Tato hesla získá slovníkovou metodou.

Hesla o dvou až čtyřech znacích byla prolomena do 60s a hesla o pěti znacích do 90s.

Šestiznaková hesla odolávala přes 12min.

Některá sedmiznaková hesla byla uhádnuta až po 7h.

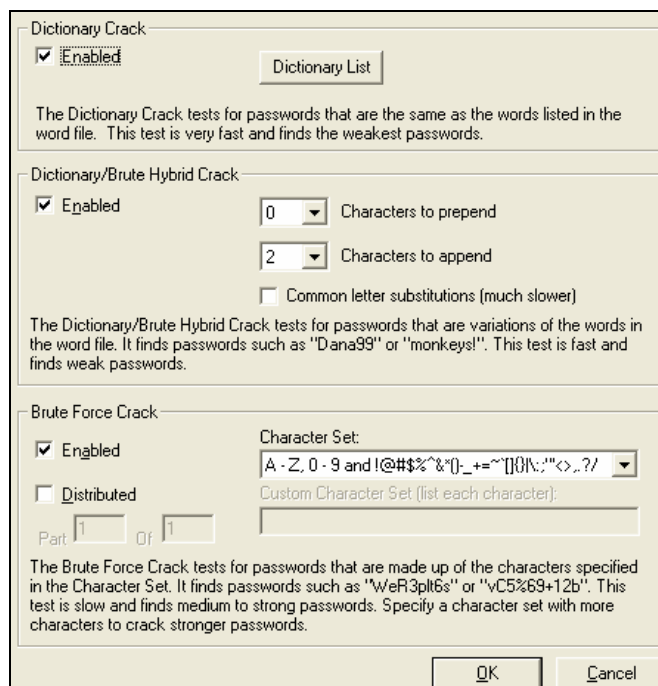
Hledání hesla s více jak sedmi znaky netrvalo déle, než hledání sedmiznakového hesla.

Pro srovnání je zde uvedeno několik podobných hesel. Například slovníkové heslo "fred" je modifikováno na "fred1", "fred02" a "fred123". Protože metoda "Hybrid" byla nastavena tak, aby hádala jen dva znaky za slovníkovým heslem, je heslo "fred123" prolomeno až metodou "Brute force" o 6h později. Metoda hybrid se však dá nastavit tak, aby za slovníkové heslo přidávala kombinace až tří znaků.

Střídání velikosti písmen na rychlost uhádnutí hesla nemá vliv, jak můžeme vidět na příkladu "praha", "prAHa" nebo "kobra5", "kOBrA5"

Test 2

Pro tento test bylo zvoleno hádání hesel ze znakové sady A-Z, 0-9, mezera a znaky !@#\$%^&*()-_+=~'[]{}|\\;""<>.,?/. Odhadovaný čas pro metodu Brute force byl 35dní, 19hodin. Tento odhad se však během testu zkracoval. Hesla byla zadávána v délce od jednoho znaku až do sedmi.



Obr. 6. Nastavení programu pro test se zvláštními znaky

Výsledky jsou zobrazeny v tabulkách v příloze P II.

Hesla s jedním, dvěma a třemi znaky byla rozluštěna ihned po použití metody Brute Force, tzn. do jedné minuty.

Čtyřznakové heslo odolávalo jen o několik sekund déle.

Heslo o pěti znacích bylo prolomeno do deseti minut.

Šestiznakové heslo bylo hledáno v řádu hodin, nejdéle však 8,5h.

Sedmiznakové heslo nebylo do konce testu rozluštěno ani jedno. Test trval 23h 16min, odhad času potřebného pro získání hesla byl na konci testu 32 dní a 20 minut.

Program L0phtcrack umožňuje uživateli nastavit jaké znaky mají být pro hledání hesel použity. Díky tomu může administrátor nastavit program tak, aby mohl najít i hesla s písmeny s diakritikou. Vyhledávání hesel s touto znakovou sadou však samozřejmě zabere mnohem více času.

