SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1206

Poboljšanje kvalitete mamaca za operatora prijenosnog sustava temeljem javno dostupnih podataka

Luka Sever

Zagreb, srpanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 1206

Pristupnik:	**Luka Sever (0036530999)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Stjepan Groš

Zadatak: **Poboljšanje kvalitete mamaca za operatora prijenosnog sustava temeljem javno dostupnih podataka**

Opis zadatka:

Mamci imitiraju stvarne uređaje i sustave u računalnoj mreži kako bi privukli napadače. Kada napadači kompromitiraju mamac, moguće je proučavati njihovo djelovanje i prikupljati spoznaje o alatima i tehnikama koje koriste. Tako dobivene spoznaje mogu pomoći u povećanju razine sigurnosti stvarnih sustava. U sklopu ranijeg projekta razvijena je jezgra prototipa mamca koja imitira dio mreže operatora prijenosnog sustava. Navedeni prototip potrebno je dovršiti kako bi ga se moglo koristiti u daljnjim eksperimentima. U sklopu završnoga rada potrebno je istražiti javno dostupne informacije o operatoru prijenosnog sustava i materijale koji opisuju SCADA sustav otvorenog koda Hat Open. Na temelju prikupljenih informacija i postojećih komponenti prototipa potrebno je razviti korisničko sučelje za ugrađeni SCADA sustav i proširiti prototip s imitacijama dodatnih komponenti karakterističnih za prijenosni sustav. Uz navedeno, potrebno je i predložiti daljnje korake koje je potrebno poduzeti kako bi uobičajene metode prikupljanja OSINT podataka napadače dovele do mamca. Radu priložiti izvorni kôd. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

Hvala Boži Kopiću iz Končar Digital na pomoći s razumijevanjem Hat-Open konfiguracijskih datoteka i kolegama koji su čitali rad i dali mišljenja i prijedloge.

SADRŽAJ

1. Uvod

Kako napreduje razvoj industrije i čovječanstvo se sve više oslanja na tehnologiju nastaju nove prijetnje sustavima koji su postali neizostavni dio današnjeg društva. Sve većom povezanošću industrijskih upravljačkih sustava, sustava koji upravljaju vitalnim industrijskim procesima, rađaju se novi načini na koje se takvi sustavi mogu ugroziti. Zastrašujuća je pomisao da netko, bez fizičkog pristupa sustavu i potencijalno tisućama kilometara udaljen može ugroziti rad sustava koji upravlja nuklearnom elektranom, vodoopskrbnim sustavom ili elektroenergetskom mrežom, i time ne samo prekinuti opskrbu vitalnih resursa već uzrokovati ogromnu materijalnu štetu i ugroziti ljudske živote. Pod okriljem tih prijetnji nastaje ideja mamaca, sustava na mreži koji izgledaju i ponašaju se kao stvarni sustavi za upravljanje industrijskim procesima, no zapravo su samo simulacija takvog sustava čiji je cilj suprotan onome stvarnog sustava, a to je privući napadače u cilju traćenja njihovog vremena, potencijalnog otkrivanja njihovog identiteta, i najvažnije, učenju o njihovim metodama i načinima napada. Naoružani takvim znanjem, operateri prijenosnih sustava mogu bolje zaštititi stvarne sustave. Ne samo da su otežali napadačima pronalazak stvarnog sustava i stvorili mehanizme kojima im napadači nenamjerno odaju svoje metode, već su dobili nevjerojatnu priliku barem ostati u korak s napadačima. Napadači često prve informacije o tvrtkama i njihovim sustavima dobivaju iz javno dostupnih izvora, koristeći javno dostupne informacije. Iako su same po sebi naizgled bezopasne, mogu dati puno informacija o meti.

U ovom radu opisana je nadogradnja postojećeg prototipa mamca za operatera prijenosnog sustava koji je razvijen u sklopu ranijeg projekta. Opisana je implementacija ranije izgrađenog grafičkog sučelja te dodavanje novog čvora u mrežu mamca koji simulira programibilni logički kontroler, kritičnu komponentu u industrijskim upravljačkim sustavima.

2. Operateri prijenosnog sustava i industrijski upravljački sustavi

Operateri prijenosnog sustava i industrijski upravljački sustavi usko su povezane teme. Unatoč generalnoj neupoznatosti građanstva s njima, oni su od iznimne važnosti za funkcioniranje današnjeg društva i pružaju čovječanstvu neke od najznačajnijih privilegija današnjice, kao što su primjerice opskrba električnom energijom, vodom i plinom.

2.1. Operateri prijenosnog sustava

Operater prijenosnog sustava (engl. *transmission system operator*) ili TSO entitet je koji ima zadaću upravljanja prijenosnim sustavom, pod što spadaju električna energija i prirodni plin.[2] Ovaj rad fokusirat će se na operatera prijenosnog sustave električne energije i za primjer uzeti HOPS (Hrvatski operater prijenosnog sustava) koji je odgovoran za vođenje elektroenergetskog sustava, prijenos, razvoj te izgradnju prijenosne mreže Republike Hrvatske.[1]

2.2. Industrijski upravljački sustavi

Industrijski upravljački sustavi, (engl. *industrial control systems*), skraćeno ICS, imaju važnu ulogu u raznim industrijama gdje sudjeluju ili uvelike obavljaju zadaću upravljanja vitalnim industrijskim procesima, od kojih su najznačajnije elektroenergetska, vodoopskrbna, telekomunikacijska i nuklearna.[19] Samim time, česta su meta malicioznih aktera, posebice u obliku naprednih ustrajnih prijetnji (engl. *advanced persistent threat*), iza čijih aktora često stoje države ili snažne organizacije koje imaju pristup velikim, naizgled beskonačnim resursima. Daleko najpoznatiji primjeri takvih napada su *Stuxnet* 2010. te napad na Ukrajinsku elektroenergetsku mrežu u prosincu 2015.

godine.

2.3. SCADA

SCADA (engl. *supervisory control and data acquisition*) sustavi su sustavi koji se koriste za nadzor, upravljanje procesima i prikupljanje podataka u industrijama poput naftne, plinske, vodene i elektroenergetske industrije. Omogućavaju operaterima sustava da na jednom mjestu motre različita stanja i procese koji se odvijaju, pokreću i zaustavljaju razne procese i mijenjaju postavke sustava. SCADA sustav također pruža vizualizaciju svih podataka, te redovni izvoz podataka u sigurnosne kopije (engl. *backup*).[9]

Esencijalne komponente SCADA sustava su HMI, sustav nadgledanja, RTU, PLC i komunikacijska infrastruktura.

HMI (engl. *human machine interface*) ima zadaću pružanja svojevrsnog mosta između sustava i čovjeka - radi se o grafičkom i/ili naredbenom sučelju putem kojeg operater komunicira sa sustavom.

Sustav nadgledanja referira se na sam server i serverski proces upravljanja SCADA sustavom, prikupljanje podataka i njihovo slanje na razna sučelja koja su spojena na SCADA uređaj, no samim time i dio SCADA uređaja.

RTU (engl. *remote terminal unit*) ima ulogu prikupljanja podataka i kontroliranja raznih senzora i instrumenata te komunikacije sa središnjim SCADA sustavom, te služi kao poveznica između krajnjih uređaja i SCADA sustava.

Programibilni logički kontroleri (engl. *programmable logic controller*) ili PLC programibilni su uređaji koji autonomno i automatizirano ovisno o podacima na njihovom ulazu kontroliraju drugo industrijsko sklopovlje. Jedna su od najbitnijih komponenti u industrijskim upravljačkim sustavima upravo zbog toga što reguliraju rad kritičnih uređaja.[19]

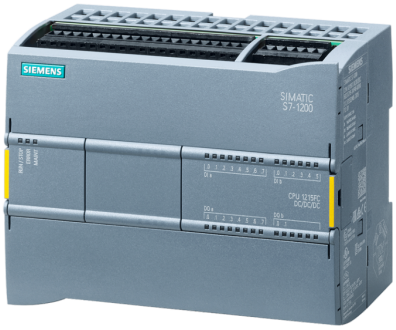
2.4. PLC

Programibilni logički sklopovi ili PLC uređaji su uređaji čija je svrha automatizacija industrijskih procesa. Sastoje se od procesora, programibilne memorije te ulaza i izlaza. PLC uređaji mogu se koristiti za kontrolu raznih uređaja u mnogim industrijama, a ovaj rad fokusirat će se na njihovu upotrebu u elektroenergetskoj industriji. Na slici 2.1 prikazan je jedan PLC uređaj proizvođača Siemens.

PLC sklopovi izgrađeni su da dugo traju unatoč radu u teškim uvjetima, otporni su na vibracije, udarce, elektromagnetske smetnje i druge vanjske utjecaje.

Napadači mogu iskoristiti ranjivosti u PLC uređajima za kompromitaciju industrijskih procesa, što može dovesti do velikih gubitaka u proizvodnji i/ili oštećenja opreme.

