VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE PHISHINGU VE WEBOVÝCH STRÁNKÁCH

THESIS TITLE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. MAREK BEŇO**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MARTIN HOLKOVIČ**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Beňo Marek, Bc.**
Program: Informační technologie Obor: Bezpečnost informačních technologií
Název: **Detekce phishingu ve webových stránkách**
Phishing Detection in Web Pages
Kategorie: Bezpečnost

Zadání:

1. Nastudujte problematiku analýzy a detekce phishingu.
2. Analyzujte stávající nástroje pro detekci phishingu. Seznamte se s nástrojem Yara, jeho rozšířením vyvíjeným ve společnosti Avast a analyzujte jeho možnosti pro detekci phishingu.
3. Na základě analýzy stávajících nástrojů navrhněte nástroj pro automatickou analýzu a klasifikaci phishing vzorků sestávajících z webových stránek.
4. Implementujte navržený nástroj.
5. Navržený nástroj otestujte a porovnejte s ostatními nástroji.
6. Dosažené výsledky analyzujte a vyhodnoťte.

Literatura:

- Mahmoud Khonji, Youssef Iraqi: Phishing Detection: A Literature Survey, IEEE COMMUNICATIONS SURVEYS & TUTORIALS (2013)
- Krutika Rani Sahu: A Survey on Phishing Attacks, International Journal of Computer Application (2014)
- Zinal Shukla, Kirtirajsinh Zala, Riddhi Kotak: A Survey of Website Phishing Detection Techniques, International Journal on Future Revolution in Computer Science & Communication Engineering (2018)

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Holkovič Martin, Ing.**
Konzultant: Milkovič Marek, Ing., Avast
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 25. října 2018

Abstrakt

Táto práca sa zaoberá návrhom nástroja na detekciu a klasifikáciu phishing útokov. Práca popisuje techniky a formy phishing útokov. Na základe analýzy existujúcich nástrojov je navrhnuté vlastné riešenie pre klasifikáciu súborov. Spracovanie vstupných dát a vytvorenie modelu zabezpečuje implementovaný nástroj. Model vstupu je založený na hybridnej analýze vstupného súboru a URL. Spracovanie modelu a definíciu pravidiel umožňuje rozširujúci modul pre nástroj YARA. Výsledné riešenie umožňuje definíciu YARA pravidiel pre klasifikáciu phishing na základe štrukturálnych vlastností phishing súboru a charakteristík zdrojovej URL.

Abstract

This work deals with the design of a phishing attack detection and classification tool. The work describes techniques and forms of phishing attacks. Based on the analysis of existing tools a solution for file classification is proposed. Input parsing and creation of input model is handled by implemented tool. Model is based on hybrid analysis of input file and URL. Analysis of input model and definition of classification rules is enabled by YARA module. Implemented solution makes it possible to define YARA rules for phishing classification based on the structural properties of a phishing file and features of source URL.

Kľúčové slová

phishing, malvér, bezpečnosť, webové stránky, YARA, Python

Keywords

phishing, malware, security, web pages, YARA, Python

Citácia

BEŇO, Marek. *Detekce phishingu ve webových stránkách.* Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Holkovič

Detekce phishingu ve webových stránkách

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Ing. Martina Holkoviča. Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....
Marek Beňo
21. mája 2019

Poďakovanie

Za vedenie, odbornú pomoc a veľkú dávku ochoty a trpezlivosti ďakujem svojmu vedúcemu Ing. Martinovi Holkovičovi. Za odborné rady a pomoc pri vypracovaní tejto práce ďakujem Robertovi Kappovi DiS., Ing. Marekovi Milkovičovi a Ing. Jakubovi Křoustkovi, Ph.D..

Obsah

Kapitola 1

Úvod

Už odjakživa existovali ľudia, ktorý sa chceli obohatiť na úkor iných. Jeden zo spôsobov ako sa takto obohatiť je ziskom cenných dát a prostriedkov prostredníctvom rôznych podvodov. Spoločne so zlepšovaním a rozširovaním komunikačných technológií a internetu narastá aj počet podvodov spoločne nazývaných pojmom phishing.

Phishing útok je jeden z najrozšírenejších útokov najmä kvôli svojej jednoduchosti, ľahkému šíreniu a množstvom rôznych foriem útoku. Jednoduché šírenie znamená možnosť stať sa obeťou pre ľudí rôznych technických znalostí a pozícií naprieč nielen korporátnym prostredím ale aj osobným. Riziko, ktoré vyplýva z odcudzenia prístupov do finančných služieb, online účtov alebo interných dát sa tak dotýka širokého spektra ľudí.

Cieľom nástroja je skombinovať techniky dostupných nástrojov so zameraním na schopnosť definície vzorov pre známe ale i vopred neznáme spôsoby phishing útokov. Technika definície vzorov umožňuje definície generických vzorov pre detekciu nových vzoriek, ale i špecifických vzorov pre sledovanie a klasifikáciu známych útokov.

Pre detekciu Phishing útokov je potrebné pokryť rôzne formy súborov na ktorých je útok založený a rôzne spôsoby jeho šírenia. V kapitole 2 sa zaoberám definíciou phishing, rôznych foriem útoku, spôsobov šírenia, samotným procesom prípravy a priebehom útoku. Kapitola 3 popisuje existujúce spôsoby detekcie, reprezentácie phishing útoku a dostupné nástroje so zameraním na nástroj `YARA`. V kapitole 4 je navrhnutý nástroj zameraný na detekciu a klasifikáciu phishing útokov založených na webových stránkach a využitiu v ostatných formách phishing útoku. Kapitola 5 popisuje implementáciu navrhovaného `YARA` modulu a jednotlivých komponent pre nástroja. Kapitola 6 obsahuje popis dátových sád a experimentov s týmito dátovými sadami za účelom otestovania implementovaného nástroja.

Kapitola 2

Phishing

Phishing je dnes vďaka jeho relatívnej jednoduchosti a nízkej obstarávacej cene [21] jeden z najrozšírenejších podvodov vôbec. Podľa odborníkov [10] tento útok v budúcnosti naberie väčšiu dôležitosť hlavne v oblasti sociálnych sietí. Dôležitým faktorom tohoto útoku je využitie technológií a ich spojenie s technikami sociálneho inžinierstva. Útočník pri tomto útoku zneužíva najslabší článok bezpečnosti - ľudí. Počas útoku útočník predstiera identitu iného subjektu, získa dôveru a vyláka obeť na návnadu. Pripravená návnada má za úlohu vylákať od obeti cenné údaje alebo ju infikovať pre umožnenie ďalších útokov.

Podľa Anti Phishing Working Group (APWG) je Phishing: *Kriminálny mechanizmus, ktorý využíva formy sociálneho inžinierstva a technického podvodu k získaniu osobných údajov a prístupových údajov k finančným účtom* [3].

Zatiaľ čo táto definícia presne vystihuje podstatu Phishingu kde je použitý technický ale aj ľudský faktor je vymedzená len na zisk špecifických dát. Finančné dáta prevládajú cieľom phishing útokov, avšak podstatný podiel majú aj ďalšie online služby ako úložiská a mailové služby [2]. V norme RFC je phishing charakterizovaný ako: *Podvod formou sociálneho inžinierstva kde dochádza k podvrhnutiu identity s cieľom oklamania obetí. Toto vedie k prezradeniu osobných údajov za účelom finančného zisku* [16].

Obecná definícia phishing útoku je náročná kvôli vynaliezavosti autorov a neustálej obmene a vzniku nových techník. Podstatou Phishing úroku je spojenie technických a sociálnych aspektov za účelom zneužitia cenných dát. Tieto dáta vedú k finančnému zisku alebo inému obohateniu útočníka prípadne slúžia ako prvý krok pre ďalšiu infekciu malware. Zneužitie získaných prístupových údajov otvára možnosti pre útok typu ransomware, vytvorenie vzdialených prístupov a prevedenie rôznych útokov priamo v internom prostredí.

2.1 Špecifickosť Phishing útoku

Phishing podvody majú výraznú odlišnosť od ostatného typu malware a útokov čo sa odráža na procese ich detekcie a klasifikácie. Keďže cieľom je oklamanie človeka a potreba interakcie obete hlavným rozdielom je prepojenie technického a ľudského aspektu. Na zá-

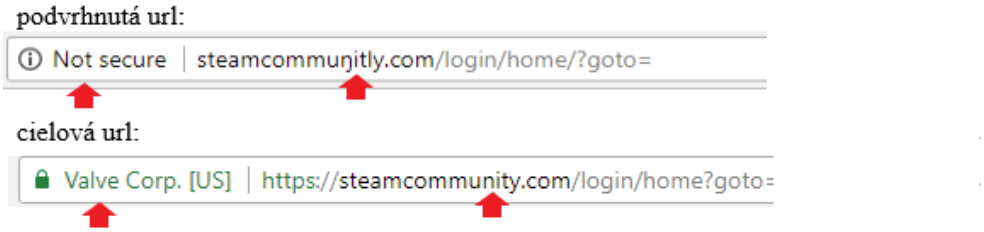
klade formy útoku vyžaduje interakcia vloženie dát do pripraveného formuláru, prípadne povolenie a spustenie makra v phishing dokumente.

Technické prevedenie phishing útoku na základe formulárov na odosielanie dát umiestnených na webových stránkach komplikuje využitie detekčných mechanizmov. Odosielanie údajov je legitímnou činnosťou a pre detekciu je možno využiť len jedinečné identifikátory umiestnené v rámci stránky.

Významným rozdielom phishing útokov oproti iným typom malware útokov je krátke trvanie tohoto útoku. Doba počas ktorej je tento útok aktívny je zväčša v rámci jednotiek hodín, čo má vplyv nielen na potrebu krátkej reakčnej doby ale i nemožnosť spätne získať vzorky útoku. Počas tejto doby je nutné útok nielen detegovať ale aj blokovať. Spojenie krátkej reakčnej doby a jednoduchosti prevedenia útoku a jeho obmien vedie k potrebe automatickej klasifikácie a detekcií phishing útokov.

2.1.1 Sociálne inžinierstvo

Dôležitou súčasťou phishing útoku je získanie dôvery obeti. K tomuto používajú útočníci niekoľko techník, hlavnou z nich je však vizuálne podobný dizajn s cieľovou stránkou. Podobný dizajn stránky je docielený skopírovaním zdrojového kódu stránky, vytvorením vierohodnej napodobeniny alebo použitím obrázku vytvoreného pomocou snímky obrazovky. Jedným z trikov je taktiež načítanie skriptov a CSS štýlov priamo zo stránok legitímnych cieľových spoločností.



Obr. 2.1: Porovnanie napodobeniny URL vo Phishing útoku a cieľovej URL.

V prípade podobného dizajnu je dôležitým ukazateľom URL [22] stránky. Dôveryhodnosť URL je možné získať napodobením cieľovej URL alebo vytvorením novej falošnej URL ktorá vzbudzuje dôveru. Napodobenie cieľovej URL v prípade phishing útoku na prihlasovaciu stránku služby `Steam` ukazuje obrázok 2.1. URL použitá pri phishing útoku využíva vizuálne podobné znaky pre oklamanie obeti, avšak rozdielny certifikát je zrejmý na prvý pohľad.

Použitie podobného dizajnu môže prispieť ku hodnovernosti stránky no pre zvýšenie interakcie s útokom sú použité techniky sociálneho inžinierstva, ktoré útočia na ľudskú stránku. Najdôležitejšou časťou útoku je oklamanie obeti a presvedčenie ku kliknutiu na odkaz, otvorenie prílohy alebo stiahnutie nástroja.

K vyvolaniu interakcie s phishing útokom využívajú útočníci pocit autority, kde napodobňujú vládnu organizáciu, vyšší manažment alebo spolupracovníka firmy. Použitie autority vyvoláva vyšší pocit dôležitosti a potrebu reakcie. Využitím tejto techniky sú falošné

dokumenty od daňových úradov, dôležitá správa od CEO spoločnosti, prípadne faktúra s výzvou na zaplatenie.

Pre zvýšenie miery úspešnosti útoku slúži tiež vyvolanie urgencie kedy je potreba vykonať akciu pre zabránenie drastickým akciám ako zablokovanie účtu, zmazanie dát alebo prepadnutie peňazí. Myšlienkou tejto techniky je donútiť obeť konať rýchlejšie ako premýšľať a neoveriť si skutočnosť predkladaných údajov. Častou aplikáciou sú peniaze a finančné účty, prípadne žiadosť o potvrdenie výhry v lotérií.

2.1.2 Motivácia útočníkov

Najdôležitejšou motiváciou útočníkov je finančný zisk. Na temnom webe (*Dark web*) sa predávajú kradnuté účty, či priamo osobné údaje. S týmto súvisia aj typy návnad, ktoré podľa správy `APWG` pre tretí štvrťrok 2018 [2] najčastejšie cielia na banky, finančné služby, krádež kreditných kariet, online mien, či ďalšie online služby vedúce k zneužitiu účtov.

Okrem finančného zisku je jedným z dôležitých cieľov vlastná sláva útočníka. Potreba preslávit svoju identitu sa prejavuje umiestnením vlastných podpisov do šírených súborov. Útočník prostredníctvom slávy a napadnutých obetí preukazuje vlastné schopnosti. Toto je dôležitým psychologickým faktorom pre začínajúceho útočníka, ale slúži aj ako portfólio v prípade predaja svojich služieb online. Podpis umiestnený na stránke je jednoduché identifikovať kvôli jeho dizajnu a používaným výrazom. Ukážku podpisu množno vidieť na obrázku 2.2.

```
<tr><td>________HACKED BY FWBJ SeRvAr__________</td></tr>
```

Obr. 2.2: Reprezentácia útočníka umiestnením podpisu do zdrojového kódu phishing stránky.

2.2 Riziko Phishing útokov

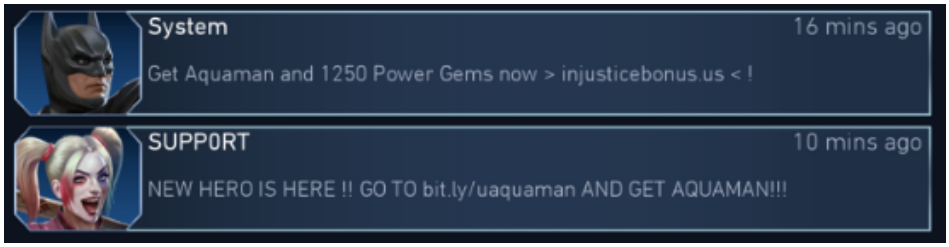
Cielený *Spear phishing* útok na vysoký manažment môže viesť ku zneužitiu prístupových údajov ku komunikačným prostriedkom, datovým skladom alebo interným nástrojom. Zneužitie týchto údajov vedie k prístupu útočníka do interných systémov a služieb čo má za následok odcudzenie dát. Okrem získania dôverných dát slúži phishing útok aj ku rozšíreniu ďalších útokov malware prostredníctvom získanych prístupových údajov.

Pri menej sofistikovaných útokoch je nevýhodou velkosť cielového publika na ktorom je možno útok previesť. Vo firemnom prostredí to znamená že zodpovednosť za nepodlahnutiu tomuto útoku nepreberá len sieťový administrátor a špecialisti ale aj všetci technický i netechnický pracovníci. Problematickými faktormi stále zostáva znovupoužívanie hesiel a možnosť umožnenia ďalších útokov.

2.2.1 Vektory útoku

Kvôli rôznym typom útoku sa využívajú rôzne formy šírenia útoku - vektory útoku. Vektor útoku značí metódu, ktorú využíva útočník pre rozšírenie svojho útoku alebo prekonanie zabezpečenia. Rozličné vektory vyžadujú rôzne úrovňe obtiažnosti, technologickej znalosti a majú inú úspešnosť v závislosti od znalostí obetí. Zoznam hlavných vektorov pre phishing útok:

Instant Messaging (IM) S rozvojom sociálnych sietí a rôznych platforiem na rýchlu komunikáciu sa rozvíjajú aj online podvody. Známe podvody zahŕňajú návnady s videom, citlivými informáciami o priateľoch, prípadne prémiové meny v online hrách. Na túto návnadu sa snaží útočník prilákať obeť pomocou URL odkazov alebo s využitím služieb URL skracovačov. Príklad takýchto správ ukazuje obrázok 2.3.



Obr. 2.3: Phishing využívajúci metódu IM pre rozšírenie odkazu na webovú stránku phishing útoku s cieľom odcudzenia užívateľských účtov v populárnej mobilnej hre.

Smishing používa na šírenie SMS správy. Kvôli limitovanej dĺžke textu a absencií obrázkov má útočník obmedzené možnosti pre vzbudenie dôvery. Ku vzbudeniu dôvery sú ako číslo odosielateľa využívané čísla SMS firiem. Pre vyvolanie reakcie sú dôležité techniky sociálneho inžinierstva, ktoré obeť navedú na návnadu. Kvôli limitom SMS a predchádzaniu odhalenia sú využívané služby skracujúce URL s možnosťou vytvoriť vlastný odkaz s krátkou dĺžkou.

Email je populárna forma komunikácie pri ktorej sú dôležitými časťami text emailu, prílohy a odosielateľ emailu. Útok prebieha zaslaním emailu, ktorý obsahuje odkaz na podvodnú URL stránku alebo infikovanú prílohu. Priložený text emailu má za úlohu vzbudiť dojem legitimity, zaujať pozornosť obeti a vyvolať reakciu. Pre zvýšenie legitimity emailu útočník podvrhuje adresu odosielateľa alebo využíva infikovanej užívateľskej stanice pre odoslanie emailu všetkým kontaktom priamo z účtu obeti. Vzorka Phishing emailu odkazujúca na podvodnú stránku je prezentovaná na obrázku 2.4.