ZÁVĚR

Předmětem teoretické části práce bylo shrnutí obecných poznatků o bezpečnosti operačních systémů. Operační systém může být ohrožen z mnoha směrů a je třeba věnovat pozornost každému z možných nebezpečí, protože systém je jen tak bezpečný, jak bezpečný je jeho nejslabší článek.

Jedním ze způsobů, jak ochránit tajná data je jejich šifrování. O tomto tématu pojednává kapitola 1. Šifrovat lze pomocí tajného nebo veřejného klíče.

Aby k datům neměl přístup kdokoli, je třeba ošetřit prokazování identity. To je možné také několika způsoby. Nejjednodušším a nejčastěji používanou metodou je ověření pomocí hesla. Pro větší bezpečnost systému se pak používá biometrická identifikace – tzn. identifikace pomocí specifických charakteristik člověka.

Operační systém může být ohrožen také neúmyslně, neopatrnými zásahy některých uživatelů. Proto je třeba omezit jejich přístupová práva, aby nedocházelo k narušení systémových zdrojů a služeb.

Bezpečnost systému ohrožují také útočníci zvnějšku. Nejčastěji se setkáváme zejména s viry, červy a trojskými koňmi. Šifrovaná data mohou být dešifrována kryptoanalatickými útoky. Řadí se sem samozřejmě i jiné útoky hackerů, ale tohle téma by vyžadovalo větší prostor, proto se mu v této práci nevěnuji.

V případě napadení nám mohou pomoci antivirové programy a firewally, které jsou dnes už na velmi vysoké úrovni. Operační systémy se také stále snaží opravovat bezpečnostní díry tím, že vydávají záplaty a opravné balíčky.

V praktické části práce jsem měla posoudit vliv délky a složitosti hesla na čas nutný k jeho prolomení známým programem L0phtcrack. Test probíhal pod operačním systémem Windows XP Professional. Výsledkem testů bylo potvrzení faktu, že hesla o délce 7 nebo 14 znaků jsou nejhůře zjistitelná a proto se nejlépe hodí pro zabezpečení účtu uživatelů. Naopak hesla s délkou do 5 znaků byla prolomena do 10 minut a to dokonce i když obsahovala zvláštní znaky.

SEZNAM POUŽITÉ LITERATURY

- [1] BARBEC, Tom. *Popis šifry Blowfish* [online]. [cit. 2005-04-16]. Dostupný z WWW: <http://moon.felk.cvut.cz/~pjv/Jak/_info/i677/html/blowfish.htm>
- [2] SOUČEK, Jakub. *Zabezpečení přístupu k datům* [online]. c2001. [cit. 2005-03-12]. Dostupný z WWW: <<http://kuba.spika.cz/diplomka/index9.htm>>
- [3] *Digitální podpis* [online]. c2000-2004. [cit. 2005-02-19]. Dostupný z WWW: <<http://www.mobilniaplikace.cz/buxus/web/cz/Products/Security/Security-Technology/Digital-Signature/>>
- [4] KOLÁŘ, Petr. *Operační systém Unix* [online]. [cit. 2005-03-12]. Dostupný z WWW: <<http://www.kai.vslib.cz/~kolar/unix/unix-s.pdf>>
- [5] TREBULA, Robert. *Vizuální návrh bezpečnostnej politiky* [online]. c2003. [cit. 2005-03-14] Dostupný z WWW: <<http://deepblue.sk/~r0b0/dp-trebula.pdf>>
- [6] KANKOVSKÝ, Pavel. *SELinux* [online] 5.5.2004 [cit. 2005-03-11] <Archív listu linux> Dostupný z WWW: <<http://www.linux.cz/lists/archive/linux/213173.html>>
- [7] *Bell-LaPadula model* [online]. [cit. 2005-04-01]. Dostupný z WWW: <http://wikipedia.infostar.cz/b/be/bell_lapadula_model.html>
- [8] HÁK, Igor. *Moderní počítačové viry* [online]. c2003. Poslední úprava 14.12.2004. [cit. 2005-02-20]. Dostupný z WWW: <<http://www.viry.cz/kniha/kniha.pdf>>
- [9] BERKA, Petr. *Počítačové viry a spam*. [online]. 2004/38-2.7.2004. [cit. 2005-02-20]. Dostupný z WWW: <<http://www.elektrorevue.cz/clanky/04038/#011>>
- [10] ŠKOPEK, Pavel. *Internetem se valí lavina a jmenuje se spam*. [online]. 26.08.2004 [cit. 2005-04-16]. Dostupný z WWW: <http://technet.idnes.cz/software.asp?r=software&c=A040823_5282260_software>
- [11] VRANÝ, Bolek. *Bezpečnost v digitálním světě - Spam a jak se mu bránit*. [online]. 30.06.2004 [cit. 2005-04-16]. Dostupný z WWW: <http://technet.idnes.cz/sw_internet.asp?r=sw_internet&c=A040629_5264238_sw_internet>
- [12] *Co byste měli o spyware vědět* [online]. 23.1.2004 [cit. 2005-03-13]. Dostupný z WWW: <<http://www.microsoft.com/cze/security/home/deviuoussoftware.msp>>
- [13] KUCHAR, Martin. *Co je to firewall a jak pracuje*. [online]. 9. prosince 2004 [cit.2005-02-26]. Dostupný z WWW: <<http://www.pctuning.cz/Default.aspx?CatID=75&ArtId=20541&LayID=3&Print=1>>