Zbog važne uloge PLC uređaja koja proizlazi iz njihove zadaće, važnosti industrije čijim procesima upravljaju i ovisnosti populacije o tim industrijama, važnost očuvanja sigurnosti PLC uređaja kao i opasnost koja prijeti iz njihovih ranjivosti ne može se podcijeniti.



Slika 2.1: Siemens S7-1200 PLC

2.5. Primjeri napada na industrijske upravljačke sustave

S porastom povezivanja ICS (engl. *industrial control systems*) sustava s računalnim mrežama i ostatkom interneta raste i broj mogućih vektora napada na takve sustave. Napadi na ICS sustave mogu biti ozbiljnih posljedica, od oštećenja skupe opreme i ispadanja važne infrastrukture iz pogona do gubitka ljudskih života. U nastavku će se navesti dva vrlo napredna primjera takvih napada, *Stuxnet*, koji je napao sustav iranske nuklearne elektrane te napad na ukrajinsku elektroenergetsku mrežu.

2.5.1. Stuxnet

Stuxnet je napredni računalni crv otkriven 2010. godine te se procjenjuje da je oštetio petinu nuklearnih centrifuga u Iranu i jedan je od najnaprednijih poznatih zloćudnih programa tog vremena.[11] Stuxnet je napad na industrijski upravljački sustav nuklearne elektrane koji je iskorištavao veliki broj ranjivosti od kojih su mnoge bile ranjivosti nultog dana. Jedna od njih bila je upravo ranjivost u Siemens SCADA sustavu[21],

a glavna meta bili su PLC uređaji koji su kontrolirali kritične procese u nuklearnoj elektrani.[19]

Stuxnet je bio izuzetno sofisticiran i kompleksan napad koji je zahtijevao visoku razinu znanja, resursa i planiranja. Smatra se napadom koji je svijetu odnosno industriji dao do znanja da industrijski upravljački sustavi nisu sigurni kao i da dobro financirani maliciozni akteri vjerojatno mogu uspješno napasti bilo koji sustav.

2.5.2. Napad na Ukrajinsku elektroenergetsku mrežu

Prvi poznati uspješni napad na elektroenergetsku mrežu dogodio se u Ukrajini u prosincu 2015. godine, gdje se procjenjuje da je bez struje ostalo oko 250000 ljudi.[14] Trideset trafostanica izbačeno je van funkcije te je bilo potrebno oko tri sata da većini kućanstava ponovo proradi opskrba električnom energijom.[14] SCADA sustavi bili su pod tuđom kontrolom a podaci na monitorima operatera vjerojatno su bili zamrznuti kako bi oni bili uvjereni da sve funkcionira regularno.[15] Nakon samog napada, tvrtka koja upravlja elektroenergetskom mrežom u Ukrajini, *Kyivoblenergo*, zaprimala je tisuće lažnih poziva u TDoS napadu (engl. *telephony denial of service*) koji je onemogućio legitimnim korisnicima komunikaciju s tvrtkom i prijavljivanje kvarova.

Posebnost ovog napada krije se u količini planiranja koja je morala stajati iza njega kao i njegova sofisticiranost i potrebna koordinacija ne bi li on bio uspješan. Ovaj napad na jasno pokazuje da su napadači postali vrlo sposobni u ciljanju kritične infrastrukture i da su spremni uložiti značajne napore kako bi postigli svoje ciljeve. Napredak tehnologije i sofisticiranost napadača zahtijevaju stalno poboljšanje sigurnosnih mjera i aktivno praćenje najnovijih prijetnji kako bi se osigurala zaštita industrijskih upravljačkih sustava.

3. Mamci

Mamci (engl. *honeypots*) računalni su sustavi koji pokušavaju izgledati kao legitiman sustav s potencijalno izloženim setom ranjivosti koje se mogu iskoristiti i sustav kompromitirati.[19] Cilj takvih sustava je privući napadače i analizirati njihove metode napada te na temelju toga razviti obranu od takvih napada. Mamci se mogu koristiti i za prikupljanje informacija o samim napadačima, a u najgorem slučaju napadač će potratiti dio svog vremena i svojih resursa na mamac umjesto na pravi sustav.

3.1. Vrste mamaca

Mamci mogu biti različitih razina složenosti te samim time i vjerodostojnosti kojom simuliraju određeni sustav. Jedna od glavnih podjela koja se može napraviti je podjela na mamce niske i visoke razine interakcije, odnosno na mamce koji su jednostavni za implementirati i održavati no mogu zavarati samo na prvi pogled te mamce zahtjevnije implementacije i skupljeg procesa održavanja, i vremenski i financijski, no mogu zavarati ne samo napadače već i alate napravljene za detekciju mamaca.

3.1.1. Mamci niske razine interakcije

Mamci niske razine interakcije (engl. *low interaction*) simuliraju stvarni sustav na niskoj razini, odnosno simuliraju samo određene servise i/ili ranjivosti i niske su kompleksnosti. To ih čini lakim za implementaciju no zato ih je vrlo lako detektirati i izbjeći. Ako napadač u početnoj fazi napada skenira mrežu i da se radi o mamcu, vjerojatno će odustati od napada na taj sustav. No ako napadač ne otkrije da se radi o mamcu, a radi se o mamcu niske razine interakcije, on će vjerojatno moći izvesti samo dio napada jer se sustav neće ponašati kao stvarni sustav, te će eventualno tad shvatiti da se radi o mamcu i odustati od napada. Iako i dalje mogu služiti kao sustav ranog upozorenja, činjenica da napadači neće moći izvesti cijeli napad znači da mamci niske razine interakcije mamci ne uspijevaju saznati mnogo o napadaču te se propušta prilika

za učenje o napadačima i njihovim metodama.

3.1.2. Mamci visoke razine interakcije

Mamci visoke razine interakcije (engl. *high interaction*) na velikoj razini vjernosti simuliraju stvarni sustav i samim time teški su za implementaciju i zahtijevaju veliku količinu resursa. Njihovoj jednostavnoj izgradnji i implementaciji prijeti i činjenica da sustavi koje emuliraju često nisu otvorenog koda (engl. *open-source*), odnosno nije u potpunosti poznato kako određeni sustavi funkcioniraju. Dobar primjer toga su Siemens PLC uređaji koji se koriste u industrijskim upravljačkim sustavima i koriste Siemens protokol S7comm zatvorenog koda (engl. *closed-source*), u smislu da nemaju nikakvu javno dostupnu tehničku dokumentaciju što uvelike otežava razumijevanje protokola. Samim time, postoji mogućnost raskrinkavanja mamca prilikom napadačevog otkrivanja da neki aspekt protokola ne funkcionira kao pravi uređaj. Pod pretpostavkom da osoba koja napada PLC uređaj ima stručnog iskustva u radu s njima, moguće je da brzo shvati da nešto nije u redu ako mamac nije dovoljno precizno napravljen. No ako se mamac uspješno i vjerodostojno izgradi i implementira, napadač će teško moći razlikovati mamac od stvarnog sustava te će vjerojatno postupati kao što bi postupao s pravim sustavom. U tom će slučaju, ako je dobro napravljen, bilježiti sve aktivnosti napadača. To uključuje spremanje svega što se upisuje u ljusku, preuzima na sustav ili ubacuje i pokušava modificirati sam kod koji je zaslužan za upravljanje fizičkim uređajima.

3.2. Primjeri mamaca

U ovom poglavlju navodi se nekoliko mamaca različitih vrsta i različitih razina interakcije, te su samo mali dio širokog izbora mamaca i alata koji se u te svrhe mogu iskoristiti. Svi navedeni mamci otvorenog su koda.

3.2.1. Cowrie

Cowrie je mamac srednje do visoke razine koji bilježi iscrpne (engl. *brute-force*) napade na SSH i Telnet servise. Primjer je mamca koji nema izravne veze sa ICS sustavima. U načinu rada mamca srednje razine emulira podskup funkcionalnoti UNIX ljuske, a u načinu rada visoke razine radi kao posrednik (engl. *proxy*) između napadača i stvarnog sustava, tako prikupljajući dodatne informacije o napadaču te omogućava

napadaču da izvede cijeli napad, dok mamac bilježi sve njegove aktivnosti motreći sav promet.[20]

3.2.2. Cisco ASA honeypot

Cisco ASA honeypot dobar je primjer mamca niske razine interakcije koji ne samo da površno emulira neku funkcionalnost već je mamac odnosno komponenta mamca sposobna detektirati isključivo iskorištavanje ranjivosti CVE-2018-0101.[6]

3.2.3. T-Pot

T-Pot primjer je mamca koji oponaša široki spektar servisa i usluga tako što integrira mnoštvo drugih mamaca otvorenog koda i nudi nekoliko efikasnih načina za pokretanje i puštanje u pogon (engl. *deployment*). Istina je da koristi mnoštvo mamaca koji su niske razine interakcije, ali smatra se mamcem visoke razine interakcije ne samo zato što inkorporira i mamce visoke razine interakcije već mnoštvo raznovrsnih mamaca, time povećavajući mogućnost privlačenja različitih vrsta napadača i pružajući bogate informacije o njihovim namjerama i metodama. Pruža i mnoštvo raznih vizualizacija, jedna od kojih je mapa napada koja filtrira zahtjeve po IP adresama i smješta ih na grafičkom prikazu na odgovarajuću geografsku lokaciju.[24]