Vishing je phishing over voice, teda phishing cez telefónne hovory. Patria sem podvody s cieľom získať prístupové údaje od obetí prevedením prieskumu firmy a nevyžiadaného telefonátu, v ktorom prebieha phishing útok. Ďalšou formou je služba prostredníctvom podvodnej technickej podpory, kde sa útočník snaží vytiahnuť od obete prístupové údaje alebo udržať ju čo najdlhšie v hovore, kvôli vysokým tarifom za dĺžku hovoru.

```
3550978361363406545=="
MIME-Version: 1.0
Date: Mon, 02 Jul 2018 20:56:13 +0900
From: ERM PLumbing Inc <mike@baziin.ga>
Message-Id: <7409125671.201872115613@mail1.mcsignup.com>
Subject: ACCOUNT#9414233-ERM PLumbing Inc
To: <REDACTED>

--===============3550978361363406545==
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 7bit


See the original invoice copy. This is the best copy we have,
with signature.

http://doinothientrieu.com/Client/Invoice-824185/
```

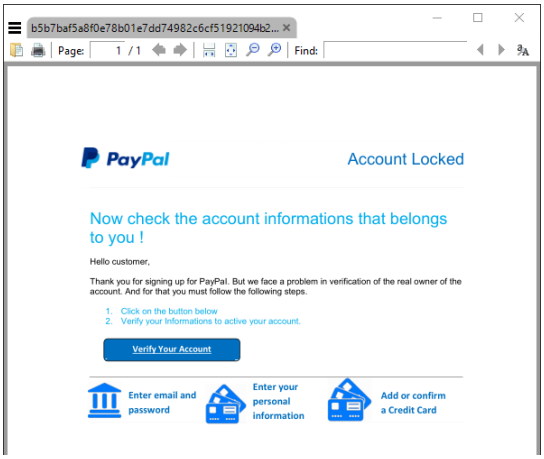
Obr. 2.4: Phishing email lákajúci obeť k navštíveniu URL prostredníctvom textu a adresy odosielateľa. Po navštívení webovej stránky útoku sú od obeti vyžadované prístupové údaje ku emailovému účtu.

Malwertising je technika, kedy sa útočník snaží dostať odkaz na svoju návnadu medzi poskytovanú reklamu na legitímnych stránkach. S využitím návrhu vyhľadávačov ako `Google`, ktorý na prvých priečkach výsledkov zaraďuje reklamu, je návnada viac dôveryhodná a je jednoduché zameniť si ju s legitímnymi výsledkami. Pre prevedenie tohoto útoku je potrebné oklamať obeť útoku na základe nadpisu webovej stránky a navrhnúť návnadu tak aby nedošlo k jej odhaleniu poskytovateľom reklamy.

Spear phishing je forma phishing útoku, kedy nie sú cieľom všetci ľudia ale iba presne vymedzená skupina ľudí. Útok zahrňuje prevedenie dopredného prieskumu cieľa a prípravu útoku na mieru na základe získaných znalostí. Unikátnosť jednotlivých útokov a ich personalizácia znamenajú, že je ťažšie ich rozlíšiť od nevyžiadaných avšak legitímnych emailov. Takto pripravený útok je taktiež viac sofistikovaný, teda náročnejší na odhalenie a viac dôveryhodný.

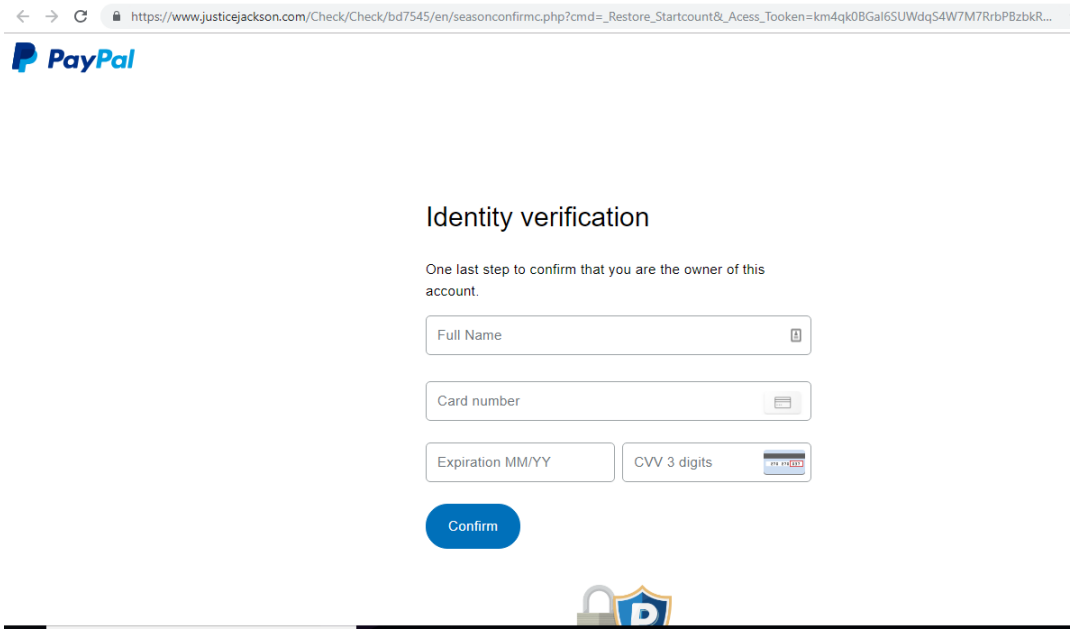
Phishing dokumenty pozostávajúce z PDF dokumentov alebo dokumentov z nástrojov sady kancelárskych balíkov sú pri phishing útoku používané s využitím samotného textu a dizajnu dokumentu pre získanie dojmu legitímnosti. V týchto prípadoch návnada pre obeť pozostáva z URL odkazu na ďalšiu vrstvu útoku ako v príklade na obrázku 2.5. Ďalšou formou útoku je použitie techník sociálneho inžinierstva pre vyvolanie interakcie s falošnými faktúrami alebo inými podvrhnutými dokumentami. Tieto dokumenty obsahujú makrá slúžiace k infekcií malware do zariadenia obeti.

Phishing vo webových stránkach využíva ako návnadu stránky, na ktoré sa snaží prilákať obeť. Táto forma Phishing útoku môže byť kombinovaná s ostatnými pre zvýšenie



Obr. 2.5: Príklad phishing dokumentu cieleného na získanie účtu v službe PayPal.

efektivity. Cieľom útoku je napodobniť dizajn cieľovej stránky alebo vytvoriť novú stránku s cieľom získať dôveru užívateľa. Útok prebieha pomocou vytvorenia formulárov alebo vstupných polí, ktoré odosielajú získané dáta útočníkovi. Príklad takejto podvodnej stránky ukazuje obrázok 2.6.



Obr. 2.6: Príklad Phishing stránky cielenej na spoločnosť PayPal, ktorá okrem prístupového účtu obeti získava aj ďalšie osobné údaje pre overenie totožnosti a údaje o kreditnej karte.

2.3 Technické aspekty

Phishing podobne ako iný malware vyžaduje technické prevedenie samotného ukradnutia osobných údajov. K tomu je potreba zaistiť návrh a prípravu útoku a pripraviť potrebné súbory na ich rozšírenie. Ďalej je nutné zaistiť ďalšie účty a služby ako zaistenie hostingu webovej stránky, emailu pre doručenie údajov, prípadne účtu v službe skracujúcej URL. Po zaistení potrebnej infraštruktúry je možné previesť rozšírenie súborov a zber údajov. Následne útočník uniká odhaleniu pozastavením a deaktivovaním útoku.

2.3.1 Príprava útoku

V prípade webového prevedenia útoku je potrebné vytvoriť všetky potrebné súbory a nasadiť ich na webový server, kde sa daný útok prevedie. Pre zjednodušenie manipulácie so súbormi a uľahčenie hromadného nasadenia sa tieto súbor zabalia do archívu nazývaného ako phishing kit.

Obr. 2.7: Obsah phishing kitu pre odcudzenie prístupových údajov na email účet obeti.

Phishing kit je najčastejšie archív `ZIP` formátu, ktorý obsahuje všetky potrebné súbory pre vykonanie daného útoku. Najdôležitejšou súčasťou sú HTML súbory so zdrojovými kódmi stránky, obrázky, CSS štýly pre vytvorenie dizajnu stránky a Javascript skripty pre rozšírenú funkcionalitu. Kvôli samotnému zberu dát na webovom serveri sú súčasťou aj PHP súbory pre príjem HTTP požiadavkov a zasielanie dát na email alebo ukladanie do logu. Ukážku súborov phishing kitu možno vidieť na obrázku 2.7.

Keďže je potreba len nahrať a rozbaliť archív, tak jeden kit je možno nasadiť na niekoľko rôznych stránok v krátkom čase. Tento útok je veľmi lacný s využitím infikovaných webserverov, sietí botnet prípadne hostingov poskytujúcich domény zdarma. Nevýhodou pre útočníka pri použití tohoto typu útoku je nemennosť súborov na rôznych cieľových stránkach, preto je ich detekcia dôležitá. V prípade detekcie týchto súborov nie je nutné blokovať každú URL na ktorú ich útočník umiestni. Použitie generických detekcií je odolné aj voči malým zmenám v zdrojovom prípade a má takto široké pokrytie rôznych phishing útokov.

2.3.2 Prevedenie útoku

Samotný dizajn stránky slúži na vzbudenie dôveryhodnosti obeti, pre získanie dát je však potrebná ďalšia funkcionalita. Toto je dosiahnuté použitým formulárov a textových polí ako ukazuje obrázok 2.6. Formulár sa využíva na odosielanie vložených dát do predpripraveného php skriptu, ktorý sa stará o zber a perzistenciu dát. Príkladom phishing útoku je priamy php skript v políčku *action*, ktoré sa nachádza vo formulári:

```
<form action="next.php" method="POST" autocomplete="OFF">
```

Pre uloženie dát od užívateľa sú pripravené polia pre email a heslo:

```
<input id="Passwd" required name="password" type="password"
placeholder="Password" class="">
```

Po vložení samotných dát je obeť navedená na pole pre prihlásenie do svojho účtu. Toto pole však v skutočnosti slúži len na spracovanie a odoslanie dát:

```
<input id="signIn" name="signIn" class="rc-button rc-button-submit"
type="submit" value="Sign in to view attachment">
```

```
<?
$ip = getenv("REMOTE_ADDR");
$message .= "--------------Dury Login Info-----------------------\n";
$message .= "Username : ".$_POST['j_username']."\n";
$message .= "Password : ".$_POST['j_password']."\n";
$message .= "IP : ".$ip."\n";
$message .= "---------------Created BY Unknown-------------\n";
$send = "michaelfleming664@yahoo.com";
$subject = "Result from Dury $ip";
$headers = "From: office<$ip@newlife.com>";
$headers .= $_POST['userid']."\n";
$headers .= "MIME-Version: 1.0\n";
$arr=array($send, $IP);
foreach ($arr as $send)
{
mail($send,$subject,$message,$headers);
mail($to,$subject,$message,$headers);
}
$fp = fopen("vip.txt","a");
fputs($fp,$message);
fclose($fp);

header("Location: https://www.wellsfargo.com/privacy-security/privacy/online");
?>
```

Kód 2.3.1: PHP skript pre zasielanie ukradnutých dát.

Podľa sofistikovanosti útoku sa útočník snaží nastražiť obeti stránku podobnú tej na ktorú je zvyknutá. Po vložení údajov do definovaných polí obeť zvolí možnosť odoslať tieto údaje kliknutím na tlačítko. Kliknutím na tlačítko pre odoslanie sa využije atribútu action značky form a údaje sa zašlú pomocou HTTP POST požiadavku do PHP skriptu.

PHP skript ako na výpise kódu 2.3.1 spočíva z metódy pre príjem dát a následne využíva metódu pre odoslanie týchto dát na email definovaný v špeciálnej premennej. V prípade zakúpenia hotového phishing kitu je práve cieľová adresa jediným údajom, ktorý musí útočník zmeniť pred nasadením kitu na webserver a spustením útoku.

Súčasťou typického phishing kitu je aj kód pre ochranu pred botmi a spoločnosťami zaoberajúcimi sa analýzou malware. Útočníci sa takto snažia vyhnúť detekcií ich útokov a zaručiť tak dlhšiu životnosť. V prípade prístupu na stránku phishing útoku z adresy definovanej na čiernej listine je užívateľovi predstavená falošná stránka. Táto falošná stránka predstavuje chybovú stránku o tom, že cieľová stránka nebola nájdená.

Napodobenie URL

Útočník vytvorí novú URL, ktorá pozostáva z kľúčových slov ktoré tematicky odpovedajú operácií s účtom. Príklad anglických kľúčových slov: account, secure, authorize, validate. Samotná stránka potom reprezentuje validáciu účtu, prípadne je použitý email s textom o podozrivej aktivite a potrebe validácie účtu. Toto všetko je spojené s praktikami sociálneho inžinierstva.

Útočník sa snaží napodobniť URL, ktorá reprezentuje legitímnu spoločnosť. V tomto prípade sa jedná o URL, ktoré sú dosiahnuté použitím preklepov, pridaním znakov alebo použitím názvov značky, ktoré cieľová spoločnosť nevyužíva. Kombinácia napodobenia URL a použitia techník sociálneho inžinierstva má za výsledok využitie klúčových slov a napodobených slov značiek. Tieto URL nie sú vizuálne podobné cieľoovej URL ale majú za úlohu dopĺňať kontext phishing útoku. Príkladom sú phishing útoky zamerané na klúčové slová aktivácie účtu a zablokovania účtu.

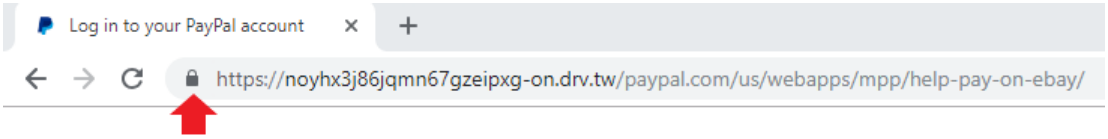
Ďalším spôsobom je použitie homografov v útoku (homograph attack), ktorý pozostáva z použitia rozdielnych ale vizuálne podobných znakov. Pri tomto útoku sa využívajú znaky mimo základnej znakovej sady najčastejšie z latinskej alebo cyrilskej abecedy. V tomto prípade je URL voľným okom takmer nerozlíšitelná a jediným spôsobom je kontrola URL v originálnej forme zakódovanej prostredníctvom Punycode [5]. Príklad takéhoto útoku vidno na obrázku 2.1.

SSL certifikáty

Spolu so zvyšujúcou sa dostupnosťou lacných domén útočníkom prospievajú aj nízke náklady a jednoduchý proces pre získanie HTTPS certifikátu pre danú webovú stránku. Zvyšujúci trend používania HTTPS certifikátov je prezentovaný aj v APWG správe [2] pre tretí štvrťrok 2018, ktorá hovorí, že až 50% Phishing útokov využíva HTTPS.

Dôležitosť použitia HTTPS na doméne prospieva ku vzbudeniu dôvery obeti. Moderné prehliadače označujú HTTPS stránky zeleným štítom prípadne ikonou zámku čo u ľudí

vzbudzuje dojem bezpečnosti. V skutočnosti však HTTPS protokol zaručuje len šifrovanie spojenia medzi užívateľom a webovou stránkou čo znemožňuje odchytenie hesiel útokmi Man in the middle (MItM) [23] nie však odchyteniu hesiel na cieľovej stránke. Využitie HTTPS pri phishing útoku na `PayPal` vidieť na obrázku 2.8.



Obr. 2.8: Indikátor bezpečného zámku HTTPS v prehliadači `Google Chrome`.

Využitie HTTPS certifikátu danou doménou však umožňuje jednoduché preukázanie identity vlastníka. Vďaka reťazcu dôvery (chain of trust) [14] je možné detegovať phishing útok cielený na stránku Google porovnaním certifikačných autorít daného certifikátu s certifikačnou autoritou Google. Keďže nikto iný ako vlastník privátnych kľúčov nemôže svoj certifikát podpísať týmito kľúčmi môžeme ich považovať za preukázanie vlastníctva stránky.

Kryptomeny

S nárastom popularity a hodnôt kryptomien narástli aj prípady podvodov kde sú kryptomeny zneužívané. Útočníci v prípade útoku na digitálne peňaženky svojich obetí získavajú priamy zisk bez nutnosti ďalej speňažovať získané dáta. V prípade zisku prístupových údajov je možno priamo previesť získanú menu na iný účet.

Tieto podvody nezneužívajú prístupové údaje do online platforiem, ale zakladajú na princípoch falošných lotérií alebo podvodoch typu Nigerijský princ [4]. V týchto prípadoch sa snažia vylákať od svojich obetí priamo cieľovú menu s prísľubom ďalšieho niekoľkonásobného zisku. Pre získanie dôvery zneužívajú napodobeniny identít známych osobností ako Elon Musk, prípadne použijú priamo prelomené účty.

V prípade úspešného útoku majú útočníci výhody pseudoanonymity a zťaženého či nemožného sledovania ukradnutých mien. Kvôli prevažnej decentralizácií väčšiny mien transakcie nemožno zrušiť a tak v prípade odhalenia podvodu je možno len zablokovať cieľové účty.