- [14] JURÁSEK, Vít. *Jak udržet Windows v bezpečí*. [online]. 4.1.2004 [cit. 2005-03-11]. Dostupný z WWW: <<http://www.zive.cz/h/Viryabezpecnost/AR.asp?ARI=114274>>
- [15] OTT, Vlastimil. *Na co se často ptáme: Balíčkovací systémy*. [online]. 16.2.2005 [cit. 2005-05-12]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/show/77005>>
- [16] HILLIARD, Bob. *Release Notes for Debian GNU/Linux 2.1 Slink) - Upgrade* [online]. version 2.1.11, 28 August, 1999. [cit. 2005-05-10]. Dostupný z WWW: <<http://www.debian.org/releases/slink/i386/release-notes/ch-upgrading.cs.html>>
- [17] DAVÍDEK, Tomáš. *apt-get* [online]. poslední změna 2005-05-04 [cit. 2005-05-10]. Dostupný z WWW: <<http://www-cjf.troja.mff.cuni.cz/~davidek/linux1/node13.html>>
- [18] FIALA, Martin. *Balíčkovací systém Mandrake Linuxu* [online]. 16.1.2004 [cit. 2005-05-14]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/show/52632>>
- [19] HUSTVEDT, Eskild. *Mandrakelinux Rpmdrake and URPMI HOWTO* [online]. c2004. [cit. 2005-05-14] Dostupný z WWW: <<http://www.mandrakehelp.com/RPM-HOWTO.html#1.3>>
- [20] HULÁN, Radek. *Fedora Core 3 pro lamy - prvotní instalace* [online]. 05.12.04 [cit. 2005-05-14]. Dostupný z WWW: <<http://hulan.cz/blog/?item=fedora-core-3-pro-lamy-prvotni-instalace>>
- [21] *Standardizace systémů spravujících senzitivní informace* [online]. [cit. 2005-04-01] Dostupný z WWW: <http://plusik.pohoda.cz/teaching/security/02_NORMY_MODEL.Y.DOC>
- [22] DOČEKAL, Daniel. *Je L0phtCrack tak nebezpečný jak mnozí tvrdí?* [online]. [cit. 2005-05-14]. Dostupný z WWW: <<http://www.pooh.cz/1001/a.asp?a=2003550&db=1001>>

Tento odkaz byl použit jako zdroj informací ke kapitole 2.1 Hesla. V současné době však není dostupný:

http://www.fi.muni.cz/~xmaly/bc/BP_xmaly.doc [cit. 2005-02-18]

- [23] SCAMBRAY, Joel, McCLURE, Stuart. *Hacking bez tajemství*. 2. vydání, Praha. Computer Press, 2002, ISBN 80-7226-644-6.