3.2.4. HoneyPLC

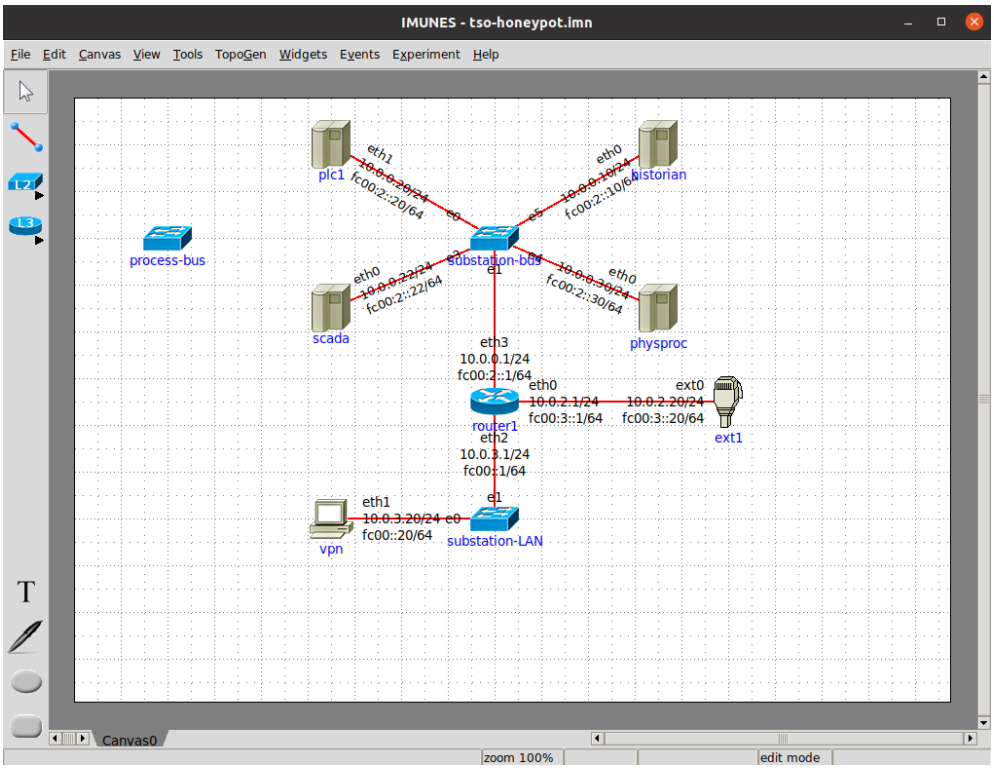
HoneyPLC[10] je mamac visoke razine interakcije koji s velikom razinom vjerodostojnosti emulira tri Siemens PLC uređaja (Siemens S7-300, S7-1200, i S7-1500), Allen-Bradley MicroLogix 1100 i ABB PM554-TP-ETH koji su bez potrebe za dodatnim postavljanjem po instalaciji mamca spremni za korištenje.

Postoje i drugi mamci koji se mogu koristiti za emulaciju PLC uređaja, no HoneyPLC je prvi svog tipa u kontekstu značajki koje pruža i razine vjerodostojnosti koju postiže.[19] Uspjeva zavarati čak i alat Shodan, koji je specijaliziran za skeniranje mreža i otkrivanje uređaja na njima, te simuliranim PLC uređajima daje *Honeyscore* koji je ekvivalentan stvarnim PLC uređajima, broj koji je procjena koliko je vjerojatno da je uređaj mamac, a ne stvarni uređaj.[19] Upravo zbog najbolje *Honeyscore* vrijednosti od PLC uređaja dostupnih za emuliranje pomoću alata HoneyPLC, u ovom radu za emulaciju PLC uređaja koristit će se Siemens S7-300 PLC uređaj.

U sklopu HoneyPLC nalazi se i *profiler* alat, koji je sposoban profilirati bilo koji PLC uređaj i omogućiti korisniku njegovo emuliranje.

4. Prototip TSO mamca

U sklopu prethodnog projekta razvijen je prototip mamca koji simulira dio mreže operatera prijenosnog sustava. Njegova mreža sastoji se od SCADA čvora i čvora koji je simulator procesa (čvor imena *physproc*) na jednoj lokalnoj mreži povezanih komutatorom i usmjerivača koji povezuje tu lokalnu mrežu sa vanjskim sučeljem i VPN čvorom. U sklopu mamca nalazili su se i čvorovi PLC te *historian* koji su pripadali lokalnoj mreži sa SCADA čvorom i simulatorom procesa, no oni nisu bili u funkciji te su maknuti iz prototipa zato što su zbog grešaka koje su se događale na samim čvorovima onemogućili pokretanje Imunes simulacije. Navedena topologija prikazana je na slici 4.1 u snimci zaslona programa Imunes.



Slika 4.1: Originalna topologija mreže mamca

Na *physproc* čvoru pokrenut je simulator procesa koji simulira mjerenja trafosta-

nice. On preko komutatora komunicira sa SCADA čvorom koristeći *iec104* protokol. Na SCADA čvoru nalazi se *hat-orchestrator* koji upravlja radom svih Hat servisa i može mu se pristupiti preko *ext1* čvora tako da se van Imunes simulacije u internet pregledniku posjeti IP adresa SCADA čvora koristeći port 23021. S njega je moguće pokretati i zaustavljati komponente koje se također nalaze na SCADA čvoru, a to su Hat GUI, Hat Monitor i Hat Manager.

Hat GUI poslužitelj omogućava pregledavanje i kontrolu funkcionalnosti Hat sustava odnosno simulirane trafostanice.[16] Nalazi se na portu 23023.

Hat Monitor poslužitelj pruža potrebnu infrastrukturu potrebnu za kontrolu nad više Hat komponenti istovremeno.[17] Nalazi se na portu 23022.

4.1. Korištene tehnogije i alati

U mamcu za operatera prijenosnog sustava ključne su tehnologije Docker i Imunes. U svrhu simuliranja mrežne infrastrukture mamca koristit će se alat Imunes, a za izgradnju samih čvorova u toj mreži koji su esencijalni funkcioniraju sustava za kontrolu industrijskih procesa koristit će se tehnologija Docker. Koriste se i drugi alati, a svi bitni su navedeni u nastavku.

4.1.1. Imunes

Imunes je mrežni simulator i emulator razvijen na Sveučilištu u Zagrebu gdje se koristi za potrebe obrazovanja i istraživanja[5] te omogućava testiranje mrežnih topologija i postavki prije njihove implementacije u stvarnom svijetu. Pruža mnoge mogućnosti prilikom izgradnja mreže, povezivanje poslužitelja i klijenata s nekoliko vrsta komutatora i usmjeritelja te NAT uređajem, a svaki uređaj može pružati razne usluge pokretanjem raznih servisa kao što su SSH, Telnet i FTP. To je dodatno prošireno mogućnošću pokretanja Docker kontejnera na čvorovima na beskonačan broj mogućih servisa, jer što god se može pokrenuti na stvarnom računalu u Linux okruženju može se pokrenuti i u Docker kontejneru, a samim time i na čvoru u Imunes mreži.

U kontekstu izgradnje mreže mamca za operatera prijenosnog sustava Imunes će se koristiti za izgradnju interne mrežne infrastrukture mamca, odnosno mreže koja će se nalaziti unutar mreže operatera prijenosnog sustava. Pruža mogućnost pokretanja Docker kontejnera na virtualnim računalima u mreži, što će se iskoristiti u ovom radu za pokretanje složenih servisa i funkcionalnosti mamca odnosno nekoliko njegovih komponenti.

4.1.2. Docker

Docker[8] je platforma otvorenog koda koja koristi kontejnerizaciju za izgradnju i pokretanje aplikacija. Kontejneri (engl. *containers*) su cjeline programskog koda koji u sebi sadrže sve što je samoj aplikaciji unutar njega potrebno za pokretanje, i tako su izolirani od ostatka sustava. To im također omogućava prenosivost, jer se mogu pokrenuti na bilo kojem sustavu koji podržava Docker.[12]

Docker slike (engl. *images*) ključni su element tehnologije Docker koje služe kao predlošci, nacrti za izradu Docker kontejnera. Slika se sastoji od nekoliko slojeva, a svaki sloj predstavlja jednu naredbu u tekstualnoj datoteci *Dockerfile* i služi izgradnji nove Docker slike, odnosno jednu naredbu koja se izvršava prilikom izgradnje slike pozivom naredbe *docker build*.