Kapitola 3

Existujúce anti-phishing nástroje

Na detekciu a klasifikáciu phishing útokov existuje množstvo nástrojov. Tieto nástroje sa odlišujú v používaných technikách detekcie, formáte vstupných dát a dosiahnutých výsledkoch. Niektoré z dostupných nástrojov neposkytujú informácie o ich internej funkcionalite, teda je možno porovnávať ich len na základe použitých techník a testovaním na pripravených testovacích dátach. Kvôli množstvu týchto nástrojov sú predstavené len nástroje ktoré sa vyznačujú svojimi vlastnosťami alebo sú používané odborníkmi na analýzu phishing útokov.

3.1 Typy spôsobu detekcie

Typ spôsobu detekcie nástroja prináša so sebou výhody ale i nevýhody. Medzi hlavné metriky patria miera falošne pozitívnych a falošne negatívnych detekcií. Ďalšie rozdiely predstavujú rýchlosť aktualizácie modelu pre detekciu, rýchlosť vyhodnocovania alebo schopnosť rozlišovať predom neznáme vzory. Na základe týchto vlastností je zrejmé, že rôzne techniky sú vhodné v odlišných aplikáciach.

3.1.1 Blacklisting

Zaradenie na čiernu listinu (*Blacklisting*) predstavuje jeden z najjednoduchších spôsobov ako blokovať phishing útoky. Tento spôsob spočíva zo zaradenia vzorky na zoznam blokovaných vzoriek na základe jeho URL, hashi SHA obsahu alebo iného vzoru charakterizujúceho vzorku alebo skupinu vzoriek. Klasifikácia samotná je na základe analytickej činnosti alebo ďalších mechanizmov.

Hlavnou nevýhodou je potreba udržiavať aktuálnu čiernu listinu. Doba od získania vstupných dát až po blokáciu závisí na dobe spracovania a klasifikácie, následného zaradenia na zoznam a stredná doba intervalu aktualizácie zoznamu medzi klientami. Pre urýchlenie tohto procesu možno použiť distribúciu len zmien v zozname, prípadne pridať zmazanie starých a nepoužívaných záznamov.

Jednou z ďalších nevýhod je nemožnosť detekcie predom neznámych vzoriek je detekcia na základe jednoznačného identifikátoru vzorky. Pri zmene tohoto identifikátoru alebo

analýze novej vzorky rovnakej rodiny je potrebné ju znovu zaradiť na listinu. Tento faktor znamená, že použitím zoznamu nemožno detegovať vopred neznáme vzorky (*0 Day detection*).

3.1.2 Heuristické funkcie

Detekcia na základe heuristických funkcií spočíva z definície funkcií nad danými vzorkami tak, aby bolo možné klasifikovať vzorky na základe ich výsledku. Tieto funkcie definujú charakteristiky vzorovej vzorky a pre ich definíciu je potreba znalosť týchto charakteristík a ich obmeny. V prípade detekcie malware alebo phishing sa jedná o funkcie ktoré definujú podozrivú funkcionalitu, chovanie alebo štruktúru vzorky. Príkladom takýchto funkcií je počet odkazov v texte, hash použitých skriptov, štýlov alebo obrázkov. Úlohou týchto funkcií je tak zjednodušiť analýzu a detekciu na základe vytvorenia funkcie pre dané chovanie, čo má za následok možnosť rýchleho označenia podozrivého chovania vo všetkých vzorkách.

Na rozdiel od techniky čiernych listín heuristické funkcie umožňujú detegovať a klasifikovať predom neznáme vzorky. Pre detekciu takýchto vzoriek stačí definícia chovania pre špecifický typ malware alebo jeho rodiny. V tomto prípade definovaná funkcia funguje aj v prípade malých zmien vo vzorke.

3.1.3 Strojové učenie

Strojové učenie (*Machine Learning*) je technika, ktorá využíva matematické modely a algoritmy pre rozhodovanie nad danými dátami. Pri použití klasifikačných algoritmov je cieľom klasifikovať vzorku teda prideliť jej jednu zo známych tried. V prípade klasifikácie súborov tieto triedy predstavujú: čistý, podozrivý a phishing súbor.

Hlavným problémom pri tvorbe kvalitného modelu pre klasifikáciu je potreba vhodnej dátovej sady. Dátová sada spočíva zo vstupných dát náhodne rozdelených na dáta určené na trénovanie a testovanie. V prípade nevyvážanej dátovej sady voči niektorej z vlastností dôjde k nekorektnému naučeniu modelu kvôli nedostatočnému nastaveniu váh voči očakávanému výsledku. Použitie takejto dátovej sady má za výsledok nízku presnosť modelu.

Učenie spočíva v spúšťaní daného modelu nad vstupnými dátami. Počas tohto procesu model adaptuje váhy, ktoré reagujú na dané vstupy. Model, ktorý je naučený má svoje váhy nastavené tak, aby pre daný vstup úspešne priradil očakávanú triedu na základe testovacích dát.

Výhodou tejto techniky je možnosť detegovať predom neznáme vzorky a jeho odolnosť voči jemným zmenám vstupných súborov. V prípade použitia nových techník pri phishing útokoch je však potrebné zaradiť ich do dátovej sady na trénovanie modelu. Pri zmenách vstupnej dátovej sady je tento model potrebné znovu natrénovať, čo môže byť zdĺhavý proces v závislosti na použitých algoritmoch. V prípade využitia rozličnej dátovej sady od reálnych dát môže dôjsť k podobnému problému ako pri nevyváženej dátovej sade.

3.2 Spôsoby reprezentácie útoku

Nástroje pre detekciu phishing využívajú nielen rôzne techniky ale taktiež sa zameriavajú na rôzne typy vstupných dát. Tieto vstupné dáta predstavujúce malware, teda aj phishing sú spoločne nazývané pod pojmom vzorka (*sample*). Prostredníctvom techník analýzy malware (*malware analysis*) možno získať viac informácií o daných útokoch a získavať vzorky pre ich reprezentáciu. Vzorky útokov sú využívané pre zdieľanie informácií, kategorizáciu útokov, trénovanie a testovanie nástrojov. Vzorku phishing útoku možno vnímať ako súbor, textovú reprezentáciu, štruktúru v danom komunikačnom programe či snímku stránky. Na základe rôznych typov vzoriek existujú rôzne prístupy nástrojov ku ich analýze a detekcií.

3.2.1 Súbor

Detekcia phishing útoku na základe obsahu súboru je jednou z najbežnejších foriem detekcie malware. V tomto prípade je v rámci analýzy dostupný celý obsah súboru a možnosť vytvárať textové a heuristické detekcie na základe reťazcov a funkcionality vzorky. V prípade detekcie vo väčšej škále je možno tieto súbory deliť do skupín na základe ich vlastností.

Dôležitou metódou je technika hashovania súborov. Pri použití hashovacích funkcií ako napríklad SHA-256 je možno vytvoriť unikátne ID pre každý súbor, čo je výhodné pri zdieľaní informácií či ukladaní výsledkov. Množstvo informácií v súbore však umožňuje vytvárať viac generické hashe, ktoré umožňujú spájať podobné vzorky do rodín alebo zhlukov.

3.2.2 URL

V prípade detekcie vzoriek pozostávajúcich z URL nie je možné analyzovať obsah súboru, ale analýza si musí vystačiť len so zdrojovou URL. Analýza má obmedzený počet dostupných informácií a preto je o to viac dôležitejšie správne navrhnutie heuristík. Heuristické funkcie zakladajú na statických vlastnostiach URL, častiach schémy URL ale aj dynamických vlastnostiach pri navštívení a sťahovaní dokumentu z URL.

Pridávanie URL na čierne listiny predstavuje efektívnu metódu ako detegovať a blokovať nežiadúce stránky. Táto metóda predstavuje jednoduchý a rýchly spôsob ako zaradiť špecifické vzorky a potlačiť ich. Hlavnou nevýhodou tejto metódy je však náchylnosť voči zmenám. V prípade malej zmeny zdrojovej URL je potreba túto URL znovu analyzovať a zaradiť na čiernu listinu.

3.2.3 Email

Nástroje zaoberajúce sa detekciou phishing v emailoch a textových správach majú za úlohu ochrániť užívateľa pred phishing útokom odhalením odkazov a infikovaných príloh. Analýza správ sa zakladá na textových charakteristikách ako prítomnosť kľúčových slov, štruktúra textu a známe reťazce phishing útokov. Okrem analýzy textu sú dôležité metadáta, odkazy URL alebo dokumenty v prílohe.

3.2.4 Snímky obrazovky

Nástroje pre detekciu na základe snímky obrazovky spoliehajú na vizuálnu podobnosť cieľovej stránky phishing útoku a danej vzorky predstavujúcej snímku návnady. Vstupná snímka obrazovky je reprezentovaná dátami vhodnými pre ďalšie spracovanie ako sú hashe, vektory a štruktúry. Pri tomto prístupe je dôležitý správny návrh techniky konverzie obrázku na dáta pre ďalšiu analýzu. V prípade použitia príliš obecnej konverzie má model nízku presnosť, zatiaľ čo pri použití príliš špecifických techník dochádza k neúspešným porovnaniam vzoriek.

Tento spôsob detekcie má výhodu pri použití techniky obfuskácie (obfuscation) v zdrojových súboroch stránky. Obfuskácia zahŕňa skrývanie reálnej funkcionality kódu jeho zakódovaním, rozdelením a spájaním po častiach prostredníctvom premenných a polí. V takto pozmenenom zdrojovom kóde je ťažké zistiť výsledné chovanie a teda je aj ťažší na automatickú klasifikáciu. Vďaka nezávislosti na zdrojových kódoch stránky je v prípade analýzy výsledných snímkov obrazovky možné obfuskácie v kóde ignorovať. Výsledná analýza tak prebieha na rovnakom dizajne stránky ako pri navštívení stránky prostredníctvom klasického webového prehliadača.

3.3 Dostupné anti phishing nástroje

Na detekciu phishing útokov je zameraných niekoľko dostupných nástrojov. Tieto nástroje sú rozličné v technikách detekcie phishing útokov, formáte vstupných dát a teda dosahujú rozličnú úspešnosť. Nasleduje prehľad nástrojov špecifických niektorou z charakteristík nástrojov.

PhishTank

`PhishTank` [17] je nástroj, ktorý slúži ako databáza vzoriek phishing útokov. Vzorky phishing útokov na `PhishTank` spočívajú z URL a snímky stránky. Dáta sú nahrávané komunitou užívateľov a doplnené ďalšími zdrojmi. Verifikácia phishing útoku vo vzorkách spolieha na manuálnu analýzu užívateľov. Okrem tejto funkcionality poskytuje `PhishTank` API v ktorom zprístupňuje všetky svoje dáta ku phishing vzorkám.

VirusTotal

`VirusTotal` [25] je platforma slúžiaca na skenovanie súborov a URL adries pomocou produktov partnerských anti-vírusových spoločností. Na klasifikáciu využíva produkty tretích strán a výsledky sprístupňuje cez webové rozhranie a API.

Google Safebrowsing

Google Safebrowsing [7] je produkt spoločnosti `Google` integrovaný do jej produktov slúžiaci na chránenie klientov pred hrozbami. Riešenie je postavené na modeli strojového učenia, ktorý vyhodnocuje navštívené URL a súbory, posilnenom o ďalšie analytické nástroje.

Na základe tejto klasifikácie sa vytvára čierna listina malware a phishing, ktorá je následné distribuovaná klientom.

isitPhishing

`IsitPhishing` [9] je webová služba, ktorá využíva heuristické funkcie a strojové učenie pre detekciu phishing útokov. Služba poskytuje API pre klasifikáciu vzorkov v reálnom čase. Pre užívateľov stránky je možné prihlásiť sa na odber upozornení o phishing útokoch na vybranú značku.

PhishingAI

`PhishingAI` [13] je produkt spoločnosti `Lookout` založený na strojovom učení určený na detekciu phishingových stránok a phishing kitov. Nástroj je špecifický publikovaním zaujímavých výsledkov prostredníctvom služby `Twitter` umožňujúc zdielať vzorky a nové techniky.

Kithunter

`Kithunter` [19] je nástroj slúžiaci na detekciu phishing kitov prostredníctvom skenovania dostupných priečinkov online web serverov. Nástroj je špecifický svojím zameraním priamo na detekciu existencie phishing kitov, čím umožňuje detegovať priamo zdroj phishing útoku. Tento nástroj je vhodný pre správcov webserverov pre rýchlu lokalizáciou útoku a vyčistenie systému.

3.4 YARA

`YARA` je nástroj a jazyk vyvíjaný spoločnosťou `VirusTotal` zameraný na detekciu a klasifikáciu malware vzoriek na základe vytvorených vzorov alebo pravidiel [26]. Nástroj `YARA` je využívaný mnohými spoločnosťami zaoberajúcimi sa online bezpečnosťou a analýzou malware. Vďaka rozšíreniu medzi týmito spoločnosťami je využívaný ku zdieľaniu identifikátorov zneužitia počítača (Indicator of Compromise, IOC). Pre zdieľanie správania daného malware stačí vytvoriť `YARA` pravidlo, ktoré deteguje tieto IOC - vytvárané súbory, procesy alebo systémové zámky.

Pomocou `YARA` možno vytvoriť pravidlá využívané pri skenovaní vstupu na známe vzory. V prípade zhody reťazca vo vstupnom súbore a reťazcov definovaných v rámci pravidla sa vyhodnotí pravidlo ako splnené. Na základe splnených pravidiel možno detegovať phishing útoky a klasifikovať vzorky podľa typu zhodných pravidiel.

Výhodou nástroja `YARA` je jednoduchosť definície vlastných vzorov a kompatibilita so známymi sandbox systémami využívanými na analýzu malware. Po definícií vlastných pravidiel tak možno zapojiť tento nástroj do existujúcej infraštruktúry.

Vďaka možnosti definície vlastných vzorov pre klasifikáciu možno vytvoriť niekoľko rôznych pravidiel na základe útoku. Tieto pravidlá môžu byť generické, určené na detekciu známych techník využívaných pri týchto útokoch alebo podozrivých reťazcov. V prípade

použitia reťazcov špecifických pre danú vzorku možno na základe unikátnych reťazcov klasifikovať vzor do rodín a kmeňov.

Pravidlá pre detekciu možno okrem rozdelenia na generické a špecifické pravidlá taktiež deliť na privátne a verejné pravidlá. Privátne pravidlá pri svojom splnení neklasifikujú vzorky a nemajú zvláštny výstup, ale sú používané v iných pravidlách. Týmto spôsobom možno vytvoriť generické pravidlá pre phishing útoky a tieto pravidlá znovupoužívať v iných viac špecifických pravidlách. Takýto dizajn pravidiel umožňuje modularitu a znovuvyužitie kódu.

3.4.1 Štruktúra pravidla

YARA pravidlo predstavuje štruktúru skladajúcu sa z niekoľkých polí popisujúcich informácie o pravidle pre ďalšiu analýzu a samotné podmienky pravidla. Príklad pravidla pre klasifikáciu phishing útokov ukazuje kód 3.4.1.

```
rule infected_09_03_18_phish_server {
   meta:
      description = "phish - file server.php"
      author = "Brian Laskowski"
      reference = "https://github.com/Hestat/lw-yara/"
      date = "2018-09-03"
      hash1 = "1c9066dd9b1d91a0cc9278629f7f0f8c7a6b9f9e0ebb1e739dd210f7a03ec025
          "
   strings:
      $s1 = "$ip_data = @json_decode(file_get_contents(\"http://www.geoplugin.
          net/json.gp?ip=\".$ip));" fullword ascii
      $s2 = "mail($own,$subj,$msg,$headers);" fullword ascii
      $s3 = "<?php"
   condition:
       ( all of them )
}
```

Kód 3.4.1: YARA pravidlo pre detekciu phishing útoku. Prevzaté zo sady pravidiel [11].

Prvým údajom je názov pravidla, ktorý slúži pre rýchlu identifikáciu vzoru v prípade definície množstva pravidiel. Ďalej sa pravidlo sa skladá z nasledujúcich polí:

meta Doplňujúce informácie zahŕňajúce popis pravidla, identifikáciu vzoru pri splnení pravidla a jeho autora. Ďalšie polia môžu definovať rodinu a kmeň malware vzorky, hodnovernosť pravidla, dátum vzniku alebo iné užívateľsky definované polia. Definované polia sú dôležité pri triedení pravidiel alebo ich zdielaní a nesú informačnú hodnotu o vzorkých na ktorých boli splnené.

strings Pravidlá využívajúce textové reťazce, ktoré sú porovnávané so vstupom špecifikujú tieto reťazce v tejto položke. Na základe týchto reťazcov je možné špecifikovať podmienku pravidla. Tieto reťazce možno definovať v rôznych kódovaniach, prípadne na ich mieste špecifikovať regulárne výrazy.

condition Podmienka pravidla určuje kedy nastáva zhoda pravidla so vzorom. Podmienky YARA pravidla podporujú logické, aritmetické, relačné a bitové operácie. Okrem týchto základných operácií podporujú ďalšie kontrolné operácie, podmienky nad kolekciami a referencie ostatných polí.

V sekcií `strings` pravidla možno definovať nasledujúce typy reťazcov:

Hexadecimálne reťazce s možnosťou definície zástupných znakov, sekvencií a skokov.

Textové reťazce s podporou ďalších modifikátorov ako: ascii, wide, nocase, fullword.

Regulárne výrazy s formátom ako textové reťazce ohraničené znakmi lomítka "/".

3.4.2 Moduly v YARA

Okrem reťazcov je možno v podmienke používať aj štruktúry a funkcie definované modulami. Pre použitie modulu stačí použiť výraz `import <názov modulu>` a následne sú dostupné všetky definície z modulu. Výhodou modulov je možnosť využívať užívateľsky definované štruktúry a funkcie, ktoré majú lepšiu vyjadrovaciu schopnosť než základné reťazce použité v pravidlách.