- [24] DEITEL, Harvey M. *Operating Systems*. Third edition. Pearson Prentice Hall. c2004. ISBN 0-13-124696-8. Part 7, Security, s.869-919
- [25] TANENBAUM Anrew S. *Modern operating systems*. Second edition. China Machine Press. English reprint edition c2002. ISBN 7-111-09156-6. Kapitola 9, Security, s. 589-670
- [26] HATCH, Brian, LEE, James. *Linux Hackerské útoky: Bezpečnost Linuxu – tajemství a řešení*. SoftPress. czech edition c2002. ISBN 80-86497-17-8
- [27] *Linux – dokumentační projekt*, Computer Press, 2003, ISBN 80-7226-761-2
- [28] ČADA, O. *Operační systémy*. Grada, 1993. ISBN 80-85623-44-7
- [29] KLIMEŠ, C. *Operační systémy I*. Ostravská univerzita v Ostravě, Katedra informatiky a počítačů. c2003. ISBN 80-7042-850-3
- [30] KOKOREVA, Olga.: *Registr Microsoft Windows XP*, Computer Press 2002, ISBN 80-7226-783-3
- [31] SOLOMON, David. A.: *Windows NT pro administrátory a vývojáře*, Computer Press 1999, ISBN 80-7226-147-9
- [32] *Microsoft Windows XP Professional Resource Kit*. Computer Press. 2002. ISBN 80-7226-608-X

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
DAC	Discretionary access control
DES	Data Encryption Standard
DoS	Denial-Of-Service
GID	Identifikace skupiny, do které vlastník patří (Unix)
KDC	Key Distribution Center
MAC	Mandatory Access Control
MD	Message Digest
PSW	Password-stealing trojani
RBAC	Role-based access control
RPM	RedHat Package Manager
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SID	Identifikátory zabezpečení (Unix)
TCSEC	Trusted Computer System Evaluation Criteria
UID	Identifikace vlastníka (Unix)

SEZNAM OBRÁZKŮ

<i>Obr. 1. Tajný klíč</i>	<i>8</i>
<i>Obr. 2. Dočasný klíč s využitím centrální autority KDC</i>	<i>9</i>
<i>Obr. 3. Veřejný klíč</i>	<i>10</i>
<i>Obr. 4. Princip digitálního podpisu</i>	<i>16</i>
<i>Obr. 5. Nastavení programu pro test s alfanumerickými znaky</i>	<i>47</i>
<i>Obr. 6. Nastavení programu pro test se zvláštními znaky</i>	<i>49</i>

SEZNAM TABULEK

<i>Tab. 1. Metoda password salting</i>	12
--	----

SEZNAM PŘÍLOH

- P I Alfanumerická hesla
- P II Hesla se zvláštními znaky
- P III CD-ROM

PŘÍLOHA P I: ALFANUMERICKÁ HESLA

Tabulky popisují, jak dlouhou dobu potřeboval program L0phtcrack k prolomení alfanumerického hesla v závislosti na délce hesla. Ve sloupci "Metoda " je typ útoku, který vedl k úspěšnému odhadu. Řazení hodnot v tabulkách je podle doby trvání.

Jednopísmenné heslo

Heslo	Čas trvání	Metoda
a	0d 0h 0m 0s	Dictionary
b	0d 0h 0m 0s	Dictionary
e	0d 0h 0m 0s	Dictionary
1	0d 0h 0m 0s	Dictionary
l	0d 0h 0m 1s	Dictionary
q	0d 0h 0m 1s	Dictionary
m	0d 0h 0m 1s	Dictionary
z	0d 0h 0m 1s	Dictionary
2	0d 0h 0m 55s	Brute Force
3	0d 0h 0m 55s	Brute Force
4	0d 0h 0m 55s	Brute Force
5	0d 0h 0m 55s	Brute Force
6	0d 0h 0m 55s	Brute Force
9	0d 0h 0m 55s	Brute Force