Nakon izgradnje slike ona se može koristiti kao predložak za stvaranje i pokretanje proizvoljnog broja kontejnera, koji su izolirani od ostatka sustava i međusobno, no imaju i mogućnost međusobnog komuniciranja u što ovaj rad neće ulaziti. Svaki kontejner izvodi se kao proces na Docker domaćinu (engl. *Docker host*), a svaki kontejner ima svoj vlastiti izolirani operacijski sustav te samim time i vlastiti datotečni sustav.

Docker slike mogu se preuzeti i sa stranice *Docker Hub* koja služi kao centralno mjesto odnosno repozitorij za dijeljenje Docker slika. Slike se mogu slobodno preuzeti i koristiti. Gotove slike, što uz slike sa stranice Docker Hub uključuje i slike koje su izgrađene lokalno, mogu se koristiti kao predlošci za izgradnju novih slika, tako da se one nadograđuju novim slojevima u novu sliku koja sadrži sve potrebno specifičnim potrebama korisnika odnosno sve potrebno za pokretanje aplikacije koja se nalazi u kontejneru.

4.1.3. Honeyd

Honeyd[22] je alat otvorenog koda namjenjen kreiranju i simulaciji virtualnih poslužitelja. Oni se mogu konfigurirati tako da pokreću proizvoljne servise čija se ponašanja ili osobnost (engl. *personality*) mogu dodatno prilagoditi tako da se ponašaju kao da se pokreću na raznim operacijskim sustavima.[22] Razvijen je kako bi olakšao razvoj i postavljanje mamaca, te je sposoban simulirati veliki broj virtualnih poslužitelja na jednom računalu od kojih se svaki može ponašati kao da je pokrenut na nekom određenom operacijskom sustavu.

Ponašanje poslužitelja određuje se konfiguracijskim datotekama koje opisuju njegovo ponašanje, određivanjem datoteke osobnosti (engl. *personality file*) koja opisuje osobnost poslužitelja. To znači da će se poslužitelj ponašati kao da je pokrenut na odre-

đenom operacijskom sustavu, što će u kontekstu ovog rada biti datoteka koja definira ponašanje PLC uređaja Siemens S7-300.

4.1.4. Hat-Open

Hat-Open[18] je biblioteka (engl. *library*) otvorenog izvora koju je razvio odnosno čiji razvoj sponzorira tvrtka Končar Digital. Sastoji se od skupa biblioteka koje olakšavaju razvoj aplikacija za udaljeno upravljanje i nadzor uređaja kao što su IoT (engl. *Internet of Things*) uređaji, PLC uređaji i slično, te je službeno još u stanju razvoja.

Jedna od ključnih karakteristika biblioteke Hat-Open je njezina modularnost i mogućnost prilagođadanja raznim potrebama korisnika. Pruža mogućnost pokretanja proizvoljnog broja komponenti korištenjem *hat-monitor* servera, upravljanje i nadzor komponenti korištenjem upravljača servisima *hat-orchestrator*, događajima potaknutu arhitekturu (engl. *event-driven architecture*) koje definira *hat-event* te komunikaciju između komponenti korištenjem *hat-gateway*, strukture podataka korištene za komunikaciju unutar jednog *Hat* sustava.[18]

4.2. Razvoj grafičkog sučelja

Za izradu grafičkog sučelja korišten je u sklopu ranijeg projekta razvijen GUI (engl. *graphical user interface*), kojim je bilo potrebno zamijeniti postojeći.

U sklopu ranije razvijenog GUI-ja nalazi se izvorni kod koji koristi biblioteke Hat-Open, te resursi potrebni za postavljanje Docker slike SCADA čvora.

4.2.1. Instalacija grafičkog sučelja

U sklopu projekta s razvijenim grafičkim sučeljem nalazi se sve potrebno za njegovo pokretanje u Imunes čvoru, te je potrebno zamjeniti dio postojećih datoteka novima. Najvažnije datoteke nalaze se u direktorijima *playground*, u kojemu su konfiguracijske datoteke za Hat komponente, te *src_js* i *src_py*, u kojima se nalaze JavaScript datoteke koje definiraju sam izgled grafičkog sučelja te Python datoteke u kojima je definirano ponašanje svih Hat komponenti.

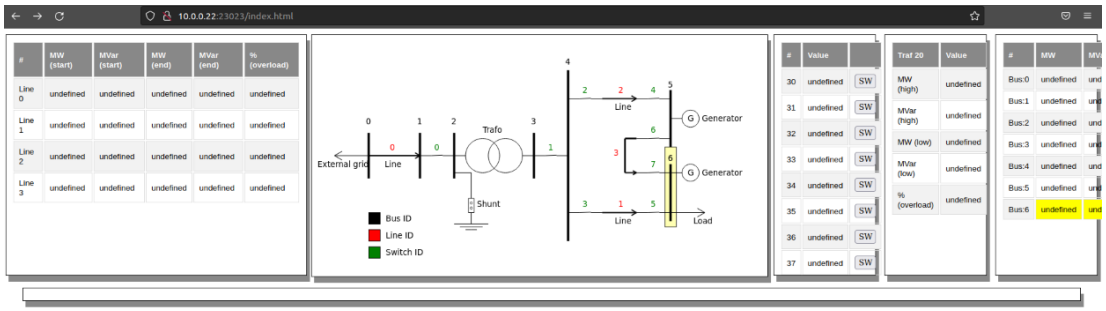
U originalnoj verziji modificiranog GUI-ja simulator procesa ne nalazi se na drugom čvoru već na istom čvoru kao i SCADA čvor, te je zbog toga potrebno modificirati dijelove koda.

U datoteci *src_py/project/devices/simulator_device_all.py* potrebno je na linijama 135 i 171 promjeniti adresu sa 127.0.0.1 na adresu *physproc* čvora, odnosno 10.0.0.30, te port na 12345, kako bi se omogućila komunikacija između simulatora i SCADA čvora.

```
address = iec104.Address('10.0.0.30', 12345)
```

Ispis 4.2.1.1: Spajanje na physproc čvor

Sad je moguće pokrenuti Imunes simulaciju, u internet pregledniku posjetiti adresu 10.0.0.23 na portu 23023 i pristupiti grafičkom sučelju koje se može vidjeti na slici 4.2.



Slika 4.2: Grafičko sučelje

4.3. Dodavanje PLC čvora

S obzirom na to da je ranije ustanovljeno da su PLC uređaji elementarni dio industrijskih upravljačkih sustava, u sklopu ovog rada pokušana je implementacija PLC čvora u sustav. Za to je korišten HoneyPLC, mamac visoke razine interakcije kojeg je razvio SEFCOM (Laboratory of Security Engineering for Future Computing) državnog sveučilišta u Arizoni, te koji ima mogućnost oponašanja nekoliko PLC uređaja.[10;19]

PLC čvor dodan je u Imunes simulaciju s IP adresom 10.0.0.10 u istu lokalnu mrežu kao čvorovi SCADA i *physproc*. U tom čvoru u mreži nalazit će se Docker kontejner na kojem je pokrenut HoneyPLC.

4.3.1. Preuzimanje zahtjeva

Za postavljanje čvora potrebno je napraviti Docker sliku koja sadrži HoneyPLC te je pokrenuti u kontejneru. S obzirom na to da je na GitHub repozitoriju HoneyPLC-a navedeno da je on testiran na operacijskom sustavu Ubuntu 18 LTS 64-bit za izgradnju

slike korištena je slika Ubuntu 18.04 koja se nalazi na platformi Docker Hub i može se preuzeti navođenjem njezinog imena u datoteci Dockerfile, na samom početku izgradnje slike.

```
FROM ubuntu:18.04
```

Ispis 4.3.1.1: Dohvaćanje bazne slike za PLC čvor u Dockerfile-u

Zahtjevi koji su navedeni u HoneyPLC uputama nalažu da je potrebno instalirati Honeyd, Python verziju 2.5 ili noviju, Python biblioteku *python-nmap* te biblioteke *snmpsim* i *lighttpd*.

Honeyd biblioteka dostupna je kao GitHub repozitorij, te budući da je potrebno klonirati repozitorij koristeći protokol SSH, kloniranje repozitorija obavit će se van Docker kontejnera, prije izgradnje slike, te će se na njega kopirati u trenutku izgradnje slike.

S obzirom na to da je Python2.5 zastario i ne može se preuzeti iz repozitorija, preuzeta je arhiva s mrežne stranice Python-a.

Za instalaciju Python biblioteke *python-nmap* nije bilo moguće koristiti upravljač paketa pip jer potrebna verzija paketa više nije putem njega dostupna, pa će se preuzeti arhiva s mrežne stranice koja i dalje ima dostupnu dovoljno staru verziju paketa. Arhiva nije dostupna za preuzimanje čak ni na mrežnoj stranici PyPI koja često sadrži zastarjele verzije paketa, pa će se koristiti arhiva preuzeta sa stranice `https://launchpad.net` Prije nastavljanja potrebno je *python-nmap* modificirati kako bi radio, greška do koje je došlo zastarjevanjem paketa. Datoteka koju je potrebno modificirati zove se *nmap.py* i nalazi se u *nmap* direktoriju unutar *python-nmap-0.1.4* direktorija. U njoj je potrebno modificirati regularni izraz koji se nalazi na liniji 116 u inicijalizacijskoj funkciji tako da se *http* zamjeni s *https*.

```
regex = re.compile('Nmap version [0-9]*\.[0-9]*[^ ]* \(
   http://nmap\.org \)')
```

Ispis 4.3.1.2: Linija koju je potrebno izmjeniti

Jedina stvar koja će se preuzeti upravljačem paketa je *snmpsim*. U uputama za instalaciju naveden je GitHub repozitorij pa je pretpostavljeno da bi se najnovija verzija mogla jednostavnije instalirati koristeći pip unutar Docker slike.