Na základe definícií v moduloch je možné do pravidiel pridať kontext a štruktúru k daným reťazcom. Pravidlo vo výpise kódu 3.4.2 ukazuje použitie modulu `Cuckoo` pre definíciu vzoru s využitím HTTP dotazu na doménu `http://someone.doingevil.com`. Toto prináša väčšie vyjadrovacie možnosti oproti použitiu základných reťazcov. V prípade použitia reťazca v tomto vzore by došlo k splneniu pravidla pri spomenutí domény kdekoľvek v texte. Tento fakt sa môže odzrkadliť na miere falošných detekcií napríklad na technických blogoch, v ktorých sa spomínajú nové hrozby.

```
import "cuckoo"

rule evil_doer
{
    strings:
        $some_string = { 01 02 03 04 05 06 }

    condition:
        $some_string and
        cuckoo.network.http_request(/http:\/\/someone\.doingevil\.com/)
}
```

Kód 3.4.2: YARA pravidlo s využitím Cuckoo modulu. Prevzaté z online dokumentácie YARA [27].

3.4.3 Aplikácia pravidiel

Aplikácia `YARA` pravidiel pozostáva z volania `yara <pravidlá> <vstup>`, kde argument <vstup> reprezentuje vstupný súbor alebo priečinok a argument <pravidlá> reprezen-

tuje cestu k súboru obsahujúcom pravidlá. Tieto pravidlá sú následné aplikované nad daným vstupom a v prípade ich splnenia sú zaradené na výstup. Výstup má formát `<názov pravidla> <vstupný súbor>` pre každé splnené pravidla. Takýto formát výstupu ukazuje na dôležitosť zavedenia správnych konvencií pre pomenovanie pravidiel.

3.4.4 Avast YARA

Implementácia nástroja `YARA` vyvíjaná v spoločnosti `Avast Software` reprezentuje modifikovaný nástroj `YARA` pre interné účely detekcie a klasifikácie malware v rámci tejto spoločnosti. Modifikácie tohoto nástroja predstavujú úpravy a rozšírenie funkcionality jednotlivých modulov ako napríklad modulov `Cuckoo` a `Pe`. Okrem modifikácií existujúcich modulov sú implementované aj vlastné moduly a to hlavne modul pre klasifikáciu `Android` aplikácií. Ďalšie modifikácie spočívajú v nastavení interných vyrovnávacích pamätí kvôli väčšej veľkosti interných pravidiel.

3.5 Zhrnutie

Dostupné existujúce nástroje používajú rôzne techniky na detekciu phishing útokov a sú určené na detekciu rozdielnych vstupných dát. Najpopulárnejšie nástroje využívajú kombináciu techník strojového učenia obohatených o ďalšie techniky ako čierne listiny. Využitie týchto techník vedie k vysokej úspešnosti detekcie avšak neponúka priestor pre podrobnú klasifikáciu rozdielnych vzoriek.

Nástroj `YARA` ponúka možnosti pre definíciu vlastných pravidiel, ktoré sú však nedostatočné na definíciu pravidiel na základe chovania, štruktúry alebo ďalších dynamických vlastností. Pre využitie týchto možností je potrebná definícia vlastného modulu, ktorý ponúka väčšiu vyjadrovaciu silu a umožňuje ďalšiu analýzu.

Dostupné nástroje umožňujú detegovať phishing útoky a teda vytvoriť klasifikáciu vzorky. Použitím týchto nástrojov však nie je možné získať ďalšie informácie o útoku ako je jeho rozšírenie, cieľová spoločnosť, podobnosť voči iným útokom. Nástroj `YARA` umožňuje definíciu pravidiel ktorými je možno definovať známe vzory vo phishing útokoch a teda nielen vytvoriť klasifikáciu ale i získať ďalšie informácie o analyzovaných vzorkách. Nástroj `YARA` pri analýze phishing vzoriek využíva len reťazce obsiahnuté v obsahu vzorky a nie je možné využívať charakteristiky vzorky. Pre rozšírenie možností nástroja `YARA` je potrebné vytvoriť nástroj pre extrakciu charakteristík a modul pre načítanie charakteristík pri skenovaní pravidiel. Takýto nástroj umožňuje určenie charakteristík vzorky na základe splnených pravidiel nad modelom vzorky.

Kapitola 4

Návrh nástroja

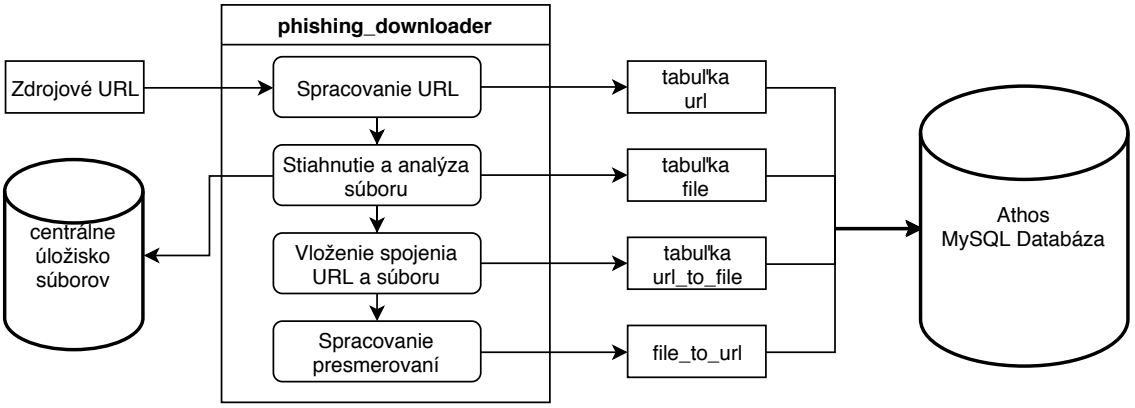
V tejto kapitole je navrhnutý nástroj `PhishCore`, ktorý spája popísané techniky pre analýzu súborov a URL vo forme hybridnej analýzy. Technika hybridnej analýzy je použitá pre vytvorenie automatickej klasifikácie vzoriek webových stránok na základe modelu vstupných dát. Vstupné dáta nástroja `PhishCore` sú získané z databázy pomenovanej `Athos`, ktorá je určená na ukladanie dát o phishing vzorkách v rámci firmy `Avast Software`. Naplnenie databázy dátami spracovanými zo zdrojov phishing URL pomocou nástrojov tretích strán popisuje sekcia 4.1. Architektúra nástroja `PhishCore`, rozdelenie na komponenty *Extraktor*, *Skener* a *Klasifikátor*, vstupy a výstupy komponent sú popísané v sekcií 4.2. Komponenty nástroja `PhishCore` sú popísané v sekciách 4.4, 4.5 a 4.6.

Použitím techniky hybridnej analýzy súboru a jeho zdrojovej URL je možné z danej vzorky phishing extrahovať väčšie množstvo charakteristík. Na základe charakteristík extrahovaných z phishing vzorky je vytvorený *model vzorky*. Na model vzorky sú aplikované `YARA` pravidlá a splnené pravidlá určujú výslednú klasifikáciu vzorky phishing. Prostredníctvom nástroja `PhishCore` je tak možno automaticky vytvoriť klasifikáciu phishing vzorky, ktorá má využitie pri triedení vzoriek phishing, odhaľovaní chýb v zdrojoch phishing a vytváraní definičných vzorov v produkte `Avast Software`.

4.1 Predpríprava a spracovanie dát

Pre aplikáciu techniky hybridnej analýzy v nástroji `PhishCore` je vstupom nástroja vzorka phishing vo formáte zdrojovej URL, súboru stránky a metadát získaných pri jej spracovaní. Zdroje phishing však zverejňujú len URL phishing stránok a preto je potrebné získať zdrojové súbory týchto stránok. K získaniu zdrojových súborov je využitý nástroj `phishing_downloader` vyvíjaný vo firme `Avast Software`.

Nástroj `phishing_downloader` má na vstupe zdroje URL a informácie o nich ukladá do databázy `Athos`. Pri spracovaní zdrojových URL najskôr URL spracuje podľa HTTP schémy [22] a uloží dáta o zdrojovej URL do tabuľky `url`. Následne je obsah URL stiahnutý a súbor na danej URL je spracovaný a uložený do existujúceho centrálneho úložiska súborov v rámci firmy `Avast Software`. Informácie o súbore sú uložené do tabuľky `file`. Relácia



Obr. 4.1: Spracovanie zdrojových URL pomocou nástroja `phishing_downloader` s využitím `MySQL` databázy `Athos` pre uloženie vstupných dát a metadát spracovávaného súboru a URL.

zdrojovej URL a stiahnutého súboru je uložená v tabuľke `url_to_file`. Na záver je v súbore prevedená detekcia techník pre presmerovanie na inú URL a táto informácia je uložená v tabuľke `file_to_url`.

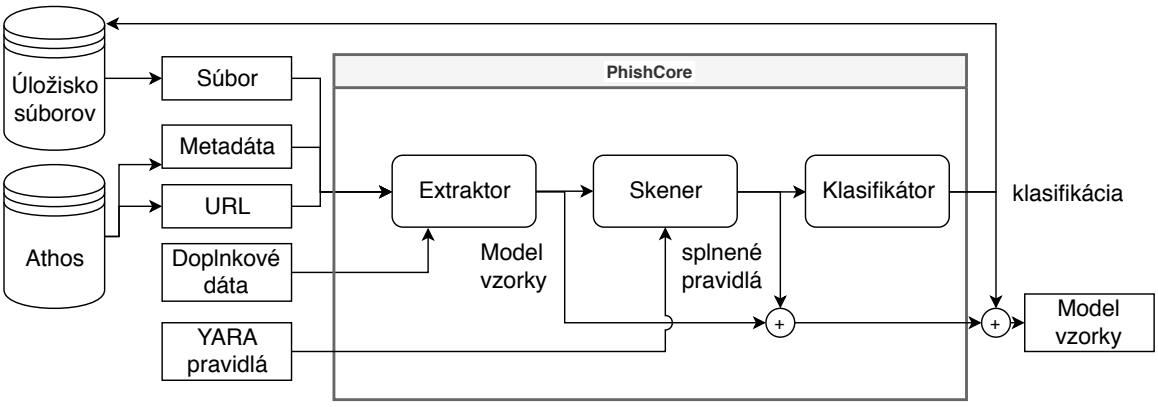
Dáta uložené v databáze `Athos` slúžia na manuálnu analýzu a ďalšie spracovanie inými nástrojmi. Na základe týchto dát možno napríklad sledovať rozšírenie a podobnosť jednotlivých rodín phishing útokov, cieľové stránky phishing kampaní a sledovať históriu zdrojových URL pre vybrané súbory. Kvôli chybám v zdrojoch URL, prípadne zmene obsahu webových stránok dochádza ku zamiešaniu vzoriek neobsahujúcich phishing do databázy. Vzorky obsiahnuté v databázy je tak potreba klasifikovať, teda zaradiť ich medzi čisté, podozrivé alebo vzorky obsahujúce phishing.

4.2 Architektúra nástroja

Nástroj `PhishCore` využíva ako svoj vstup dáta uložené do databáze `Athos`. Vstupné dáta z databázy `Athos` sú odoslané na vstup nástroja `PhishCore` a použité pri vytvorení modelu vzorky. Výsledná klasifikácia phishing vzorky je nakoniec uložená v databázy `Athos` a umožňuje triedenie a ďalšie spracovávanie vzoriek.

Nástroj `PhishCore` možno rozdeliť do niekoľkých logických celkov, kde jeden celok tvorí jedna *komponenta*. Architektúra nástroja `PhishCore` zobrazená na obrázku 4.2 ukazuje rozdelenie na jednotlivé komponenty a vstupy a výstupy komponent. Komponenta *Extraktor* spracováva vzorku phishing na vstupe nástroja `PhishCore` a na základe extrahovaných charakteristík a heuristík vytvára model vzorky. Výsledný model vzorky je vstupom komponenty *Skener*, ktorá nad týmto modelom aplikuje `YARA` pravidlá. Model vzorky je následne doplnený o `YARA` pravidlá, ktoré boli po aplikácií nad modelom vzorky splnené. Položka triedy klasifikácie uložená v splnených pravidlách je využitá v komponente *Klasifikátor*, ktorá vytvorí výslednú klasifikáciu modelu phishing vzorky, ktorá je pridaná do modelu

vzorky. Výsledkom spracovania phishing vzorky je model vzorky doplnený o výstupy komponent. Tento model nesie všetky charakteristiky, heuristiky a známe vzory a na základe klasifikácie modelu tak možno previesť ďalšiu analýzu phishing vzoriek.



Obr. 4.2: Architektúra nástroja PhishCore.

4.3 Model vzorky

Model vzorky slúži na reprezentáciu phishing vzorky počas procesu analýzy a poskytuje rozhranie pre jednoduchý prístup k objektom využívaným pri analýze. Model vzorky pozostáva z charakteristík vzorky získaných z komponenty `Extraktor`, splnených pravidiel získaných z komponenty `Skener` a klasifikácie z komponenty `Klasifikátor`. Model vzorky obsahuje nasledujúce polia:

file_contents (objekt) Model súboru na základe obsahu súboru.

file_analysis (objekt) Heuristiky definované na základe modelu súboru.

url_contents (objekt) Model URL spracovaný podľa HTTP schémy.

url_analysis (objekt) Heuristiky definované na základe modelu URL.

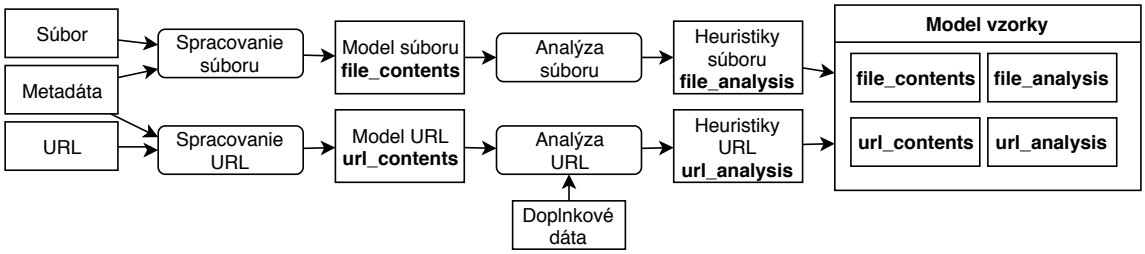
matched_rules (zoznam) Splnené pravidlá na základe modelu a heuristík definovaných nad súborom a URL.

classification (slovník) Výsledná klasifikácia phishing vzorky na základe splnených pravidiel.

4.4 Extrakcia charakteristík

Komponenta `Extraktor` má za úlohu vytvoriť zo vstupu model vzorky. Aplikáciou techniky hybridnej analýzy je zároveň analyzovaný vstupný súbor aj zdrojová URL a ich charakte-

ristiky a heuristiky sú zahrnuté v modely vzorky. Proces spracovania súboru do jednotného modelu vzorky je zobrazený na obrázku 4.3. Pri extrahovaní charakteristík sú využívané doplnkové dáta popísané v sekcií 4.4.3. Tieto dáta vo forme zoznamov ako napríklad biela a čierna listina URL sú použité v jednotlivých heuristikách pri vyhľadávaní časti modelu vzorky na tomto zozname.



Obr. 4.3: Extrakcia charakteristík súboru a URL z phishing vzorky.

4.4.1 Spracovanie a analýza súboru

Súbory potrebné pre analýzu sú získané z centrálneho úložiska súborov a načítané v nástroji `PhishCore`. Pre ďalšiu analýzu je ich obsah spracovaný do modelu obsahujúce charakteristiky súboru v objekte `file_contents` a heuristiky nad súborom v objekte `file_analysis`. Spracovanie vstupných súborov prebieha pomocou knižnice `beatifulsoup` [20], ktorá slúži na spracovanie a vyhľadávanie značiek v DOM strome HTML súborov.

Pomocou knižnice `beatifulsoup` je získaný objekt `BeatifulSoup`, ktorý obsahuje metódy pre navigáciu a vyhľadávanie v DOM Strome. S využitím rozhrania BeatifulSoup možno jednoducho vyhľadávať HTML značky používané pri phishing útoku a vytvoriť zoznamy značiek a ich atribútov. Tieto zoznamy sú uložené v modeli vzorky v objekte `file_contents`. Využitie knižnice pre načítanie zoznamu všetkých značiek `<form>` a zoznamu atribútov `action` týchto značiek ukazuje kód 4.4.1.

```
fbs_content = BeautifulSoup(fcontent, 'html.parser')
forms = fbs_content.find_all('form')
form_actions = list((form_element.get('action')) for form_element in forms)
```

Kód 4.4.1: Kód pre spracovanie vstupného súboru pomocou knižnice `beatifulsoup` do zoznamu značiek

Objekt `file_contents` je následne využitý k analýze súboru a vytvoreniu objektu `file_analysis`, ktorý obsahuje položky indikujúce možný phishing útok na základe štruktúry súboru a použitých značiek. Táto analýza sa zameriava na použité odkazy URL, lokálne odkazy, využitie `favicon` ikony a zdroje obrázkov a skriptov.

4.4.2 Spracovanie a analýza URL

Okrem vstupného súboru využíva nástroj `PhishCore` pripojenie databázy `Athos` z ktorej načítava zdrojové URL pre vstupné súbory. Využíva k tomu reláciu zdrojových URL a súborov prostredníctvom ktorej možno sledovať šírenie súboru na rôznych URL a zmeny obsahu na URL. Vytvorenie modelového objektu URL `url_contents` a objektu heuristík URL `url_analysis` ukazuje obrázok 4.3.