Dvoupísmenné heslo

Heslo	Čas trvání	Metoda
al	0d 0h 0m 0s	Dictionary
mt	0d 0h 0m 1s	Dictionary
s6	0d 0h 0m 43s	Hybrid
xx	0d 0h 0m 57s	Brute Force
qw	0d 0h 0m 57s	Brute Force

Třípísmenné heslo

Heslo	Čas trvání	Metoda
aaa	0d 0h 0m 0s	Dictionary
Tim	0d 0h 0m 1s	Dictionary
MI5	0d 0h 0m 32s	Hybrid
258	0d 0h 0m 57s	Brute Force
qwe	0d 0h 0m 57s	Brute Force
QwE	0d 0h 0m 57s	Brute Force
xzm	0d 0h 0m 57s	Brute Force

Heslo o čtyřech písmenech

Heslo	Čas trvání	Metoda
1234	0d 0h 0m 0s	Dictionary
fred	0d 0h 0m 0s	Dictionary
42na	0d 0h 0m 57s	Brute Force
Jana	0d 0h 0m 57s	Brute Force
n42a	0d 0h 0m 57s	Brute Force
nula	0d 0h 0m 57s	Brute Force
qwer	0d 0h 0m 57s	Brute Force
QwEr	0d 0h 0m 57s	Brute Force
1946	0d 0h 0m 58s	Brute Force
ahoj	0d 0h 0m 58s	Brute Force

Heslo o pěti písmenech

Heslo	Čas trvání	Metoda
qwert	0d 0h 0m 1s	Dictionary
QweRt	0d 0h 0m 1s	Dictionary
fred1	0d 0h 0m 21s	Hybrid
pet 5	0d 0h 0m 38s	Hybrid
heslo	0d 0h 0m 58s	Brute Force
Katka	0d 0h 0m 59s	Brute Force
praha	0d 0h 0m 59s	Brute Force
prAHa	0d 0h 0m 59s	Brute Force
k48sd	0d 0h 1m 0s	Brute Force
pet p	0d 0h 1m 2s	Brute Force
xxxxx	0d 0h 1m 6s	Brute Force
hokej	0d 0h 1m 8s	Brute Force
3dom1	0d 0h 1m 11s	Brute Force
14589	0d 0h 1m 13s	Brute Force

Heslo o šesti písmenech

Heslo	Čas trvání	Metoda
123456	0d 0h 0m 0s	Dictionary
aaaaaa	0d 0h 0m 0s	Dictionary
cookie	0d 0h 0m 0s	Dictionary
qwerty	0d 0h 0m 2s	Dictionary
fred02	0d 0h 0m 21s	Hybrid
mutace	0d 0h 1m 20s	Brute Force
narcis	0d 0h 3m 28s	Brute Force
fr63ed	0d 0h 3m 45s	Brute Force
cau xp	0d 0h 5m 30s	Brute Force
qwertz	0d 0h 9m 5s	Brute Force
148524	0d 0h 10m 15s	Brute Force
neco54	0d 0h 10m 18s	Brute Force
bizon7	0d 0h 11m 15s	Brute Force
kobra5	0d 0h 12m 13s	Brute Force
kOBrA5	0d 0h 12m 13s	Brute Force

Heslo o sedmi písmenech

Heslo	Čas trvání	Metoda
1234567	0d 0h 0m 0s	Dictionary
favorit	0d 0h 25m 36s	Brute Force
dededed	0d 1h 44m 56s	Brute Force
ddddddd	0d 1h 47m 23s	Brute Force
qwertzu	0d 3h 0m 37s	Brute Force
kobra41	0d 6h 14m 44s	Brute Force
fred123	0d 6h 3m 56s	Brute Force
1596369	0d 6h 50m 32s	Brute Force
vodo555	0d 7h 2m 31s	Brute Force