Budući da je poveznica za preuzimanje *lighttpd* biblioteke koja je navedena u uputama poveznica na internetsku stranicu određene verzije biblioteke, ona će se unutar Docker kontejnera preuzeti i instalirati.

Sve preuzete datoteke i direktoriji trebaju biti kopirani u Docker sliku, što se može obaviti naredbom *COPY* u Dockerfile-u koja je objašnjena kasnije u radu.

4.3.2. Instalacija zahtjeva

Kako bi se svi potrebni zahtjevi instalirali, uz preuzete datoteke i direktorije, prilikom izgradnje Docker slike na nju će se kopirati i skripta napisana u Bash-u namjenjena instalaciji svih potrebnih zahtjeva.

Kako bi se instalirali svi potrebni zahtjevi, prije svega potrebno je u prilikom izgradnje Docker slike instalirati sve potrebne pakete koji sudjeluju u instalaciji ili ih paketi koje pokušavamo instalirati zahtijevaju. Prije svega, potrebno je instalirati *sudo* paket. Ostali paketi bit će vidljivi u prikazu skripte, te ih se zbog brojnosti ovdje neće nabrajati ili opisivati.

Za instalaciju Python-a potrebno je koristeći naredbu *tar* otpakirati arhivu. Budući da se radi o arhivi ekstenzije *.tar.tgz*, naredba će se pokrenuti s opcijama *-xvzf*. Nakon otpakiravanja arhive potrebno je unutar direktorija koji je nastao kao rezultat otpakiravanja pokrenuti skriptu *configure* koja će pripremiti prevođenje i instalaciju Python-a s parametrom *–prefix=/usr/local/python2.5* kako bi se Python instalirao u direktorij */usr/local/python2.5*. Parametar nije nužan, ali je je dobra praksa paziti na lokaciju instalacije ako postoji nekoliko različitih verzija programa na računalu. Nakon toga naredbom *make* Python će se prevesti i izgraditi. Tad će se naredbom *checkinstall* iz prevedenog izvornog koda Python-a izgraditi paket koji će se odmah i instalirati ako se naredba pokrene s opcijom *install=yes*. Predat će se i opcija *pkgname python2.5* kako bi se paket nazvao *python2.5*, opcija *nodoc* koja specificira da ne treba generirati dokumentaciju, te opcija *default* koja će, umjesto da korisnika prilikom instalacije traži da unese neke podatke, koristiti zadane vrijednosti. Nakon toga Python bi trebao biti instaliran, no potrebno je još kreirati simboličku poveznicu (engl. *symbolic link*) prema izvršnoj datoteci Python-a i smjestiti ju u direktorij */usr/bin* kako bi se Python verzija 2.5 mogla pokrenuti iz bilo kojeg direktorija pozivanjem samo naredbe *python*. To je potrebno kako bi se Honeyd instalacija kasnije uspješno izvršila.

```
#!/bin/bash
apt update && apt -y install sudo
sudo apt upgrade -y
sudo apt install software-properties-common -y
sudo add-apt-repository ppa:gijzelaar/snap7
sudo apt-get update
```

```
sudo apt-get install checkinstall libsnap7-1 libsnap7-dev
   iproute2 rpcbind inetutils-inetd python-pip wget
   libpcre2-dev libbz2-dev libevent-dev libdumbnet-dev
   libpcap-dev libpcre3-dev libedit-dev bison flex libtool
   automake zlib1g-dev zlib1g make nmap net-tools vim nano -y

tar -xvzf Python-2.5.4.tgz
cd Python-2.5.4
./configure --prefix=/usr/local/python2.5
make
sudo checkinstall --pkgname python2.5 --provides python2.5
   --nodoc --install=yes --default
cd ..
rm Python-2.5.4.tgz # cleanup
ln -s /honey/Python-2.5.4/python /usr/bin/python
```

Ispis 4.3.2.1: Dio skripte - potrebne biblioteke, Python

Nakon toga instalirat će se koristeći pip upravitelj paketima Python biblioteka *snmpsim* zajedno sa *db-sqlite3*, kojeg zahtjeva Honeyd.

Instalacija samog alata Honeyd nešto je jednostavija od Python instalacije. Budući da je Honeyd preuzet ranijim kloniranjem GitHub repozitorija i kopiran na Docker sliku, potrebno ga je instalirati koristeći skripte i naredbe *autogen.sh, configure, make* i *make install* koje se nalaze u korijenskom direktoriju *Honeyd/*. Skripta *autogen.sh* generira sve potrebno za konfiguraciju, *configure* priprema prevođenje i instalaciju, *make* prevodi, a *sudo make install* instalira Honeyd.

Ako je paket *python-nmap* prebačen na Docker sliku prilikom izgradnje i modificiran po uputama iz prethodnog poglavlja, instalirat će se tako da se iz korijenskog direktorija izvodnog koda paketa pokrene naredba *python setup.py install*.

Alat *lighttpd* instalirat će se preuzimanjem arhive s internetske stranice alata programom *wget*, otpakiravanjem alata naredbom *tar* te pokretanjem skripti i naredbi *configure, make* te *make install*.

```
pip install db-sqlite3 snmpsim

cd honeyd
chmod +x autogen.sh configure
```

```
./autogen.sh
./configure
make
sudo make install
cd ..

cd python-nmap-0.1.4/
python setup.py install
cd ..
wget https://download.lighttpd.net/lighttpd/releases-1.4.x/
   lighttpd-1.4.71.tar.gz
tar -xvf lighttpd-1.4.71.tar.gz
cd lighttpd-1.4.71
./configure
make
make install
cd ..

wget https://download.lighttpd.net/lighttpd/releases-1.4.x/
   lighttpd-1.4.71.tar.gz
tar -xvf lighttpd-1.4.71.tar.gz
cd lighttpd-1.4.71
./configure
make
make install
cd ..
```

Ispis 4.3.2.2: Dio skripte - db-sqlite3, snmpsim, Honeyd, python-nmap, lighttpd

4.3.3. Ostatak instalacije i konfiguracija

Prije svega, sada je potrebno instalirati modificiranu biblioteku snap7 koja dolazi u sklopu HoneyPLC alata. Prije toga potrebno je instalirati originalni snap7 paket koristeći apt upravljač paketima. Nakon toga potrebno se pozicionirati u *./snap7/build/unix* direktorij koji se nalazi unutar HoneyPLC direktorija te pokrenuti naredbu *make -f x86_64_linux.mk install* kako bi se prevele i instalirane modificirane biblioteke snap7 paketa.

Jedna od važnijih značajki HoneyPLC alata je *fingerprinting* uređaja. *Fingerprin-*

ting se odnosi na proces identifikacije karakteristika ciljanih sustava temeljen na njihovim odgovorima na različite upite.

U sklopu alata HoneyPLC nalazi se pet direktorija koji sadrže datoteke s *fingerprint* potpisima u formatu kojeg definira alat nmap, a budući da je odabran PLC Siemens S7-300, potreban je njegov *fingerprint* kao i datoteka *./snap7/build/bin/x86_64-linux/libsnap7.so-300*. Datoteku *./snap7/build/bin/x86_64-linux/libsnap7.so-300* potrebno je kopirati u */usr/lib* direktorij, a datoteke s *fingerprint* potpisima odnosno datoteku *plc-profiles/Siemens S7-300/s7-300-nmap-fingerprint.txt* potrebno je dodati na kraj datoteke s potpisima koja se nalazi u Honeyd direktoriju na lokaciji */usr/share/honeyd/nmap-os-db*.

Uz to, potrebna je i izvršna datoteka s7comm poslužitelja koja se nalazi u direktoriju *snap7/examples/cpp/x86_64-linux/server* te koju je potrebno kopirati u direktorij */usr/share/honeyd* preimenovanu u *s7commServer*. Njegova funkcija je da služi kao most (engl. *gateway*) između Honeyd-a i PLC-a. Konačno, potrebno je promjeniti dozvole *s7commServer* izvršne datoteke kako bi Honeyd mogao pokrenuti poslužitelj, za što se pokreće naredba *chmod 777 /usr/share/honeyd/s7commServer*, koja će u Linux okruženju svim korisnicima dati apsolutna prava nad datotekom i tako omogućuje pokretanje poslužitelja alatu Honeyd, no program *s7commServer* može se i zasebno pokrenuti.