Po načítaní zdrojovej URL súboru z databázy dochádza k jej spracovaniu a uloženiu charakteristík URL do modelového objektu `url_contents`. Objekt `url_contents` obsahuje položky extrahované zo zdrojovej URL podľa HTTP schémy. Objekt `url_contents` je následne analyzovaný rôznymi technikami pre detekciu phishing útoku v URL. Tieto techniky obsahujú kontrolu na kľúčové slová, kontrolu separátorov, počtu podozrivých reťazcov a zložiek alebo podobnosť s populárnymi legitímnymi URL. Tieto informácie sú uložené v objekte modelu phishing vzorky `url_analysis` pre ďalšiu aplikáciu pravidiel a klasifikáciu vzoriek.

4.4.3 Doplnkové dáta pre analýzu

Okrem vzorky phishing útoku sú pri analýze používané ďalšie dáta slúžiace k rozšíreniu možností používaných heuristík a sú v nástroji `PhishCore` uložené prostredníctvom zoznamov. Tieto zoznamy zahŕňajú čiernu listinu (blacklist) domén umožňujúcich užívateľom nahrávať vlastný obsah, teda platformy pre blogy, hostingy a kloudové služby. Ďalší zoznam pozostáva z bielej listiny (whitelist) URL u ktorých nemožno predpokladať phishing útoky slúžiacich na odfiltrovanie falošných poplachov nástroja `PhishCore`. Ďalej je použitý zoznam podozrivých domén najvyššej úrovne (TLD), ktorý pozostáva z lacných domén často zneužívaných pre hosting phishing útokov. Tieto zoznamy sú uložené v textových súboroch uložených spolu s nástrojom a načítané pri spustení programu.

4.5 Aplikácia pravidiel

Nástroju `YARA` je možné na vstup predať okrem vstupného súboru a súboru s pravidlami aj súbor poskytujúci dáta pre modul. Po implementácií vlastného modulu tak možno do modulu `YARA` načítať model phishing vzorky serializovaný v `JSON` formáte [6]. Pomocou knižnice Jansson [12] pre spracovanie formátu `JSON` je možné naplniť štruktúry v module `YARA` položkami modelu vzorky. Štruktúry a funkcie definované v module `YARA` je potom možné využívať v podmienkach pre splnenie pravidiel. V pravidlách `YARA` tak možno využiť charakteristiky a heuristiky súboru a URL stránky vytvorené pomocou nástroja `PhishCore`.

Pomocou `YARA` pravidiel možno definovať vzory nad vzorkami phishing. V prípade splnenia pravidla je možné vstupu priradiť klasifikáciu podľa triedy pravidla. Vytvorené pravidlá sú definované v súboroch a rozdelené podľa ich vlastností. Toto umožňuje jednoduchú správu kolekcie pravidiel. Všetky tieto pravidlá sú predané nástroju `YARA` a výstupom sú splnené pravidlá nad danou vzorkou. Splnené pravidlá tak poskytujú informácie o známych vzoroch vo vzorke a umožňujú tak jednoduchšiu klasifikáciu.

Komponenta `Skener` aplikuje `YARA` pravidlá nad modelom phishing vzorky. Nástroj `YARA` je používaný pomocou balíčku `yara-python` [24] implementujúceho `Python` rozhranie nástroja `YARA`. Použitím balíčku možno využívať `YARA` v nástroji `PhishCore` a spojiť tak extrakciu charakteristík, aplikáciu pravidiel a klasifikáciu vzorky v jednom nástroji.

4.6 Klasifikácia vstupu

Komponenta `Klasifikátor` vytvára klasifikáciu modelu phishing vzorky. Klasifikácia je vytvorená na základe známych vzorov vo vzorke phishing získaných aplikovaním `YARA` pravidiel. Pravidlá používané pre klasifikáciu majú definovanú triedu klasifikácie pomocou meta premennej. Na základe pravidiel je vytvorená čiastočná klasifikácia súboru, URL a vzorky. Čiastočná klasifikácia vzorky je následne použitá v rozhodovacej matici pre zmenu klasifikácie súboru a URL. Výsledkom je klasifikácia súboru, URL a vzorky. Možné triedy klasifikácie predstavujú:

Čisté (Clean) vzorky, neobsahujúce phishing. Najčastejšie sú to stránky chybových HTTP kódov [18] 404 a 500, domovské stránky poskytovateľov hosting po odstránení stránok, vyčistené napadnuté stránky alebo iné stránky, ktoré sa dostali do zdroja phishing chybou poskytovateľov zdroja.

Neškodné (Harmless) vzorky obsahujúce techniky využívané v legitímnych stránkach avšak nie je možné potvrdiť neprítomnosť phishing. Trieda `Harmless` slúži pre neoverené pravidlá alebo pravidlá u ktorých dochádza ku falošným poplachom a nemohli byť zaradené do triedy `Clean`.

Neznáme (Unknown) vzorky v ktorých nebolo možné získať klasifikáciu, teda neboli splnené žiadne `YARA` pravidlá. V prípade náležitosti do tejto klasifikačnej triedy je vstup príliš malý na detekciu známych vzorov alebo potrebné pravidlá neboli definované.

Podozrivé (Suspicious) vzorky neobsahujúce phishing ale obsahujúce podozrivé značky v kóde alebo využívajúce podozrivé techniky pre presmerovanie, obfuskáciu zdrojových kódov alebo podozrivé umiestnenie skriptov a obrázkov webových stránok.

Phishing vzorky obsahujúce phishing na základe značiek použitých v zdrojovom kóde a definovaných známych vzorov phishing. Tieto pravidlá popisujú techniky pre napodobenie cieľovej stránky a zároveň nie sú splnené na cieľových stránkach phishing útokov.

Conflict vzorky, ktoré boli zároveň klasifikované ako čisté a phishing, teda došlo k chybe pri spracovaní. Výsledná klasifikácia `Conflict` poukazuje na nesprávne definované pravidlá a vzorky označené touto klasifikáciou je potrebné analyzovať ručne. Pre zmenu klasifikácie je potreba zmeniť existujúce klasifikačné pravidlá tak aby nedochádzalo ku ich splneniu na vzorkách mimo klasifikačnej triedy, prípadne vytvorenie nových pravidiel.

Čiastočná klasifikácia

Čiastočná klasifikácia vstupu je vytvorená na základe splnených pravidiel príslušných do jednotlivých tried klasifikácie. V prípade splnených pravidiel `Phishing` a `Clean` dochádza ku chybe definovaných pravidiel a je vrátená klasifikácia `Conflict`. Táto situácia nastáva v prípade kedy pravidlá vyvolajú falošných poplach, teda klasifikačné pravidlo je príliš obecné a je splnené aj nad vzorkami nepatriacimi do triedy klasifikácie.

Zmena klasifikácie tiež nastáva v prípade splnenia väčšieho množstva pravidiel klasifikačnej triedy `Suspicious`. V tomto prípade je pravidlám pridaná väčšia dôležitosť a trieda klasifikácie je zvýšená na triedu `Phishing`. Problematickým je správne nastavenie hranice pre zmenu klasifikácie, ktorá závisí na kvalite a počte pravidiel z triedy `Suspicious`.

V prípade že nedôjde ku klasifikácií `Conflict` a nebol splnený dostatočný počet pravidiel klasifikácie `Suspicious` je čiastočná klasifikácia pridelená podľa priority splnených pravidiel. Priorita pravidiel je určená podľa ich dôležitosti a to nasledovne: `Phishing`, `Clean`, `Suspicious`, `Harmless`. V prípade že nedošlo ku splneniu žiadnych pravidiel náležiacim týmto triedam, nie je možné čiastočnú klasifikáciu vytvoriť a je vrátená trieda `Unknown`.

Zmena klasifikácie

Čiastočná klasifikácia je vytvorená na základe splnených pravidiel daného typu pre súbor alebo URL. Pre určenie celkovej klasifikácie modelu je však braná do úvahy aj klasifikácia vzorky. Klasifikácia vzorky je vytvorená na základe pravidiel definovaných nad súborom i URL zároveň. Klasifikácia vzorky má vyššiu presnosť kvôli lepším vyjadrovacím schopnostiam pravidiel definovaných nad kombnáciou súboru a URL. Pre spresnenie čiastočnej klasifikácie je použitá rozhodovacia matica, ktorá mení čiastočný výsledok na základe klasifikácie vzorky.

		čiastočná klasifikácia			
		Clean	**Harmless**	**Suspicious**	**Phishing**
klasifikácia vzorky	**Clean**	Clean	Clean	Clean	Conflict
	Harmless	Clean	Harmless	Suspicious	Phishing
	Suspicious	Clean	Suspicious	Suspicious	Phishing
	Phishing	Conflict	Phishing	Phishing	Phishing

Tabuľka 4.1: Rozhodovacia matica pre určenie výslednej klasifikácie súboru na základe čiastočnej klasifikácie súboru a klasifikácie vzorky.

Rozhodovaciu maticu pre zmenu klasifikácie ukazuje tabuľka 4.1. Čiastočná klasifikácia súboru vytvorená v predošlom kroku a klasifikácia vzorky sú skombinované pre vytvorenie výslednej klasifikácie. V prípade kombinácie klasifikácie `Phishing` a `Clean` je detegovaná chybu programu a je vrátená trieda `Conflict`. Zmena klasifikácie dochádza ku pri kombinácií silných a slabých tried klasifikácie, teda kombinácia `Phishing/Suspicious` a kombinácia `Clean/Harmless`. V týchto prípadoch dochádza ku zmene na základe využitia hybridnej analýzy súboru a URL. V prípade ak čiastočná klasifikácia nie je vytvorená kvôli tomu že

žiadne pravidlá daného typu neboli splnené, záverečná klasifikácia je vytvorená na základe klasifikácie vzorky.

Na základe výslednej klasifikácie možno pristúpiť ku blokovaniu vzoriek technikami popísanými v sekcií 3.1. Vďaka rozdeleniu klasifikácie súboru a URL, možno rozlíšiť situácie kedy URL stránky je zjavne phishing avšak súbor je čistý. Táto situácia nastáva spolu s opačnou situácia kedy je čistá URL ale súbor obsahuje phishing. Detekcia na základe domény URL, teda nie je možné vytvoriť kvôli blokácií poskytovateľov hosting alebo cloud služieb.

Kapitola 5

Implementácia nástroja

Implementovaný nástroj `PhishCore` je rozdelený do niekoľkých komponent, kde každá komponenta implementuje jednu časť pipeline architektúry. Toto rozdelenie podporuje znovuvyužitelnosť a udržovatelnosť týchto komponent. Extrakciu charakteristík a tvorbu modelu vzorky implementuje komponenta extraktor. Aplikáciu definovaných pravidiel nad modelom vzorky implementuje komponenta skener. Klasifikáciu vzorky do jednotlivých tried spracováva komponenta klasifikátor.

5.1 Implementácia YARA modulu

Tak ako bolo popísané v sekcií 3.4 modul v nástroji `YARA` umožňuje v pravidlách používať definované funkcie a typy. Pre využitie modelu vzorky definovaného komponentou extraktor je nutná implementácia vlastného modulu *phish*. Tento modul po kompilácií umožňuje využitie ľubovoľných užívateľských funkcií v jazyku `YARA`.

5.1.1 Deklarácia a zostavenie

Pre použitie v pravidlách je potrebné tento modul deklarovať v zozname modulov `YARA`. Po deklarovaní je možné modul využívať v pravidlách a bez tejto deklarácie by samotný modul nebol viditeľný v nástroji `YARA`. Deklarácia modulu je vykonaná definíciou v súbore `libyara/modules/module_list` a je podmienená definíciou makra `MODULE(phish)`. Definíciu modulu ukazuje kód 5.1.1.

```
#ifdef PHISH_MODULE
MODULE(phish)
#endif
```

Kód 5.1.1: Definícia modulu v zozname modulov `YARA`.

Pre kompiláciu modulu do nástroja `YARA` je potrebné tento modul deklarovať v rámci súboru `Makefile.am`. Pridanie zdrojového súboru modulu je podmienené definíciou makra modulu `PHISH_MODULE`. Kód 5.1.2 ukazuje pridanie zdrojového kódu modulu pre kompiláciu.

```
#ifdef PHISH_MODULE
MODULES += modules/phish.c
#endif
```

Kód 5.1.2: Zaradenie modulu pre kompiláciu a zlinkovanie v nástroji `YARA`.

Makro modulu `PHISH_MODULE` pre podmienený preklad modulu je definované vstupnými argumentami nástroja `YARA`. Makro je definované na základe argumentu nástroja `-disable-phish`, ktorý vynucuje kompiláciu nástroja bez modulu `phish`. Argumenty nástroja sú špecifikované v skripte pre konfiguráciu `configure.ac`.

```
AC_ARG_ENABLE([phish],
  [AS_HELP_STRING([--disable-phish], [disable phish module])],
  [if test x$enableval = xno; then
    build_phish_module=false
  else
    build_phish_module=true
  fi],
  [ build_phish_module=true ])

AM_CONDITIONAL([PHISH_MODULE], [test x$build_phish_module = xtrue])
```

Kód 5.1.3: Definícia makra modulu pomocou vstupných argumentov.

5.1.2 Definícia štruktúr a funkcií

Implementovaný modul `phish` je v nástroji `YARA` definovaný v súbore `libyara/modules/phish.c`. V rámci tohoto súboru sú definované pomocné funkcie a implementácie funkcií rozhrania pre jazyk `YARA`, deklarácie štruktúr a funkcií pre jazyk `YARA` v module a funkcie nutné pre funkcionalitu modulu (`module_initialize`, `module_finalize`, `module_load`, `module_unload`).

Definície štruktúr a funkcií rozhrania modulu sú ohraničené príkazmi: `begin_declarations` a `end_declarations`. V rámci tejto sekcie sú nasledovne obsiahnuté deklarácie premenných, polí, štruktúr, funkcií a slovníkov. Príklad internej štruktúry obsahujúcej tieto deklarácie ukazuje kód 5.1.4.

```
begin_struct("div");
    declare_string_array("classes");
    declare_function("class", "s", "i", div_class);
    declare_function("class", "r", "i", div_class_regexp);
    declare_string_array("ids");
    declare_function("id", "s", "i", div_id);
    declare_function("id", "r", "i", div_id_regexp);
end_struct("div");
```

Kód 5.1.4: Štruktúra rozhrania pre značky `<div>` v rámci štruktúry `file_contents`.

Premenné modulu sú deklarované pomocou svojho mena a dátového typu, teda reťazce v rámci modulu sú definované príkazom: `declare_string("name")`, kde `"name"` značí názov deklarovanej premennej. Pre datové typy `integer` a `float` potom platia príkazy `declare_integer("name")` a `declare_float("name")`.

Polia umožňujú zoskupenie niekoľkých hodnôt rovnakého typu do jednej položky v rámci štruktúry. Deklarované sú pridaním suffixu `_array` ku príkazu ako napríklad: `declare_string_array("name")`, kde `"name"` značí názov pola.

Štruktúry sú deklarované pomocou príkazu `begin_struct("name")`, kde `"name"` značí názov štruktúry. Štruktúry umožňujú lepšie logické rozdelenie jednotlivých položiek a vyznačujú sa hlavne možnosťou rekurzívnych definícií. Na pole `ids` z kódu 5.1.4 tak možno pristupovať v `YARA` pravidle pomocou výrazu `file_contents.div.ids`. Toto rozdelenie naviac umožňuje definovať rovnako pomenované polia v rámci rôznych kontextov, napríklad polia pre atribúty `id` značiek `div` a `input`. Funkcie sú deklarované pomocou príkazu: `declare_function("name", "args", "ret", func_ref)`, kde `"name"` značí meno funkcie, `"args"` datové typy argumentov, `"ret"` návratový datový typ a `func_ref` definovanú funkciu pre obsluhu volania. Dostupné datové typy sú reťazce(`"s"`), celočíselná hodnota(`"i"`), hodnota s pohyblivou čiarkou(`"f"`) a regulárne výrazy (`"r"`).

Definícia obslužnej funkcie je pomocou príkazu `define_function("name")`, kde `name` je meno v deklarácií funkcie. Telo funkcie následne obsahuje obslužný kód volania a návratový typ je predávaný pomocou príkazov `return_string`, `return_integer` a `return_float` pre reťazec, celočíselnú hodnotu a hodnotu s pohyblivou čiarkou.

```
define_function(div_id)
{
    return_integer(
        string_match(
            parent(),
            "ids[%i]",
            string_argument(1)));
}
```

Kód 5.1.5: Funkcia pre obsluhu volania `file_contents.div.id("value")` s využitím pomocnej funkcie `string_match` pre vyhľadávanie reťazca v poli.

5.1.3 Naplnenie štruktúr z modelu vzorky

V prípade, že v jednom zo skenovaných pravidiel dôjde k použitiu modulu pomocou volania `import "phish"` modul sa načíta a zavolá sa funkcia `module_load`. V rámci tejto funkcie dochádza ku spracovaniu vstupných dát vo formáte `JSON` pomocou knižnice `Jansson` [12].

Výstupná správa vo formáte `JSON` z komponenty obsahujúca serializáciu modelu vzorky je predaná na vstup nástroja `YARA` pomocou prepínača `-x`. V tomto prípade dochádza ku spracovaniu vstupu zo súboru. Ďalšou možnosťou je využitie knižnice `yara-python`, kedy možno využívať nástroj `YARA` priamo v komponente skener. Vstupný model vzorky je predaný priamo vo formáte slovníku a vstup je spracovávaný rovnako.