Heslo o více než sedmi písmenech

Písmen v hesle	Heslo	Čas trvání	Metoda
8	12345678	0d 0h 0m 0s	Dictionary
9	123456789	0d 0h 0m 1s	Hybrid
10	1234567890	0d 0h 0m 1s	Hybrid
12	asdfghjk12yx	0d 0h 1m 6s	Brute Force
14	dinosaurus jde	0d 0h 15m 29s	Brute Force
13	trinactpismen	0d 0h 28m 6s	Brute Force
10	Pokahontas	0d 0h 38m 6s	Brute Force
8	trpaslik	0d 1h 4m 33s	Brute Force
9	devetpism	0d 1h 5m 28s	Brute Force
10	desetpismn	0d 1h 5m 28s	Brute Force
13	pismena13cisl	0d 1h 26m 27s	Brute Force
14	ctrnact159abcd	0d 1h 55m 49s	Brute Force
12	kacer Donald	0d 2h 0m 36s	Brute Force
11	jedenactpis	0d 2h 30m 12s	Brute Force
11	pism11cisl	0d 2h 38m 21s	Brute Force
12	qwertzuipas	0d 3h 10m 30s	Brute Force
14	krtek ve meste	0d 4h 7m 35s	Brute Force
10	1472583690	0d 6h 24m 22s	Brute Force
9	potopa152	0d 6h 26m 26s	Brute Force
10	potopa1458	0d 6h 26m 26s	Brute Force
10	0123456789	0d 7h 2m 20s	Brute Force

PŘÍLOHA P II: HESLA SE ZVLÁŠTNÍMI ZNAKY

Tabulky popisují, jak dlouhou dobu potřeboval program L0phtcrack k prolomení hesla se zvláštními znaky v závislosti na délce hesla. Ve sloupci "Metoda" je typ útoku, který vedl k úspěšnému odhadu. Řazení hodnot v tabulkách je podle doby trvání.

Jednopísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
#	0d 0h 0m 56s	Brute Force
&	0d 0h 0m 56s	Brute Force
:	0d 0h 0m 56s	Brute Force
?	0d 0h 0m 56s	Brute Force
@	0d 0h 0m 56s	Brute Force

Dvoupísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
1#	0d 0h 0m 1s	Hybrid
a\$	0d 0h 0m 1s	Hybrid
@+	0d 0h 0m 56s	Brute Force
@8	0d 0h 0m 56s	Brute Force
@z	0d 0h 0m 56s	Brute Force
+6	0d 0h 0m 56s	Brute Force

Třípísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
13?	0d 0h 0m 1s	Hybrid
n?9	0d 0h 0m 35s	Hybrid
p#%	0d 0h 0m 37s	Hybrid
s5#	0d 0h 0m 44s	Hybrid
st*	0d 0h 0m 48s	Hybrid
%&#	0d 0h 0m 56s	Brute Force

Čtyřpísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
pro#	0d 0h 0m 40s	Hybrid
\$+1m	0d 0h 0m 58s	Brute Force
J@n@	0d 0h 1m 2s	Brute Force
z 3%	0d 0h 1m 2s	Brute Force
/*-+	0d 0h 1m 3s	Brute Force
+-* /	0d 0h 1m 5s	Brute Force

Pětipísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
fred*	0d 0h 0m 21s	Hybrid
&jine	0d 0h 1m 5s	Brute Force
hi#me	0d 0h 1m 7s	Brute Force
qw*er	0d 0h 1m 33s	Brute Force
as4@3	0d 0h 5m 38s	Brute Force
%%&%%	0d 0h 7m 49s	Brute Force
p5 %+	0d 0h 9m 0s	Brute Force

Šestipísmenné heslo se zvláštními znaky

Heslo	Čas trvání	Metoda
qwert%	0d 0h 0m 42s	Hybrid
&123ab	0d 3h 37m 13s	Brute Force
@@@@@@	0d 7h 11m 11s	Brute Force
fred	0d 7h 53m 54s	Brute Force
1234	0d 7h 56m 42s	Brute Force
Tim---	0d 8h 28m 15s	Brute Force

PŘÍLOHA P III: CD-ROM

Součástí bakalářské práce bylo vypracování zjednodušené HTML verze vhodné pro publikování na internetu, vytvoření prezentace pro přednášky předmětu Operační systémy a návrh souboru zkouškových otázek. Soubory jsou dostupné na přiloženém CD-ROM.