Potrebna je još jedna, konačna preinaka bez koje Imunes ne bi mogao pokrenuti kontejner. Naime, Imunes će prilikom stvaranja čvora dodati i izvršiti datoteku */boot.conf* koja računalu dodjeljuje IP adrese i započinje razne mrežne demon (engl. *daemon*) servise. Problem se javlja sa zadnjom naredbom u datoteci zbog toga što ona poziva naredbu *inet*, no prilikom instalacije tog paketa naredbom *sudo apt install inetutils-inetd* izvršna datoteka koja odgovara alatu inetd je datoteka */usr/bin/inetutils-inetd* pa je potrebno kreirati simboličku poveznicu */usr/bin/inetd* koja pokazuje na datoteku *inetutils-inetd* u istom direktoriju naredbom *ln*.

```
cd honeyplc-source/snap7/build/unix
make -f x86_64_linux.mk install
cd ../../../..

cp
   honeyplc-source/snap7/build/bin/x86_64-linux/libsnap7.so-300
 /usr/lib/
cp honeyplc-source/snap7/examples/cpp/x86_64-linux/server
```

```
    /usr/share/honeyd/s7commServer
chmod 777 /usr/share/honeyd/s7commServer

cat honeyplc-source/plc-profiles/Siemens\
    S7-300/s7-300-nmap-fingerprint.txt >>
    /usr/share/honeyd/nmap-os-db

ln -s /usr/sbin/inetutils-inetd /usr/sbin/inetd

echo "HoneyPLC set up!"
```

Ispis 4.3.3.1: Dio skripte - snap7, konfiguracija i kraj

Nakon pisanja skripte, potrebno je izgraditi Docker sliku koja će ju prilikom izgradnje, nakon kopiranja svih potrebnih resursa na sliku, i pokrenuti. Ta datoteka odnosno *Dockerfile* prikazana je u idućem isječku koda.

```
WORKDIR honey

COPY Honeyd honeyd
COPY honeyplc-docker-setup-script.sh .
COPY Python-2.5.4.tgz .
COPY python-nmap-0.1.4 python-nmap-0.1.4
COPY honeyplc-source honeyplc-source
RUN bash honeyplc-docker-setup-script.sh

# docker build -t imunes/honeyplc -f Dockerfile .
```

Ispis 4.3.3.2: Dockerfile za izgradnju honeyplc slike

Datoteka *Dockerfile* se sastoji od naredbi koje se izvršavaju prilikom izgradnje slike. Prva naredba *WORKDIR* specificira tekući direktorij iz kojeg će se izvršavati sve ostale naredbe, to jest postaje pretpostavljeni direktorij prilikom rada sa slikom.

Naredba *COPY* kopira datoteke i direktorije s lokalnog računala na sliku, prvo navodeći izvorni direktorij, a zatim odredišni direktorij na slici, s napomenom da je pretpostavljeni direktorij na slici onaj koji je specificiran naredbom *WORKDIR*, pa će u ovom slučaju prilikom korištenja naredbe *COPY* i specificiranja odredišnog direktorija *'.'* datoteke biti kopirane u sliku u direktorij */honey*.

U zadnjoj liniji datoteke nalazi se komentar koji specificira naredbu kojom je na-

mjenjeno izgraditi sliku. Ona će izgraditi sliku s imenom *imunes/honeyplc* koristeći upravo datoteku *Dockerfile* iz tekućeg direktorija. Zadnji parametar predan naredbi *docker build* zove se kontekst izgradnje (engl. *build context*) i predstavlja direktorij u kojemu se nalaze sve datoteke koje će se kopirati na sliku, a u ovom slučaju to je upravo direktorij u kojemu se nalazi datoteka *Dockerfile*. To efektivno znači da će datoteke lokalnog stroja koje se nalaze kao prvi parametar naredbe *COPY* u datoteci *Dockerfile* biti tražene u direktoriju *build context*, što je u ovom slučaju tekući direktorij ('.').

4.3.4. Pokretanje

Prije pokretanja potrebno je modificirati predložak datoteke koja se predaje programu Honeyd koja definira IP adrese koje će u kontekstu ovog rada biti adrese na kojima će se nalaziti PLC uređaj i definira se *personality file*, što je ime *fingerprint* potpisa koji je prethodno kopiran u datoteku */usr/share/honeyd/nmap-os-db*. U ovom slučaju njegov naziv je *Siemens Simatic 300 programmable logic controller*, što je vidljivo u datoteci *honeyplc/plc-profiles/Siemens S7-300/s7-300-nmap-fingerprint.txt*.

Program *s7commServer* moguće je pokrenuti zasebno od simuliranog PLC uređaja, odnosno zasebno od pokretanja Honeyd specificiranjem IP adrese na kojoj sluša promet, što u okviru ovog rada znači da može slušati na lokalnom *loopback* sučelju odnosno IP adresi 127.0.0.1, ili na adresi 10.0.0.10, IP adresi koja mu je dodjeljena u Imunes simulaciji. Pretpostavljeni port za komunikaciju snap7 protokolom je 102.

Ako se specificira nepostojeća adresa odnosno adresa koja računalu nije prirodjeljena, što može biti provjereno korištenjem naredbe *ifconfig*, program će prestati s radom dizanjem *segmentation fault* iznimke bez dodatnih poruka.

U konfiguracijskoj datoteci koja se predaje kao parametar Honeyd programu potrebno je stvoriti bazni sustav te kao podsustav dodati *s7commServer* izvršnu datoteku. Potrebno je još postaviti njegov (engl. *fingerprint*) te IP adresu koja je njemu dodjeljena.

```
sudo honeyd -d -f config.s7-300 10.0.0.0/24
```

Ispis 4.3.4.1: Pokretanje Honeyd specificiranjem konfiguracijske datoteke i lokalne mreže uređaja

Kako bi se pokrenuo emulator i izbjegao prethodno nastali *segmentation fault* problem, koji se događao prilikom pokretanja emulatora i znači da je program pokušao pristupiti memoriji kojoj nije smio pristupati, pokušano je pokretanje Honeyd sa konfiguracijskom datotekom koja pokreće program na *loopback* IP adresi odnosno *loopback*

sučelju te stvaranje posrednika (engl. *proxy*) na vanjsku adresu čvora po uzoru na rješenje navedenog problema (engl. *issue*) na GitHub repozitoriju HoneyPLC alata.[25]

Iako se *s7commServer* može nezavisno pokrenuti, nažalost problem s dizanjem *segmentation fault* iznimke nije razriješen. Prilikom pokretanja naredbe 4.3.4.1. odnosno pokretanja Honeyd *s7commServer* diže *segmentation fault* iznimku čiji uzrok nije otkriven.

```
 create base
add base subsystem "/usr/share/honeyd/s7commServer" shared
   restart

clone host1 base
set host1 personality "Siemens Simatic 300 programmable logic
   controller"

add host1 tcp port 102 proxy 127.0.0.1:102
bind 10.0.0.10 host1
```

Ispis 4.3.4.2: Datoteka config.s7-300

5. Podaci javnog izvora

Otvoreni podaci (engl. *open source information*) (OSINT) podaci su iz javnih izvora, te sadržavaju više podataka nego što se na prvi pogled čini. Otvoreni podaci mogu biti podaci o ljudima, organizacijama, događajima, lokacijama, proizvodima, uslugama i slično. Ako se do podatka ne dođe čitanjem privatne baze podataka neke tvrtke bez ovlaštenja, primjerice, ili nekim drugim ilegalnim putem, onda se radi o podatku javnog izvora.

Skupina otvorenih podataka skupljenih u logičku cjelinu sa širim značenjem od samih sirovih podataka naziva se obavještajnim podacima javnog izvora(engl. *open source intelligence*).[13] Otvoreni podaci relevantni su za stvaranje kvalitetnog mamca upravo zato što su javno dostupni i mogu se legalno koristiti, te tako mogu poslužiti kao izvor informacija o stvarnim sustavima koje mamac oponaša, odnosno napadačima služe kao prvotni izvor informacija o sustavu koji napadaju.[3]

Da bi napadač uopće mogao znati da stvarni sustav za upravljanje industrijskim procesima odnosno mamac postoji i može mu se pristupiti preko mreže, mora imati nekakav izvor informacija o njemu. Iako se do tih podataka može pristupiti ilegalnim putem, primjerice provaljivanjem u druge sustave koje znaju za ICS ili u osobno računalo osobe koja je zadužena za dio ICS-a i ima veliku razinu ovlasti u samom ICS-u i tu naći potrebne informacije o digitalnim koordinatama sustava, ima smisla za osnovno informiranje o operateru prijenosnog sustava iskoristiti već ionako javno dostupne podatke.

5.1. OSINT alati

U svijetu istraživanja podataka javnog izvora postoji mnoštvo alata za prikupljanje i analizu podataka. U nastavku će biti opisani neki od njih. U svrhu saznavanja što je javno dostupno o mrežama operatera prijenosnog sustava primjer operatera bit će HOPS, Hrvatski operater prijenosnog sustava.