Podobne ako to je v prípade štruktúr a funkcií, ktoré kopírujú štruktúru modelu vzorky v komponente extraktor, tak aj formát `JSON` správy odpovedá štruktúre modelu vzorky. V rámci spracovania teda vstup pozostáva z niekoľkých objektov obsahujúcich polia a položky. V module možno ku jednotlivým objektov v štruktúre priradiť ich dáta. V tomto prípade sa jedná o časť správy odpovedajúcej časti modelu, ktorý predstavuje daný objekt. Ukážku spracovania objektu ukazuje kód 5.1.6.

```
file_contents = get_object(module_object, "file_contents");
object_json = (json_t *)json_object_get(json, "file_contents");
file_contents->data = (void *)object_json;
```

Kód 5.1.6: Spracovanie objektu `file_contents` z načítanej vstupnej `JSON` správy a uloženie do objektu v rámci modulu.

Po uložení celého objektu dochádza k rozbaleniu reťazcov v objekte do jednotlivých premenných v module. Pomocou funkcie `json_unpack` je možné špecifikovať formát spracovávaného objektu, teda očakávané dátové typy kľúčov a hodnôt. Hodnoty rozbalených premenných sú uložené do špecifikovaných premenných, ktoré sú potom pomocou funkcie knižnice `set_string` uložené v danom objekte modulu. Tento proces je znázornený v kóde 5.1.7 a je vykonávaný pre všetky jednoduché datové typy modelu vzorky.

```
if (json_unpack(object_json, "{s:s, s:s}",
    "favicon", &favicon, "title", &title) == 0)
{
    set_string(favicon, module_object, "file_contents.favicon");
    set_string(title, module_object, "file_contents.title");
}
```

Kód 5.1.7: Uloženie reťazcov z objektu `file_contents` do premenných v module.

Model vzorky okrem základných dátových typov obsahuje aj polia jednotlivých atribútov elementov. Tieto polia nie sú rozbalené ale načítané v cykle po jednotlivých hodnotách. Pre prechod položkami polí vo vstupnom reporte je použitá funkcia `json_array_foreach` a v rámci cyklu sa využíva funkcie knižnice `set_string`, v ktorej je naviac použitý index hodnoty v rámci pola. Táto funkcionalita je exportovaná do pomocnej funkcie `set_module_struct`. Časť kódu funkcie `set_module_struct` pre spracovanie položiek v cykle je znázornená v kóde 5.1.8

```
array_json = (json_t *)json_object_get(object_json, array_name);
json_array_foreach(array_json, index, value)
{
    set_string(json_string_value(value), module_object, struct_item_name, index);
}
```

Kód 5.1.8: Výsek kódu pre nastavenie položiek polí zo vstupného reportu do polí v module.

5.1.4 Zhrnutie

Po vykonaní kódu vo funkcií `module_load` dochádza ku spracovaniu vstupnej správy a nastavení jednotlivých hodnôt do položiek v rámci modulu. Tieto položky sú definované v sekcií deklarácií, ktorá určuje dátové typy a mená týchto položiek a definuje funkcie pre zaobchádzanie s nimi. Po preložení vlastného modulu sú definované funkcie a štruktúry prístupne pri aplikácií pravidiel. Do podmienok pravidiel možno pomocou definovaných funkcií v module zaviesť ďalšie rozhranie a umožniť tak vyhľadávanie reťazcov a aplikáciu regulárnych výrazov nad položkami modulu. Pomocou `YARA` pravidiel tak možno detegovať známe vzory vo vstupných súboroch a využiť tieto pravidlá pre klasifikáciu.

5.2 Komponenta extraktor

Komponenta extraktor je zodpovedná za spracovanie vstupnej phishing vzorky do modelu vzorky. Trieda `SampleModel` reprezentujúca model vzorky obsahuje triedy pre reprezentáciu charakteristík súboru `FileContents`, heuristík súboru `FileAnalysis`, charakteristík URL `UrlContents` a heuristík URL `UrlAnalysis`. Tieto triedy sú vytvárané triedou pre spracovanie chrakteristík a vytvorenie modelu vzorky `FeatureExtraction`, triedou pre spracovanie súboru `FileAnalysisService` a triedou pre spracovanie URL `UrlAnalysisService`.

FeatureExtraction

Trieda `FeatureExtraction` má za úlohu získanie obsahu súboru a využitie ostatných služieb pre vytvorenie modelu vzorky. Pre správne spracovanie súboru je potrebné rozlíšiť jeho kódovanie a správne načítaný obsah odoslať na extrakciu charakteristík.

```
def extract_features(self, content, file_sha):
    # vytvorenie modelu suboru na zaklade jeho obsahu
    fc = self.fa_service.create_model(content, file_sha)

    # trieda heuristik na zaklade modelu suboru
    fa = self.fa_service.analyse_contents(fc)

    # model URL na zaklade schemy URI
    uc = self.ua_service.create_model(fa.source_url)

    # trieda heuristik URL
    ua = self.ua_service.analyse_contents(uc)

    return sample_model.SampleModel(fc, fa, fa.source_url, uc, ua)
```

Kód 5.2.1: Kód pre vytvorenie objektov modelov a heuristík súborov a URL. Na základe vytvorených objektov a zdrojovej url je vytvorený model vzorky.

Rozlíšenie kódovania a dekódovanie obsahu je implementované vo funkcií `get_contents` pomocou knižnice `python-magic` [8]. Implementácia metódy `extract_features` pre vytvo-

renie modelu vzorky je zobrazená na kóde 5.2.1. Pri vytvorení modelu vzorky sú využité triedy a služby implementujúce spracovanie jednotlivých častí.

Táto trieda využíva všetky ostatné triedy komponenty extraktor a poskytujé jednotné rozhranie. Rozhranie pre vytvorenie modelu vzorky je vhodné pri využití nástroja `PhishCore` ako knižnicu v iných nástrojoch pre spracovanie malware. Rozdelenie spracovania modelu do jednotlivých tried umožňuje ich zmenu bez dopadu na zvyšok spracovania vzorky.

SampleModel

Trieda `SampleModel` reprezentuje model vzorky definovaný v sekcií 4.3. Trieda obsahuje okrem definovaných položiek modelu vzorky získaných z jednotlivých komponent položky slúžiace pre prácu s modelom pri manuálnej analýze. Pridanými položkami sú `source_url`, `file_path`, `file_sha256` a `url_sha256`. Tieto položky slúžia ku identifikácií vstupnej phishing vzorky na lokálnom úložisku, čo umožňuje užívateľovi rýchlejšie vyhľadávanie vzoriek pri kontrole klasifikácie a návrhu nových pravidiel.

Po skončení analýzy vzorky je tento model serializovaný do výstupnej správy pre analytika. Táto správa slúži ako zdroj ďalších informácií počas manuálnej analýzy phishing vzoriek. Výstupná správa obsahuje položky modelu a splnené pravidlá potrebné ku identifikácií nových vzorov vo phishing vzorkách a návrhu pravidiel.

fc (objekt) Instancia triedy `FileContents` reprezentujúca model súboru.

fa (objekt) Instancia triedy `FileAnalysis` obsahujúca heuristiky súboru.

source_url (reťazec) Zdrojová URL súboru získaná z databáze `Athos`.

uc (objekt) Instancia triedy `UrlContents` reprezentujúca model súboru.

ua (objekt) Instancia triedy `UrlAnalysis` obsahujúca heuristiky súboru.

matched_rules (zoznam) Zoznam identifikátorov splnených `YARA` pravidiel.

classification (slovník) Výsledná klasifikácia zdrojového súboru, URL a vzorky.

file_sha256 (reťazec) Hash SHA-256 obsahu súboru pre identifikáciu súboru v lokálnom úložisku a databázi `Athos`.

url_sha256 (reťazec) Hash SHA-256 zdrojovej URL pre identifikáciu URL v databázi `Athos`.

file_path (reťazec) Cesta ku zdrojovému súboru v lokálnom úložisku.

FileContents

Trieda `FileContents` reprezentuje objekt modelu súboru `file_contents`, ktorý obsahuje položky získané prostredníctvom spracovania zdrojového kódu v súbore. Použité položky

sú definované na základe `HTML` značiek a ich atribútov. Obsah značky získaný pomocou knižnice `BeatifulSoup4` je následne spracovaný do polí podľa atribútov značky. Tieto polia obsahujú hodnoty atribútu danej značky v poradí v akom sa nachádzajú v zdrojovom súbore. Na základe pola atribútov je následne spočítaný hash, ktorý môže byť spolu s polom použitý pri definícií `YARA` pravidiel nad modelom phishing vzorky. Použité atribúty pre vytvorenie polí obsiahnutých v modeli súboru ukazuje tabuľka 5.1.

Okrem polí definovaných na základe atribútov značiek obsahuje model aj položky predstavujúce obsah značiek. Tieto položky sú definované pre `HTML` značky bez atribútov, ktorých obsah je dôležitý pre prevedenie phishing útoku. Takéto polia modelu sú `favicon` a `title`.

značka	*atribúty*
a	class, href, id, style, text
button	class, id, onclick
comment	text
div	class, id
favicon	
form	action,class, method, name
h1	class, text
h2	class, text
img	alt, src
input	id, name, placeholder, src, type, value
link	href, rel, type
meta	name, property, text
script	src, text, type
span	class
style	text, type
title	

Tabuľka 5.1: Tabuľka značiek a atribútov použitých v modeli súboru `file_contents`.

FileAnalysis

Trieda `FileAnalysis` obsahuje heuristiky definované na základe modelu súboru `file_contents`. Heuristiky zakladajú na hodnotách získaných pomocou aplikácie regulárnych výrazov na časti modelu, prípadne poliach získaných z phishing databázy `Athos` popísanej v sekcií 4.1.

company_name (reťazec) Meno cieľovej spoločnosti phishing útoku, na základe splneného regulárneho výrazu na zdrojovej URL súboru. Regulárne výrazy pre spoločnosti sú definované v doplnkových dátach pre analýzu.

company_regex (reťazec) Regulárny výraz, ktorý bol splnený pre cieľovú spoločnosť.

content_type (reťazec) Typ obsahu súboru získaný z databázy `Athos`. Typ obsahu umožňuje priradiť súbor do niekoľkých tried na základe splnených regulárnych výrazov nad

obsahom súboru. Triedy súborov na základe poľa `content_type` sú napríklad chybové stránky (Error), súbory pochádzajúce z hostingu zdarma (FreeDomain) alebo zašifrované zdrojové kódy stránok (Crypted).

detected_by (reťazec) Mená detekcií vytvorených nad súborom získané z databázy `Athos`. Pole obsahuje mená všetkých detekcií definovaných pre klienta `Avast` ktoré sú splnené nad súborom.

favicon_path_type (reťazec) Poľe definujúce typ cesty ikony `favicon` ako absolútny (Absolute) alebo relatívny (Relative).

phish_group (reťazec) Cieľová skupina útoku na základe regulárnych výrazov definovaných na dobsahom súboru.

Trieda FileAnalysis obsahuje ďalšie polia nezahrnuté v modeli vzorky pre analýzu pomocou `YARA` pravidiel. Tieto polia sú využité pre manuálnu analýzu a môžu byť využité pri tvorbe ďalších heuristík použitých v modeli.

FileAnalysisService

Trieda `FileAnalysisService` má za úlohu spracovanie obsahu súboru do objektu modelu súboru `file_contents` a objektu heuristík súboru `file_analysis`. Táto funkcionalita je implementovaná vo funkciách `create_model(content, file_sha)` a `analyse_contents(fc)`. Pre naplnenie polí objektu obsahuje pomocné funkcie `get_source_urls(fa, fc)` pre získanie zdrojových URL z databázy `Athos` a `get_target_company(url_text)` pre získanie cieľovej spoločnosti phishing útoku na základe URL.

```
at.parse_url("http://user:pass@www.subdomain.domain.com:80/dir/index.html"
    ";parameters?id=1101&userid=1023#auth")
{'url': 'http://user:pass@www.subdomain.domain.com:80/dir/index.html'
    ';parameters?id=1101&userid=1023#auth',
 'scheme': 'http',
 'netloc': 'user:pass@www.subdomain.domain.com:80',
 'path': '/dir/index.html',
 'params': 'parameters',
 'query': 'id=1101&userid=1023',
 'fragment': 'auth',
 'username': 'user',
 'password': 'pass',
 'hostname': 'www.subdomain.domain.com',
 'port': 80,
 'is_ipv4': False,
 'domain_name': 'domain',
 'subdomains': 'www.subdomain',
 'TLD': 'com'}
```

Kód 5.2.2: Využitie knižnice `anttools` pre spracovanie URL.

UrlContents

Trieda `UrlContents` obsahuje položky definované podľa URI schémy [22]. Tieto položky sú spracované zo zdrojovej URL pomocou knižnice `anttools`. Výsledkom sú polia: url, scheme, username, password, hostname, port, params, query, fragment. Okrem týchto položiek sú pridané polia `domain_name` obsahujúce doménu druhého rádu, `subdomains` obsahujúce všetky ďalšie subdomény, obsahujúce doménu najvyššieho rádu `TLD(Top Level Domain)` a pole `is_ipv4` indikujúce či daná URL obsahuje ipv4 adresu. Príklad spracovania zdrojovej URL a výsledné polia obsiahnuté v modeli `url_contents` ukazuje kód 5.2.2.

UrlAnalysis

Trieda `UrlAnalysis` obsahuje funkcie definované nad modelom URL. Pre rozlíšenie phishing v URL je potrebné definovať známe vzory phishing útokov. Definícia známych vzorov je možná pomocou redukcie častí URL na zoznam kľúčových slov nájdených v časti URL ako je to u polí `domain_key_words`, `path_key_words` a `url_sequences`. Známe kľúčové slová a sekvencie sú definované v doplnkových dátach pre analýzu a ovplyvňujú hodnotu hashe SHA-256 definovanej nad týmito polami. Hashe polí sú výhodné pre automatickú klasifikácie vzoriek, teda použitie v `YARA` pravidlách. Zoznamy kľúčových slov sú dôležité pre manuálnu analýzu a potrebné na pregenerovanie hodnôt pri zmene známych kľúčových slov.

Pre rozšírenie možností detekcie sú pridané funkcie indikujúce prítomnosť časti modelu na zozname, ktorý je definovaný v doplnkových dátach pre analýzu. Tieto dáta sú zozbierané pomocou ručnej analýzy phishing vzoriek a pridané na zoznam,čo umožňuje následne sledovať známe hodnoty modelu. Zhoda časti modelu a položiek na predom definovaných dátach pre analýzu je využitá poliach `suspicious_source_tld`, `not_valid_domain` a `url_type`

suspicious_source_tld (bool) Premenná indikujúca prítomnosť TLD zdrojovej URL na zozname podozrivých TLD definovanom v doplnkových dátach pre analýzu.

not_valid_domain (bool) Indikátor chyby pri spracovaní zdrojovej URL do modelu URL.

url_type (reťazec) Typ URL určený prítomnosťou URL na zoznamoch definovaných v doplnkových dátach pre analýzu. Tieto zoznamy obsahujú položky poskytovateľov služieb hosting (Hosting), skracovačov URL (Short), online nástrojov pre tvorbu formulárov a ich zdieľanie (Form), bielu listinu stránok (White) alebo žiadnu zhodu na jednom zo zoznamov (Nothing).

domain_key_words (zoznam reťazcov) Zoznam kľúčových slov nájdených v doméne URL. Zdrojová URL je rozdelená na doménu a cestu ,v ktorých sú následne vyhľadávané kľúčové slová. Daná časť URL je rozdelená podľa definovaných oddeľovačov a rozdelená tak na jednotlivé časti - tokeny. Tieto tokeny sú následne vyhľadávané na zozname kľúčových slov a v prípade zhody zaradené do zoznamu kľúčových slov pre danú časť URL. Postupnosť nájdených kľúčových slov popisuje štruktúru URL a môže byť využitá pre rozpoznanie známych vzorov.

domain_key_words_hashed (reťazec) SHA-256 hash zoznamu reťazcu kľúčových slov v doméne URL.

path_key_words (zoznam reťazcov) Zoznam kľúčových slov nájdených v ceste URL obdobne ako pri doméne.

path_key_words_hashed (reťazec) SHA-256 hash zoznamu reťazcu kľúčových slov v ceste URL.

url_sequences (zoznam reťazcov) Zoznam známych sekvencií nájdených v phishing URL. Zdrojová URL je rozdelená na tokeny pomocou rozdeľovačov a tieto tokeny sú vyhľadávané na zvláštnom zozname známych seknvencií.

url_sequences_hashed (reťazec) SHA-256 hash zoznamu známych sekvencií.

UrlAnalysisService

Trieda `UrlAnalysisService` má za úlohu spracovanie URL a vytvorenie objektu modelu URL `url_contents` a objektu heuristík `url_analysis`. Táto funkcionalita je implementovaná vo funkciách `create_model(source_url)` a `analyse_contents(url_contents)`. Trieda obsahuje ďalšie pomocné funkcie pre vytvorenie heuristík nad modelom URL vo funkciách `tokenize_url(txt, separators)` pre vytvorenie zoznamu tokenov z URL a `analyze_keywords(uc)` pre vytvorenie zoznamu kľúčových slov nájdených v častiach URL. V triede sú implementované aj pomocné funkcie `get_hash(inp)` pre vytvorenie hashí položiek objektov a `get_source_url_type(source_url)` pre určenie typu URL na základe zhody na zoznamoch doplnkových dát pre analýzu.

5.3 Komponenta skener

Komponenta skener je zodpovedná za aplikáciu `YARA` pravidiel. Na základe modelu vzorky je prostredníctvom modulu v nástroji naplnená štruktúra obsahujúca premenné a funkcie odpovedajúce modelu. Definované pravidlá využívajú túto štruktúru v podmienke pre ich splnenie.