5.1.1. theHarvester

Harvester (stilizirano: *theHarvester*) alat je za prikupljanje informacija koji je namjenjen u svrhu penetracijskog testiranja.[4] Napisan je u programskom jeziku Python te koristi više izvora za prikupljanje informacija. Alat prikuplja podatke kao što su imena, adrese elektroničke pošte, poddomene i IP adrese.

Osim što ga je moguće instalirati kao izvršni kod u Linux okruženju, moguće je i klonirati izvorni kod sa *GitHub* repozitorija i pokrenuti.

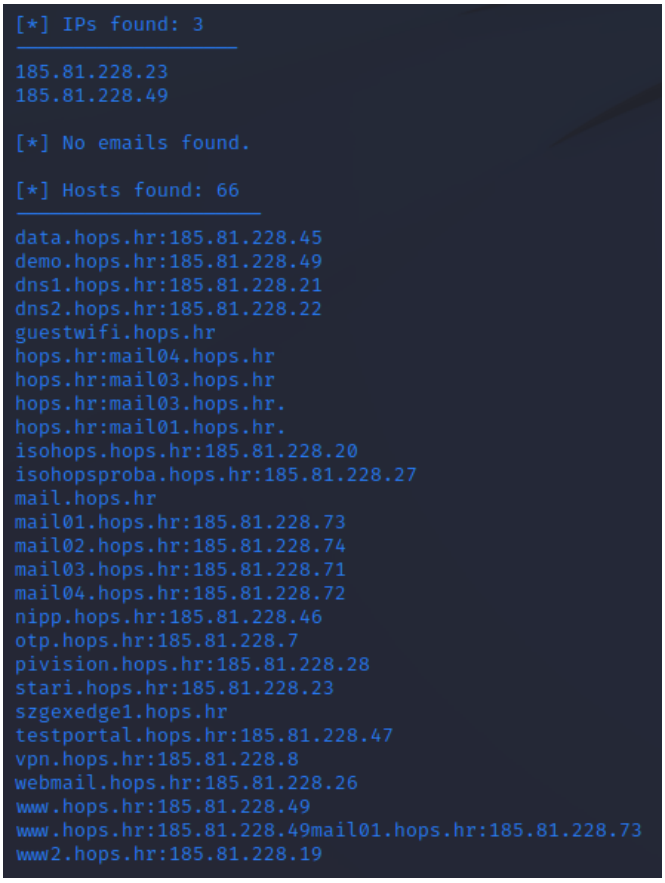
Nisu svi izvori podataka koje alat koristi odmah dostupni za uporabu, već je za dio njih potrebno imati korisnički račun na odgovarajućem servisu i tako dobiti pristup podacima i API ključ. Taj ključ moguće je unijeti u konfiguracijsku datoteku alata Harvester te će se onda i on koristiti za prikupljanje podataka.

Koristeći naredbu 5.1 pokreće se alat Harvester. Naredba pretražuje domenu hops.hr, ograniči izlaz na pet stotina rezultata i odabere sve dostupne izvore informacija. Nakon pokretanja Harvester će redom koristiti usluge koje koristi kao izvore i pokušati prikupiti informacije.

```
theHarvester -d hops.hr -l 500 -b all
```

Ispis 5.1.1.1: Naredba za pokretanje alata theHarvester

Na slici 5.1 prikazan je dio izlaza nakon pokretanja naredbe



Slika 5.1: Rezultat theHarvester upita

Izvršavanjem naredbe prikupljen je nezanemariv broj IP adresa (pridruženih domenama) što uključuje adrese DNS poslužitelja, adrese poslužitelja elektroničke pošte, adrese poslužitelja za web stranice, VPN poslužitelja i adrese odnosno domene koje su vjerojatno namjenjene internoj uporabi zaposlenicima. Također vidimo domene koje u sebi sadrže riječi *proba*, *test* i *stari*, što implicira postojanje potencijalno ranjivih servisa i podsustava koji su namjenjeni testiranju i razvoju, ili su pak zastarjeli ali su zbog nekog razloga još uvijek u pogonu.

U trenutku originalnog testiranja alata bilo je moguće pretraživati i platformu LinkedIn, koja je društvena mreža koja služi poslovnom umrežavanju. Ako se spremi rezultat narebe u datoteku imena *hops.log* i pokrene lanac naredbi

```
cat hops.log | grep --ignore-case -e manager -e lead -e
    administrator -e head | cut -d ' ' -f1-2 | sort | uniq |
    wc -l
```

Ispis 5.1.1.2: Naredba za filtriranje ispisa

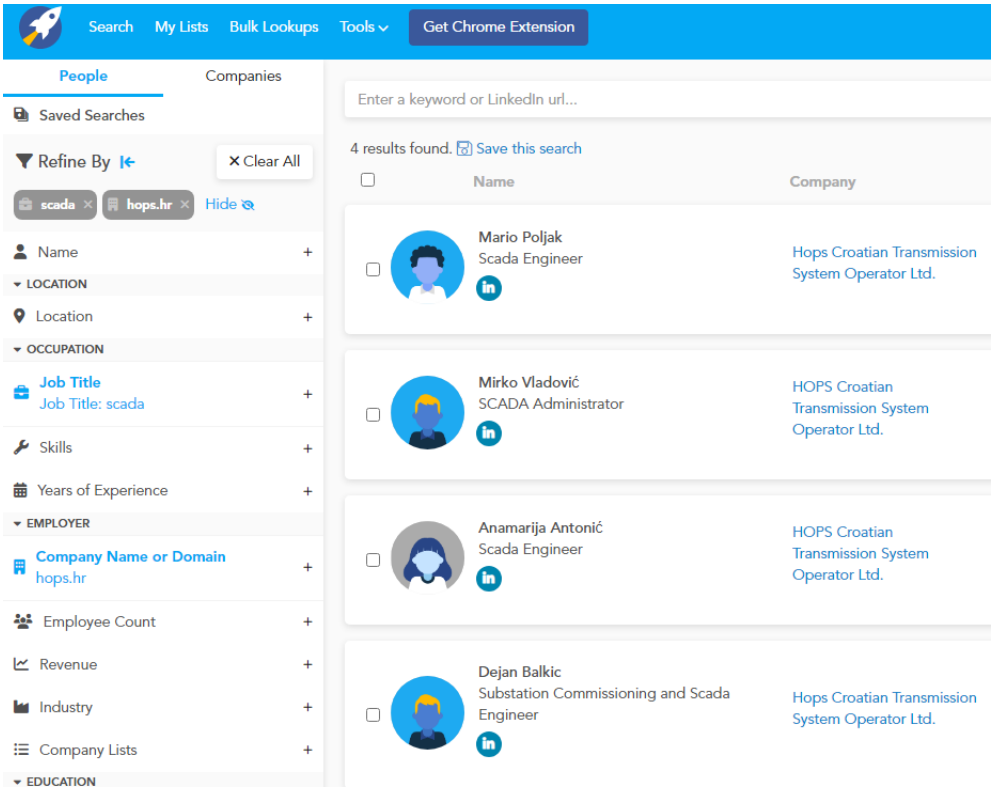
dobije se broj zaposlenika koji imaju neku od riječi (*manager, lead, administrator,*

head) u nazivu radnog mjesta. Micanjem zadnje naredbe u lancu (*wc -l*) dobije se popis zaposlenika koji imaju neku od navedenih riječi u nazivu radnog mjesta, odnosno vrlo je jednostavno saznati imena zaposlenika na visokim pozicijama. Nažalost, u trenutku pisanja ovog rada LinkedIn više nije dostupan kao izvor informacija zbog toga što Google blokira skripte, što znači da Harvester više ne može koristiti Google kako bi dobio podatke sa stranice LinkedIn.[7]

5.1.2. RocketReach

S nalaženjem zaposlenika i njihovih podataka koji rade u određenoj tvrtki ili na određenoj poziciji može pomoći platforma RocketReach. RocketReach je platforma koja služi pronalaženju adresa elektroničke pošte ili kontakt informacijama zaposlenika raznih tvrtki. Podatke dobiva iz javno dostupnih izvora, s internetskih stranica društvenih mreža, internetskih stranica tvrtki i javnih evidencija.[23] Činjenica da pretražuje društvene mreže znači da pretražuje i LinkedIn, čime se može zaobići problem koji je nastao s alatom Harvester i njegovim izravnim korištenjem Google pretraživača za pretraživanje platforme LinkedIn.

Primjerice, moguće je pretražiti zaposlenike čija domena tvrtke je hops.hr i koji u imenu radnog mjesta imaju riječ SCADA. Rezultat pretraživanja prikazan je na slici 5.2.



Slika 5.2: Rezultat RocketReach upita

Pritiskom na gumb *Get Contact Info* koji se nalazi desno od imena zaposlenika i imena tvrtke mogu se dobiti kontakt informacije gotovo svakog ovako pronađenog zaposlenika.