5.3.1 Definícia YARA pravidiel

Pre správnu funkcionalitu komponenty skener je potrebné definovať `YARA` pravidlá, ktoré sú nad modelom aplikované. `YARA` pravidlá sú definované v troch sadách pravidiel v moduli nástroja `rules` podľa časti vzorky ,na ktorú sú cielené. Pravidlá sú rozdelené podľa časti vzorky na:

súbor Pravidlá uložené v sade pravidiel `file_rules` využívajúce charakteristiky a heuristiky súboru v objektoch `file_contents` a `file_analysis`.

URL Pravidlá uložené v sade pravidiel `url_rules` využívajúce charakteristiky a heuristiky URL v objektoch `url_contents` a `url_analysis`.

vzorka Pravidlá uložené v sade pravidiel `sample_rules` využívajúce kombináciu súboru a URL, teda pokrývajúce celú vzorku phishing útoku.

Sada pravidiel uložená v nástroji obsahuje pravidlá definované s využitím implementovaného modulu `phish` popísaného v sekcií 5.1. V pravidlách je použitá položka metadát `classification`, ktorá značí triedu klasifikácie pravidla. Na základe vplyvu klasifikácie pravidla na výslednú klasifikáciu je potom prípustná miera falošných poplachov, teda phishing vzoriek mimo triedy klasifikácie na ktorých je pravidlo splnené. Pre triedy klasifikácie `Clean` a `Phishing` je teda potrebné aby nedochádzalo k žiadnym falošným poplachom.

```
rule WebhostappChasePhishing
{
    meta:
        author = "Marek Beno, Avast"
        classification = "Phishing"
    condition:
        phish.file_contents.title contains "Chase" and
        phish.file_contents.img.alt("www.000webhost.com") and
        Redirect
}
```

Kód 5.3.1: Klasifikačné YARA pravidlo definované na základe súboru vzorky. Phishing útok využíva klúčové slovo Chase v titulke stránky avšak zároveň je webová stránka nasadená na stránkach poskytovateľa hostingu zdarma `000webhost` a využíva techniky pre presmerovanie stránky definované v pravidle `Redirect`.

`YARA` pravidlo ukázané v kóde 5.3.1 ukazuje príklad klasifikačného pravidla s triedou `Phishing`. Pravidlo `WebhostappChasePhishing` však spolieha len na nepriame indikátory phishing útoku a neobsahuje žiadne zo značiek použitých pre získanie prístupových údajov ani sekvenciu jednoznačne identifikovateľnú v danej verzií phishing útoku. Pre zvýšenie bezpečnosti tohoto pravidla možno pridať do podmienky pravidla element špecifikujúci vstupné pole pre heslo a zamedziť tak splneniu pravidla nad stránkami neobsahujúcimi phishing útok ako napríklad blogy a bezpečnostné stránky informujúce o útokoch. Pre zamedzenie falošných poplachov možno pridať do podmienky položku ukázanú v kóde 5.3.2.

```
phish.file_contents.input.name("password")
```

Kód 5.3.2: Položka podmienky popisujúca vstupné pole s názvom `password` v zdrojovom súbore phishing vzorky.

Návrh nových pravidiel je potrebný v prípade že nad vstupnou vzorkou neboli splnené žiadne pravidlá a v prípade, že došlo k nesprávnemu určeniu klasifikácie. Pri nesprávnom určení klasifikácie je potrebné zmeniť existujúce podmienky, prípadne pridať nové položky. Výber nových položiek pri vytváraní pravidla na základe phishing modelu je založený na

rozdielnosti phishing vzorky od cieľovej stránky aby nedošlo k falošným poplachom a zároveň umožňuje identifikáciu phishing útoku.

5.3.2 Aplikácia pravidiel

Aplikácia pravidiel prebieha v triede `ScanningService`, ktorá využíva definované pravidlá a model vzorky. Pravidlá sú aplikované po častiach podľa typu, kvôli ďalšej klasifikácií modelu po častiach. Výsledkom je slovník obsahujúci pravidlá, kde kľúčom je typ pravidiel a hodnotou slovník obsahujúci splnené pravidlá podľa triedy klasifikácie.

Aplikácia pravidiel je prevedná v metóde `get_matches(rules, sample)`, kde rules sú pravidlá daného typu a sample je model vzorky phishing. Model vzorky získaný pomocou komponenty extraktor je serializovaný do `JSON` formátu a použitý ako dáta modulu `phish`. Pre aplikáciu `YARA` pravidiel je použitý nástroj `YARA` za pomoci `Python` knižnice `yara-python`. Táto knižnica obsahuje implementáciu nástroja `YARA` doplnenú o implementovaný modul `phish`. Vďaka tejto knižnici možno splnené pravidlá ďalej spracovávať a previesť nad nimi ďalšiu klasifikáciu.

Implementáciu aplikácie pravidiel a spracovania výsledkov ukazuje kód 5.3.3. Pre aplikáciu pravidiel pomocou metódy `yara.Rules.match` je použitý obsah skenovaného súboru `fcontent` uložený v modeli vzorky a serializovaný model `phish_data` ako dáta modulu. Výsledkom je zoznam splnených pravidiel, ktoré sú uložené do slovníku na výstupe podľa triedy klasifikácie.

```
# model vzorky vo formate JSON pre data modulu
phish_data = {
    'phish': json.dumps(sample, cls=ComplexEncoder).encode('utf-8')
}
# aplikacia pravidiel
yaraMatches = rules.match(data=sample.fc.fcontent, modules_data=phish_data)
# slovnik vysledkov podla klasifikacie
for match in yaraMatches:
    rule_classification = match.meta["classification"]
    match_results[PhishClass.from_string(rule_classification)].append(match.rule)
```

Kód 5.3.3: Kód pre aplikáciu `YARA` pravidiel.

5.4 Komponenta klasifikátor

V rámci komponenty klasifikátor je vytváraná klasifikácia modelu vzorky. Klasifikácia vzorky phishing určuje jednu z klasifikačných tried popísaných v sekcií 4.6.

Klasifikácia modelu

Klasifikácia modelu phishing vzorky je implementovaná v triede *PrevalentClassificationService*. Klasifikácia modelu vzorky je vytvorená na základe splnených pravidiel pomocou

metódy `get_classification(matched_rules)`. Splnené pravidlá sú použité pre vytvorenie čiastočnej klasifikácie súboru, URL a klasifikácie vzorky pomocou metódy `get_partial_classification(match_dict)`.

Metóda `get_combined_classification(partial, sample)` implementuje zmenu výslednej klasifikácie na základe klasifikácie vzorky a čiastočnej klasifikácie súboru alebo URL. Celková klasifikácia je vytvorená použitím rozhodovacej matice, ktorá je indexovaná čiastočnou klasifikáciou a klasifikáciou vzorky a výsledkom je celková klasifikácia. Celková klasifikácia súboru a vzorky je zmenená v prípade stretu klasifikačných tried `Clean` a `Phishing`, kedy dochádza k zmene na klasifikačnú triedu `Conflict`.

Kapitola 6

Testovanie nástroja

Táto kapitola popisuje testovanie výsledkov nástroja na pripravených testovacích dátach a následne popisuje experimenty vykonané na testovacích dátach. Testovacie dáta sú vytvorené na základe zoznamov populárnych URL a zdrojov phishing útokov spoločnosti Avast. Tieto dáta pozostávajú z adries URL webových stránok a ich obsahu. Testovacie dáta sú uložené v databázy `Athos`, ktorá je zároveň využitá aj pre uloženie výslednej klasifikácie. Nad testovacími dátami sú vykonané experimenty s cieľom zistiť mieru falošných poplachov a úspešnosti nástroja pre jednotlivé triedy klasifikácie.

6.1 Testovacie dáta

Testovacie dáta použité pri experimentoch pozostávajú z nasledujúcich sád dát:

1. Sada klasifikačných `YARA` pravidiel pre klasifikáciu vzoriek počas experimentov.
2. Testovacia sada čistých vzoriek pozostáva zo vzoriek neobsahujúcich phishing útoky pre určenie miery falošných poplachov.
3. Testovacia sada vzoriek zo zdroja PhishTank obsahuje vzorky reálnych dát používaných pri analýze phishing v prostredí Avast.
4. Testovacia sada manuálne klasifikovaných vzoriek malware analytikmi `Avast`.

6.1.1 Sada klasifikačných YARA pravidiel

Pre účely testovania a experimentácie s nástrojom bola vytvorená sada `YARA` pravidiel pre klasifikáciu phishing vzoriek. Táto sada pravidiel bola vytvorená na základe vzoriek phishing získaných z databáze `Athos`.

Dátová sada obsahuje pravidlá zakladajúce na charakteristikách súborov, URL a vzorky v odpovedajúcich súboroch `file_rules`, `url_rules` a `sample_rules`. Počet pravidiel definovaných v týchto súboroch ukazuje tabuľka 6.1. Pravidlá klasifikácie `Phishing` a `Clean`

		klasifikácia vzorky			
	počet `YARA` pravidiel	**Phishing**	**Suspicious**	**Harmless**	**Clean**
model	**súbor**	16	23	1	5
	URL	4	9	3	1
	vzorka	5	3	0	0

Tabuľka 6.1: Počet `YARA` pravidiel definovaných v testovacej sade podľa časti vzorky a klasifikácie.

majú najväčší vplyv na výslednú klasifikáciu vzorky avšak pravidlá `Clean` sú zložité na návrh preto je ich počet v testovacej sade nízky. Najviac pravidiel je klasifikácie `Suspicious` kde sú definované pravidlá ukazujúce na podozrivé techniky, ktoré však neznačia prítomnosť phishing útoku. Pravidlá `Suspicious` a `Phishing` tak tvoria najvýznamnejšiu rolu pri tvorbe klasifikácie vzorky.

Príklad klasifikačného pravidla vzorky použitého v datovej sade ukazuje kód 6.1.1. Toto pravidlo využíva techniky hybridnej analýzy a kombinuje dva samostatne podozrivé vzory do jedného. Pravidlo klasifikuje kombináciu kľúčových slov v zdrojovej ceste obrázku a `WordPress` stránky ako phishing.

```
rule WordpressPhishingImgSrc
{
  meta:
    author = "Marek Beno, Avast"
    classification = "Phishing"
  condition:
    phish.file_contents.img.src("adobe")
        or phish.file_contents.img.src("chase")
        or phish.file_contents.img.src("PayPal") and
        // cesta URL obsahuje zname zlozky wordpress blogu
    rule_wordpress
}
```

Kód 6.1.1: Klasifikačné YARA pravidlo využívajúce model súboru a klasifikačné pravidlo `rule_wordpress` definované nad cestou URL.

6.1.2 Testovacia sada čistých vzoriek

Testovacia sada čistých vzoriek pozostáva zo vzoriek, ktoré neobsahujú phishing útok a zároveň pozostáva zo vzoriek s vysokým dopadom v prípade falošného poplachu - klasifikácie ako Phishing. Nesprávna klasifikácia `Phishing` veľmi populárnej stránky má za následok veľký počet falošných poplachov a následnú stratu dôveryhodnosti nástroja u užívateľa. Táto sada pozostáva z najpopulárnejších stránok Internetu u ktorých by falošných poplach mal veľký dopad na užívateľov. Táto sada bola vytvorená na základe zoznamov najpopulárnejších stránok služieb `Alexa` [1] a `Majestic million` [15].

V prípade služby `Alexa` bol využitý archívny zoznam `top100k` kedže aktuálne zoznamy nie sú verejne dostupné. Aktuálny zoznam najpopulárnejších stránok bol získaný zo služby `Majestic Million` z ktorého bolo vybraných sto tisíc najlepších stránok. Tieto dva zoznamy boli skombinované a ich prienik zaradený do testovacej sady. Takýmto spôsobom boli získané dlhodobo populárne stránky.

Takto získaná testovacia sada bola spracovaná s využitým nástroja `phishing_downloader` a vložená do repliky databázy `Athos` obdobne ako to je u reálnych dát. Pre podmienky testovania nástroja boli následne z tohoto zoznamu odobrané vzorky ktoré sa nepodarilo úspešne spracovať z dôvodu neaktívnosti stránky a uložiť do centrálneho úložiska súborov v Avast. Výsledkom je sada 34443 najpopulárnejších webových stránok pripravená na spracovanie s cieľom kontroly nástroja voči falošným poplachom.

6.1.3 Testovacia sada vzoriek phishing zo zdroja PhishTank

Testovacia sada bola vytvorená na základe zdroja phishing vzoriek PhishTank. Testovacia sada celkom 2938 bola vytvorená zo vzoriek Phishing pochádzajúcich zo zdroja `PhishTank` po dobu dvoch týždňov. Tieto vzorky boli spracované nástrojom `phishing_downloader` a uložené v databázy `Athos`. Táto testovacia sada predstavuje reálne dáta používané pri analýze phishing a obsahuje vzorky všetkých tried klasifikácie. Keďže `PhishTank` zakladá na kolaboratívnom hodnotení súborov v testovacej sade sa nachádzajú aj vzorky neobsahujúce phishing kvôli nekorektnému hlasovaniu užívateľov alebo chybe zdrojov tretích strán, ktoré `PhishTank` využíva.

	cieľová spoločnosť						
	nezistená	**Google**	**PayPal**	**Microsoft**	**Itau**	**Ecoin**	**Iné**
počet	1266	1173	1111	263	40	37	314

Tabuľka 6.2: Tabuľka ukazujúca počet vzoriek cielených na danú cieľovú spoločnosť na dátovej sade vzoriek zo zdroja `PhishTank`.

Počas spracovania vzoriek nástrojom `phishing_downloader` a ich uloženia do databázy `Athos` bola u vzoriek vyhodnotená cieľová spoločnosť útoku `phish_group`. Cieľové spoločnosti phishing útoku vzoriek datovej sady zobrazuje tabuľka 6.2. Z cieľových stránok je celkom 1266 útokov cielených na nezistenú spoločnosť. Medzi najpopulárnejšími spoločnosťami phishing útokov dominujú spoločnosti `Google` a `PayPal` nasledované spoločnosťou `Microsoft`. Medzi najčastejšie ciele phishing útokov patrí ako jediná banka `Itau` nasledovaná skupinou kryptomien.

Táto testovacia sada reprezentuje sadu reálnych dát spracovávaných pri analýze phishing útokov. Vybrané vzorky však pochádzajú z jediného zdroja, ktorý využíva komunitu užívateľov pre nahranie a klasifikáciu vzoriek. Tento aspekt prispieva k rozšíreniu spoločností `Google` a `PayPal` na testovacej sade. Výsledkom je testovacia sada vzoriek pre porovnanie detekcie vzoriek nástroja `PhishCore` s ostatnými nástrojmi.

6.1.4 Testovacia sada manuálne klasifikovaných vzoriek

Testovacia sada bola vytvorená na základe manuálne klasifikovaných vzoriek phishing z databázy `Athos`. Vzorky phishing pochádzajúce z viacerých zdrojov boli manuálne klasifikované analytikmi malware. Výsledná klasifikácia je uložená v databázy `Athos`.

Z klasifikovaných vzoriek bola vybraná sada 9943 vzoriek, ktorá bola ručne klasifikovaná ako `malware`. Táto klasifikácia odpovedá klasifikácií `Phishing` nástroja `PhishCore` a je teda vhodná pre testovanie úspešnosti nástroja. Nad touto testovacou sadou bola prevedená analýza cieľovej spoločnosti phishing útoku. Výsledky tejto analýzy zobrazuje tabuľka 6.3.

	cieľová spoločnosť						
	PayPal	**nezistená**	**Facebook**	**Email**	**Microsoft**	**Dropbox**	**Iné**
počet	3189	1580	1515	969	543	497	1336

Tabuľka 6.3: Tabuľka ukazujúca počet vzoriek cielených na danú cieľovú spoločnosť na dátovej sade manuálne klasifikovaných vzoriek. Medzi populárne spoločností patrí aj cieľová skupine Email, ktorá obsahuje phishing útoky na poskytovateľov emailových služieb s účelom odcudzenia emailových účtov.

Rozdelenie cieľových spoločností v tejto testovacej sade odpovedá populárnym cieľovým spoločnostiam phishing útokov. Zahrnutím rôznych zdrojov phishing vzoriek tak bola zaručená aj rôznorodosť sofistikovanosti útokov. Výsledkom je testovacia sada vzoriek cielených na rozličné spoločnosti a pochádzajúcich z rozličných zdrojov. Táto sada je cieľená pre porovnanie automatickej klasifikácie nástrojom a klasifikáciou vytvorenou pri manuálnej analýze analytikom.

6.2 Experimenty

Táto sekcia popisuje experimenty s nástrojom `PhishCore` a testovacími sadami popísanými v predošlej sekcií. Vykonané boli nasledovné experimenty:

- Aplikácia nástroja na testovacej sade čistých vzoriek má za úlohu zistiť mieru falošných poplachov na testovacej sade čistých vzoriek.
- Aplikácia nástroja na testovacej sade vzoriek phishing ukazuje výsledky nástroja na reálnych dátach v praxi a porovnanie nástroja.
- Aplikácia nástroja na testovacej sade s manuálnou klasifikáciou porovnáva automatickú klasifikáciu phishing vzoriek a klasifikáciu analytikom počas manuálnej analýzy.
- Výkonnostné testovanie nástroja sa zameriava na dobu potrebnú na analýzu vzoriek.

6.2.1 Čisté vzorky

V tomto experimente bola využitá testovacia sada čistých vzoriek popísaná v sekcií 6.1.2 a sada YARA pravidiel popísaná v sekcií 6.1.1. Nástroj `PhishCore` bol spustený nad testova-

		klasifikácia vzorky					
		Phishing	**Suspicious**	**Unknown**	**Harmless**	**Clean**	**NULL**
model	**súbor**	1	25896	6174	0	2229	143
	URL	0	235	9705	0	23747	756

Tabuľka 6.4: Výsledky nástroja `PhishCore` na testovacej sade čistých dát. Zobrazený je počet pravidiel klasifikácie phishclass, ktoré boli splnené na danej časti vzorky.

cou sadou a výsledné klasifikácie boli uložené v replike databázy `Athos`. Výsledné klasifikácie súborov a URL ukazuje tabuľka 6.4.

Zo získaných výsledkov je vidno že klasifikácia súborov obsahuje pomerne veľa nerozhodnutých súborov patriacich do klasifikačných tried `Suspicious` a `Unknown` a pomerne málo súborov patriacich do triedy `Clean`. Tento jav je spôsobený náročnosťou definície klasifikačných pravidiel `Clean`, keďže je jednoduchšie detegovať techniky používané u phishing útoku ako ich absenciu. Na tejto sade vznikol aj jeden falošný poplach kedy došlo k vyhodnoteniu súboru klasifikáciou `Phishing` na základe množstva splnených pravidiel klasifikácie `Suspicious`. Tento jav vedie k zvýšeniu hranice pre zmenu klasifikácie na `Phishing` spolu so zvyšujúcim sa množstvom používaných `YARA` pravidiel. Klasifikácia `NULL` značí zlyhanie nástroja pri spracovaní phishing vzorky zapríčinené chybou knižníc pre spracovanie modelu súboru a URL. Klasifikácia URL má oproti súborom vysokú úspešnosť vďaka použitiu doplnkových dát pre analýzu, ktoré obsahujú bielu listinu URL používanú pri analýze malware. Celkovú úspešnosť nástroja na tejto testovacej sade ukazuje tabuľka 6.5.

	Súbor	**URL**
počet správnych klasifikácií	2229 (6.47%)	23747 (68.9%)
celkový počet vzoriek	34443	34443

Tabuľka 6.5: Úspešnosť nástroja PhishCore na testovacej sade čistých vzoriek.

6.2.2 PhishTank

V tomto experimente bola použitá sada vzoriek zo zdroja PhishTank popísaná v sekcií 6.1.3 a sada YARA pravidiel popísaná v sekcií 6.1.1. Nástroj `PhishCore` bol spustený nad testovacou sadou a výsledné klasifikácie boli uložené v databáze `Athos`. Tabuľka 6.6 ukazuje výsledky klasifikácie dátovej sady.

Pre vyhodnotenie klasifikácie boli použité výsledky nástroja `VirusTotal`. Nástroj `VirusTotal` pre špecifikované súbory vracia výsledky skenov pomocou dostupných detekčných nástrojov. Výsledkom analýzy súboru je zoznam detekčných nástrojov a detekčné vzory(detekcie) splnené nad daným súborom. Pre porovnanie výstupnej klasifikácie bol použitý počet nástrojov s detekciou pokrývajúcou daný súbor. Počet nástrojov s detekciou bol následne prevedený na triedy klasifikácie odpovedajúce výsledkom nástroja `PhishCore`. Výsledná klasifikácia patrila do triedy `Clean` v prípade, že pre daný súbor existovalo menej

		klasifikácia vzorky					
		Phishing	**Suspicious**	**Unknown**	**Harmless**	**Clean**	**NULL**
model	**súbor**	687	1744	140	0	367	15
	URL	41	796	1652	0	317	300

Tabuľka 6.6: Počet klasifikácií nástroja `PhishCore` pre danú časť vzorky pre vzorky z testovacej sady vzoriek phishing zo zdroja `PhishTank`.

ako tri nástroje s detekciami tohoto súboru. Následne do triedy `Phishing` boli zaradené súbory s tromi a viac detekciami z dostupných nástrojov. Z dôvodu nemožnosti presného rozdelenia boli zanedbané klasifikácie `Suspicious` a `Harmless`.

	Phishing	**Clean**
počet správnych klasifikácií	561 (81.65%)	364 (99.18%)
celkový počet súborov	687	367

Tabuľka 6.7: Úspešnosť nástroja PhishCore na testovacej sade PhishTank.

Takto získaná klasifikácia z nástroja `VirusTotal` bola porovnaná s klasifikáciou nástroja `PhishCore`. Výslednú úspešnosť pre použité klasifikačné triedy ukazuje tabuľka 6.7. Tieto výsledky ukazujú na nedostatok klasifikačných pravidiel pre `Phishing` vzorky a vysokú úspešnosť rozoznania `Clean` vzoriek. Vysoká úspešnosť v rozoznaní `Clean` vzoriek je dôležitá, pretože falošný poplach nástroja má oveľa väčší dopad ako falošne negatívny poplach. Tento rozdiel je zohľadnený v testoch nástrojov a taktiež má vyšší dopad na užívateľa.

6.2.3 Manuálna klasifikácia

V tomto experimente bola využitá testovacia sada popísaná v sekcií 6.1.4 a sada YARA pravidiel popísaná v sekcií 6.1.1. Výsledky nástroja `PhishCore` po spustení nad touto testovacou sadou ukazuje tabuľka 6.8. Z celkového počtu 9943 vzoriek bolo správne klasifikovaných 74.49% vzoriek ako `Phishing`. Tieto výsledky vykazujú vysokú úspešnosť nástroja nad touto dátovou sadou. Spomedzi všetkých vzoriek bolo ďalších 22.70% vzoriek vyhodnotených ako `Suspicious`. Takto klasifikované vzorky môžu byť po manuálnej kontrole analytikom zaradené do triedy `Phishing` avšak je potrebné odstrániť prípadné falošné poplachy týchto pravidiel na čistých vzorkách.

	klasifikácia vzorky					
	Phishing	**Suspicious**	**Unknown**	**Harmless**	**Clean**	**Conflict**
počet	7407 (74.49%)	2258 (22.70%)	75	2	6	195

Tabuľka 6.8: Úspešnosť nástroja PhishCore na testovacej sade manuálne klasifikovaných vzoriek.

Vysoká úspešnosť nástroja je vďaka využitiu definičných vzorov `Avast` ako jednu z charakteristík súboru a definícia generických pravidiel na základe známych vzorov v phishing útokoch. Použitie generických `Suspicious` pravidiel má za následok klasifikačné pravidlá vhodné pre kombináciu s ďalšími pravidlami a užitočné pri triedení a manuálnej analýze vzoriek neklasifikovaných ako `Phishing`.

6.2.4 Výkonnostné testovanie nástroja

Pri analýze phishing vzoriek je dôležité množstvo vzoriek, ktoré je nástroj schopný spracovať. Doba spracovania vzoriek nástroja `PhishCore` závisí na množstve použitých `YARA` pravidiel a definovaných heuristík. Heuristiky nástroja vyžadujú ďalšiu funkcionality ako napríklad vyhľadávanie vzoriek v súbore alebo aplikáciu regulárnych výrazov. Použité `YARA` pravidlá umožňujú dosiahnuť vysokú rýchlosť skenovania súbor avšak použitie komplexných regulárnych výrazov a funkcií `YARA` modulu môže viesť k spomaleniu skenovania. Najdôležitejším faktorom spracovania vzoriek je komunikácia s ostatnými nástrojmi a to najmä vytvorenie klasifikácie súboru v centrálnom úložisku súborov.

	dávka dát		
číslo merania	dĺžka[s]	**1**	**2**	**3**
	1	1015	1069	1064
	2	1028	1035	1064
	3	1009	1080	1055
	4	1045	1043	1036
	5	1028	1070	1069
	priemer	1025	1059.4	1057.6

Tabuľka 6.9: Dĺžka analýzy vybraných phishing vzoriek.

Podľa expertov pre analýzu phishing firmy Avast je priemerná doba manuálnej analýzy nezaradených vzoriek 1-5 minút podľa sofistikovanosti útoku a podobnosti vzoriek. Priemerná doba analýzy phishing vzorky nástroja `PhishCore` spočítaná na základe meraní uvedených v tabuľke 6.9 je 1.047 sekúnd na 1 vzorku phishing. Priemerná doba analýzy bola získaná na základe piatich meraní dĺžky analýzy pre každú z troch dátových sád skladajúcich sa z náhodne vybraných 1000 vzoriek. Oproti manuálnej analýze tak ide o značné zrýchlenie avšak je potrebné počítať s časom pre definíciu `YARA` pravidiel a návrh nových heuristík. Pri opakujúcich sa vzoroch v phishing vzorkách a klasifikácií nových vzoriek na základe týchto vzorov - `YARA` pravidiel nástroj `PhishCore` prináša značné zrýchlenie analýzy a klasifikácie phishing vzoriek.

Kapitola 7

Záver

V rámci tejto práce som popísal phishing útoky a sociálne i technické aspekty phishing útoku. Na základe týchto techník som analyzoval dostupné nástroje pre detekciu phishing útokov. Tieto nástroje ukazujú rozdielne použité techniky detekcie, formát vstupných dát a teda rozličné charakteristiky a úspešnosť. Z dostupných nástrojov ďalej popisujem nástroj `YARA` a jeho využitie pre analýzu phishing útokov. Dostupné nástroje pre detekciu phishing útokov avšak nie sú vhodné pre klasifikáciu webových stránok na základe charakteristík adresy URL a obsahu stránky.

Navrhujem nástroj, ktorý kombinuje dostupné techniky a využíva koncept kombinovanej analýzy súborov a URL pomocou `YARA` pravidiel. Tento nástroj rozširuje nástroj `YARA` prostredníctvom modulu pre zlepšené možnosti detekcie a klasifikácie phishing útokov. Využitie vlastného modulu prináša možnosť definície vlastných `YARA` pravidiel na základe modelu definovaného nad štruktúrou súboru a URL.

Implementácia navrhovaného nástroja zahrňuje extrakciu charakteristík vstupného súboru a URL do modelových tried, vytvorenie tried obsahujúcich heuristiky súboru a URL a vytvorenie modelu phishing vzorky obsahujúcich získané triedy. Na základe modelu phishing vzorky sú aplikované `YARA` pravidlá definované v nástroji. Zoznam splnených pravidiel je použití pri vytvorení klasifikácie súboru a URL. Výsledkom spracovania súboru a URL nástrojom je vytvorený model uložený pre ďalšie využitie a klasifikácia uložená v databázy pre využitie v ďalších nástrojoch pre analýzu malware.

Pre účely testovania nástroja boli vytvorené testovacie sady na základe ktorých boli prevedené experimenty s cieľom vyhodnotiť úspešnosť nástroja. V experimente využívajúcom testovaciu sadu vzoriek neobsahujúcich phishing vznikol pri použití nástroja len jeden falošný poplach, čo ukazuje vysokú dôveryhodnosť nástroja. Testovacia sada založená na dátach zo zdroja phishing `PhishTank` bola využitá v experimente, ktorý ukazuje úspešnosť klasifikácie až 99.18% na čistých vzorkách a potrebu veľkého množstva klasifikačných pravidiel pre vzorky `phishing`. V experimente, ktorý porovnáva manuálnu klasifikáciu vzoriek a vytvorenú automatickú klasifikáciu bola dosiahnutá úspešnosť klasifikácie až 74.49%. Experiment pre výkonnostné testovanie nástroja ukazuje približne 100 krát zrýchlenie automatickej klasifikácie oproti manuálnej analýze analytikom.

Okrem vytvorenia automatickej klasifikácie je výhodou implementácie pomocou `YARA` pravidiel možnosť sledovania rozšírenia phishing útokov, jednotlivých phishing kampaní a cieľových spoločností útokov. Získanie týchto informácií je dôležité pre zlepšenie schopnosti detekcie phishing útokov, sledovanie trendov v phishing útokoch, vyhľadávanie neznámych variánt phishing útokov a využitie pri návrhu ďalších nástrojov pre detekciu phishing.

Výsledný nástroj slúži pre automatickú klasifikáciu vzoriek `phishing` útokov v prostredí firmy `Avast`. Využitie nástroja okrem klasifikácie vzoriek spočíva v prínose pre manuálnu analýzu týmu analytikov malware a phishing. Na základe vytvorenej klasifikácie a získaných charakteristík vstupných dát sú vytvárané detekčné vzory pre ochranu koncových užívateľov klienta `Avast` pred phishing útokmi.

Rozšírenie tohto nástroja do budúcna spočíva v návrhu nových pravidiel pre klasifikáciu vzoriek a pridanie heuristík modelu na základe nových techník autorov phishing útokov. Výsledky nástroja môžu byť vyžité v nástrojoch využívajúcich strojové učenie pre detekciu malware alebo analýzu phishing pomocou grafových databáz.

Literatúra

[1] Alexa: *Keyword Research, Comptetitive Analysis, & Website Ranking | Alexa*. [Online; navštívené 22.5.2019].
URL https://www.alexa.com/

[2] APWG: *Phishing Attack Trends Report - 3Q 2018*. [Online; navštívené 22.5.2019].
URL http://docs.apwg.org/reports/apwg_trends_report_q3_2018.pdf

[3] APWG: *Unifying the Global Response to Cybercrime | APWG*. [Online; navštívené 22.5.2019].
URL https://www.antiphishing.org/

[4] Australian Competition & Consumer Commision: *Nigerian scams | Scamwatch*. [Online; navštívené 22.5.2019].
URL https://www.scamwatch.gov.au/types-of-scams/unexpected-money/nigerian-scams

[5] Costello, A.: Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA). RFC 3492, The Internet Engineering Task Force (IETF), Marec 2003.
URL https://www.ietf.org/rfc/rfc3492.txt

[6] Crockford, D.: The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627, The Internet Engineering Task Force (IETF), Júl 2006.
URL https://www.ietf.org/rfc/rfc4627.txt

[7] Google: *Safe Browsing – Google Safe Browsing*. [Online; navštívené 22.5.2019].
URL https://safebrowsing.google.com/

[8] Hupp, A.: *python-magic · PyPI*. [Online; navštívené 22.5.2019].
URL https://pypi.org/project/python-magic/

[9] isitPhishing: *isitPhishing - Anti phishing tools and information*. [Online; navštívené 22.5.2019].
URL https://www.isitphishing.ai/

[10] Kaspersky: *Kaspersky Security Bulletin 2018. Threat Predictions for 2019 | Securelisth.* [Online; navštívené 22.5.2019].
URL https://securelist.com/kaspersky-security-bulletin-threat-predictions-for-2019/88878/

[11] Laskowski, B.: *Yara Ruleset for scanning Linux servers for shells, spamming, phishing and other webserver baddies.* [Online; navštívené 22.5.2019].
URL https://github.com/Hestat/lw-yara/tree/master/includes

[12] Lehtinen, P.: *Jansson — C library for working with JSON data.* [Online; navštívené 22.5.2019].
URL http://www.digip.org/jansson/

[13] Lookout: *Phishing AI (@PhishingAi) | Twitter.* [Online; navštívené 22.5.2019].
URL https://twitter.com/phishingai

[14] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, R. Nicholas: Internet X.509 Public Key Infrastructure: Certification Path Building. RFC 4158, The Internet Engineering Task Force (IETF), September 2005.
URL https://tools.ietf.org/html/rfc4158

[15] Majestic: *Majestic Million - Majestic.* [Online; navštívené 22.5.2019].
URL https://majestic.com/reports/majestic-million

[16] P. Cain, D. Jevans: Extensions to the IODEF-Document Class for Reporting Phishing. RFC 5901, The Internet Engineering Task Force (IETF), Júl 2010.
URL https://www.ietf.org/rfc/rfc5901.txt

[17] PhishTank: *PhishTank | Join the fight against phishing.* [Online; navštívené 22.5.2019].
URL https://www.phishtank.com/index.php

[18] R. Fielding, J. Reschke: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231, The Internet Engineering Task Force (IETF), Jún 2014.
URL https://tools.ietf.org/html/rfc7231

[19] Ragan, S.: *Kit Hunter: A basic phishing kit detection tool.* [Online; navštívené 22.5.2019].
URL https://github.com/SteveD3/kit_hunter

[20] Richardson, L.: *beautifulsoup4 · PyPI.* [Online; navštívené 22.5.2019].
URL https://pypi.org/project/beautifulsoup4/

[21] Symantec: *2018 Internet Security Threat Report.* [Online; navštívené 22.5.2019].
URL https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf

[22] T. Berners-Lee, R. Fielding, L. Masinter: Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, The Internet Engineering Task Force (IETF), Január 2005.
URL https://tools.ietf.org/html/rfc3986

[23] Tulloch, M.: *Microsoft Encyclopedia of Security*, *1*, ročník 1. One Microsoft Way, Redmond, Washington: Microsoft Press, prvé vydanie, Júl 2003, ISBN 0-7356-1877-1.

[24] VirusTotal: *The Python Interface for YARA*. [Online; navštívené 22.5.2019].
URL https://github.com/VirusTotal/yara-python

[25] VirusTotal: *VirusTotal - Free Online Virus, Malware and URL Scanner*. [Online; navštívené 22.5.2019].
URL https://www.virustotal.com/en/

[26] VirusTotal: *YARA - The pattern matching swiss knife*. [Online; navštívené 22.5.2019].
URL https://virustotal.github.io/yara/

[27] Yara: *Cuckoo module - yara 3.8.1 documentation*. [Online; navštívené 22.5.2019].
URL https://yara.readthedocs.io/en/v3.8.1/modules/cuckoo.html

Príloha A

Obsah priloženého DVD

- `data_sets` - Dátové sady použité v experimentoch s nástrojom `PhishCore`.
- `phish` - Zdrojové kódy implementovaného `YARA` modulu `phish`.
- `PhishCore` - Zdrojové kódy implementovaného nástroja `PhishCore`.
- `tex` - Zdrojové kódy technickej správy.
- `dp_xbenom01.pdf` - Text technickej správy.