Uz dostupnu internetsku stranicu, RocketReach nudi i API koji se može koristiti za automatizirano pretraživanje, što iskorištava i alat Harvester. Tako je uz pretraživanje internetske stranice RocketReach moguće do informacija koje on prikuplja doći i koristeći alat Harvester, pod uvjetom da mu se preda RocketReach API ključ, te time možemo automatizirati pretraživanje i prikupljanje podataka sa stranica kao što je LinkedIn.

5.2. Prijedlozi za poboljšanje mamca javno dostupnim informacijama

Mamac bi trebao, uz samo postojanje na mreži, trebao imati prisutnu informacijsku okolinu vjerodostojnu onoj pravog sustava. Mamac bi trebao biti povezan sa DNS poslužiteljima stvarnih sustava ili bi trebao imati vlastite DNS poslužitelje koji bi davali vjerodostojne informacije o mreži mamca, tako da implicira da je povezan s operate-

rom prijenosnog sustava.

Također, moguće je kreiranje lažnih profila zaposlenika koji bi bili povezani s mamcem sa stvarnim adresama elektroničke pošte. Za slučaj da napadač pokuša kontaktirati zaposlenika, u kontroliranoj okolini moguće je simuliranom greškom otvarati sumnjivu elektroničku poštu gdje se napadač lažno predstavlja i šalje zloćudni kod, u cilju ugrožavanja mamca.

6. Zaključak

Mamci koji oponašaju stvarne servise i podsustave od velike su važnosti u području sigurnosti, jer ne samo da otežavaju pronalazak stvarnih sustava već omogućavaju operaterima stvarnih sustava učenje o napadačima i njihovim metodama.

Postoji mnoštvo dostupnih mamaca različitih razina složenosti, od onih koji su jednostavni za postavljanje i korištenje do onih koji zahtjevaju veliku količinu vremena i resursa za postavljanje i održavanje. Kako bi se odabrali najbolji mamci za određenu primjenu potrebno je uzeti u obzir mnoge čimbenike, najvažniji od kojih je odabir mamca koji najbolje oponaša ciljani, stvarni sustav.

Iako postoji mnoštvo dostupnih mamaca, potrebna je značajna količina truda i znanja za njihovo postavljanje. Neke od najboljih biblioteka namjenjene izgradnji mamaca zastarjele su i više se ne održavaju, što otežava njihovo korištenje. Iako se postavljanje mamca čini jednostavnim, iza instalacije svakog alata kriju se potencijalni problemi koji mogu ugroziti i naizgled najjednostavnije instalacije.

Informacije javnog izvora od velike su važnosti te mogu otkriti puno informacija o tvrtki i njezinim zaposlenicima. Napadači ih koriste za prikupljanje informacija o tvrtki i njezinim zaposlenicima, što je bitno iskoristiti prilikom izrade mamca kako bi se postigla što veća vjerodostojnost stvarnom sustavu.

LITERATURA

[1] HOPS - O nama. https://www.hops.hr/o-nama, 2023. Pristupljeno: 21.5.2023.

[2] EntsoE Event. https://www.entsoe-event.eu/transmission_system_operator.html, 2023. Pristupljeno: 21.5.2023.

[3] Babak Akhgar, P. Saskia Bayerl, i Fraser Sampson. *Open source intelligence investigation : investigation from strategy to implementation*. Advanced sciences and technologies for security applications. Springer, Cham, Switzerland, 2016. ISBN 3319476718.

[4] Matthew Brown, Jay Townsend, Lee Baird, Christian Martorella, et al. theHarvester. https://github.com/laramies/theHarvester, 2009.

[5] Goran Cetušić, Miljenko Mikuc, Denis Salopek, Valter Vasić, Marko Zec, Nikola Đurak, Ana Kukec, Sanja Vasić, Ana Lipničan, i Zrinka Puljiz. IMUNES. http://imunes.net/, 2004. Pristupljeno: 22.5.2023.

[6] Cymmetria. Cisco ASA honeypot. https://github.com/Cymmetria/ciscoasa_honeypot, 2018. Cymmetria Research.

[7] DayNja. theHarvester Issue 1242. https://github.com/laramies/theHarvester/issues/1242, 2022.

[8] Docker. Docker - Build, Share, and Run Any App, Anywhere. https://www.docker.com/. Pristupljeno 15.7.2023.

[9] Donald Krambeck. An Introduction to SCADA Systems. https://www.allaboutcircuits.com/technical-articles/an-introduction-to-scada-systems/, Kolovoz 2015. Pristupljeno: 21.5.2023.

[10] Efrén López and SEFCOM. HoneyPLC: PLC Honeypot. https://github.com/sefcom/honeyplc/, 2021.

[11] Kevin Hemsley i Dr. Ronald E. Fisher. History of industrial control system cyber incidents. 2018.

[12] IBM. Docker. https://www.ibm.com/topics/docker, 2021. Pristupljeno: 22.5.2023.

[13] SANS Institute. What is open source intelligence? *SANS Institute Blog*, N/A. Pristupljeno: 1.6.2023.

[14] Dr. Ronald E. Fisher Kevin E. Hemsley. Analysis of the cyber attack on the Ukrainian power grid. Technical report, Electricity Information Sharing and Analysis Center (E-ISAC), 2016.

[15] Kim Zetter. Everything We Know About Ukraine's Power Plant Hack. https://www.wired.com/2016/01/everything-we-know-about-ukraines-power-plant-hack/, Siječanj 2016. Pristupljeno: 21.5.2023.

[16] Božo Kopić, Jakov Krstulović Opara, Zlatan Sinčanica, i Adam Trstenjak. HAT-GUI. https://hat-gui.hat-open.com/server.html, . Pristupljeno 20.4.2023.

[17] Božo Kopić, Jakov Krstulović Opara, Zlatan Sinčanica, i Adam Trstenjak. HAT-Monitor. https://hat-monitor.hat-open.com/monitor.html, . Pristupljeno 20.4.2023.

[18] Božo Kopić, Jakov Krstulović Opara, Zlatan Sinčanica, i Adam Trstenjak. Hat-open. http://hat-open.com/, 2020 - 2023.

[19] Efrén López-Morales, Carlos Rubio-Medrano, Adam Doupé, Yan Shoshitaishvili, Ruoyu Wang, Tifanny Bao, i Gail-Joon Ahn. Honeyplc: A next-generation honeypot for industrial control systems. U *In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, Nov 2020.

[20] Michel Oosterhof. Cowrie. https://github.com/cowrie/cowrie, 2015. Pristupljeno: 31.5.2023.

[21] National Vulnerability Database. Cve-2010-2772 siemens. https://nvd.nist.gov/vuln/detail/CVE-2010-2772, 2010. Pristupljeno 21.5.2023.

[22] Niels Provos. Honeyd. https://www.honeyd.org/, 1999 - 2023.

[23] RocketReach. RocketReach. https://rocketreach.co/about_us, 2022.

[24] Telekom Security. TPot Community Edition. https://github.com/telekom-security/tpotce, 2016.

[25] Francesco Trolese. Issue 2. https://github.com/sefcom/honeyplc/issues/2, 3 2021. Pristupljeno 5.6.2023.

Poboljšanje kvalitete mamaca za operatora prijenosnog sustava temeljem javno dostupnih podataka

Sažetak

Mamci su računalni sustavi koji se koriste za privlačenje napadača s ciljem učenja o njihovim metodama.

U radu su opisani ključni pojmovi vezani uz operatere prijenosnih sustava i industrijske upravljačke sustave i dani opisi nekoliko mamaca. Raniji prototip mamca nadograđen je boljim grafičkim sučeljem i neuspješno je dodan čvor koji simulira programibilni logički kontroler. Istražen je pojam javno dostupnih informacija, predstavljeni alati za istraživanje javno dostupnih informacija i dani prijedlozi poboljšanje mamca korištenjem javno dostupnih podataka.

Postoji mnoštvo dostupnih mamaca otvorenog koda različitih razina složenosti. U mrežu je dodan HoneyPLC mamac koji koristi nekoliko zastarjelih biblioteka koje su otežale proces instalacije mamca, i zbog greške u kodu nepoznatog izvora dodani čvor nije funkcionalan.

Ključne riječi: Mamac, operater prijenosnog sustava, industrijski upravljački sustav, javno dostupne informacije, PLC

Raising the quality of a honeypot system for a transmission system operator based on publicly available data

Abstract

Honeypots are computer systems used for attracting attackers with the goal of learning about their methods.

The paper describes key concepts related to transmission system operators and industrial control systems and gives descriptions of several honeypots. An earlier prototype of a honeypot was upgraded with a better graphical interface and an unsuccessful attempt was made of adding a node that simulates a programmable logic controller. The concept of publicly available data was explored, tools for exploring publicly available data were presented and suggestions for the improvement of the honeypot based on publicly available data were given.

There are many open source honeypots of different levels of complexity. The HoneyPLC honeypot was added to the network, which uses several outdated libraries that made the process of in stalling the honeypot more difficult, and due to an error in the code of unknown origin, the added node is not functional.

Keywords: Honeypot, transmission system operator, industrial control system, open source intelligence, PLC