SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 185

PODEŠAVANJE I ISPITIVANJE VATROZIDA NA TEMELJU ZADANIH POLITIKA VISOKE RAZINE APSTRAKCIJE

Marta Gračner

Zagreb, lipanj 2023.

Sadržaj

Uvod

U digitalnom dobu u kojem danas živimo, sigurnost sustava postaje sve popularnija tema. Porastom broja prijetnji i napada na organizacije i pojedince, raste i potreba za zaštitom sustava. Mnogi izvori dokumentiraju sve češće napade i kompromitacije na velike organizacije koje prenose i velike medijske kuće. Međutim, napadači nisu usmjereni isključivo na poznate korporacije već se velik dio incidenata odnosi i na manje tvrtke koje su najčešće napadnute nekom vrstom ransomware-a. Stoga je važno uočiti velik utjecaj zaštite vlastitih mrežnih sustava.

Napretkom tehnologije, na tržište dolaze razna nova rješenja koja zamjenjuju zastarjela rješenja za zaštitu mrežnih sustava. Međutim, neki sigurnosni uređaji ostaju prisutni od samih početaka, uz relativno male promjene i nadogradnje. Jedan od takvih uređaja je vatrozid koji može učinkovito otkloniti većinu jednostavnih pokušaja proboja perimetra mrežnog sustava.

Postavljanje vatrozid pravila kojima se efektivno filtrira promet u velikim sustavima može biti kompleksan posao sklon pogreškama. Osobe zadužene za postavljanje vatrozida moraju biti upoznate sa svim tehničkim aspektima mrežnog prometa. Također moraju biti vješte u transformaciji sigurnosnih politika koje je odredila organizacija u vatrozid pravila koja će odgovarati tim politikama i na adekvatan način štititi mrežu od neautoriziranog pristupa. Izvješće tvrtke Verizon iz 2022. godine tvrdi da su ljudske pogreške odgovorne za čak 13% sigurnosnih propusta tijekom zadnjih nekoliko godina [17]. Tako veliki postotak svjedoči o značajnom utjecaju ljudskog elementa u kreiranju konfiguracija raznih uređaja u mrežnom sustavu te o ogromnom riziku vezanom za ručno podešavanje.

S ciljem olakšavanja konfiguriranja vatrozida administratorima sustava i ostalim zainteresiranim pojedincima, razvijen je alat za automatizirano prevođenje sigurnosnih pristupnih politika u vatrozid pravila. Alatu je svrha minimizirati broj pogrešaka u konfiguraciji vatrozid pravila. Osim toga, razvijen je i alat za automatizirano ispitivanje ispravnosti generiranih pravila kako bi se olakšala njihova validacija i ranije uočavanje nepravilnosti, bez potrebe za učitavanjem pravila na fizički mrežni sustav. Navedeni alati trebali bi značajno smanjiti rizik pogrešnih konfiguracija i doprinijeti sigurnosti mrežnog

sustava te potencijalno informirati i educirati administratore sustava o pravilnom pisanju pravila i podešavanju vatrozida.

Prvo poglavlje detaljnije će objasniti što je vatrozid, čemu služi i što sve može obuhvaćati nakon čega slijedi kratki pregled problematike prevođenja i razvrstavanja vatrozid pravila. U trećem poglavlju bit će opisane dvije tehnologije korištene u sklopu ovog diplomskog rada, a to su iptables i IMUNES. Sljedeće poglavlje opisuje razvijene alate, njihovu svrhu, strukturu i izlazne podatke. U posljednja tri poglavlja bit će izneseni dobiveni rezultati, diskusija te ideje za buduće nadogradnje alata, nakon čega slijedi zaključak, a na samom kraju je popis korištene literature.

1. Vatrozid

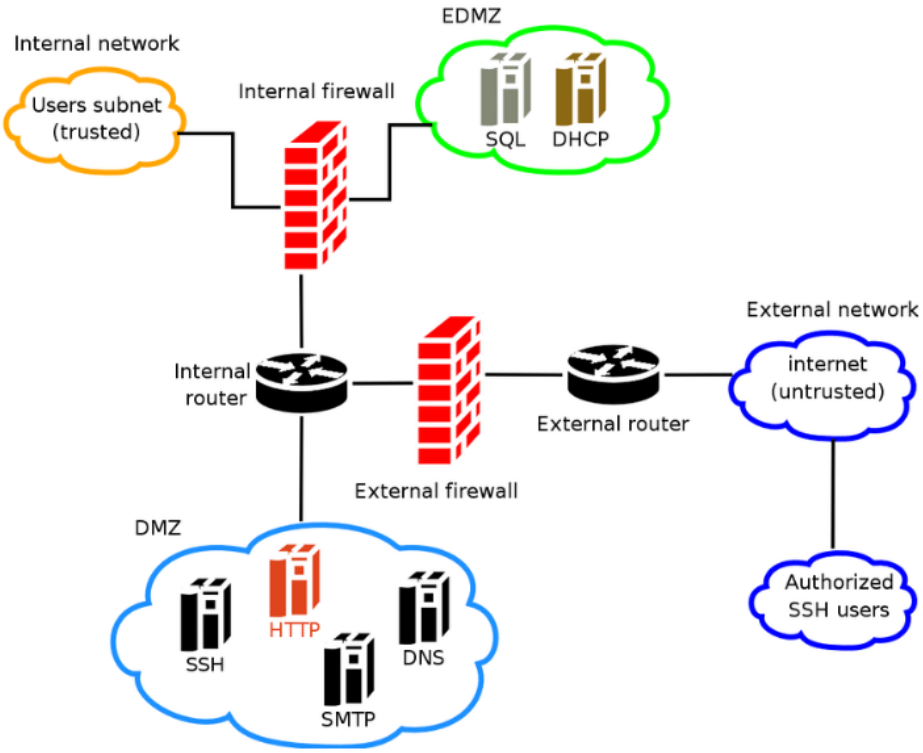
Svi podaci koji se žele razmijeniti putem mreže između dva ili više računala moraju biti formirani zajedno s odgovarajućim pratećim podacima u mrežne pakete. Prateći podaci određuju put kojim će se paketi slati te način na koji će se raspakirati i objediniti u cjelinu na samom odredištu [1]. Skup svih paketa koji se u nekom trenutku kreću kroz mrežu može se opisati pojmom mrežni promet. Postoji niz uređaja čija je svrha nadgledanje ili manipulacija mrežnim prometom, a jedan od njih je vatrozid.

Pravila vatrozida proizlaze iz pristupnih politika koje definira organizacija, gdje svaka politika na apstraktnijoj razini definira objekte koji međusobno mogu ili ne smiju komunicirati. Objekt se u ovom kontekstu odnosi na računalo i softver koji se na njemu nalazi. Politike se mogu definirati na nekoliko razina apstrakcije, od opisa korištenjem prirodnog jezika pa sve do opisa pomoću domenskih jezika i onih temeljenih na grafovima.

Vatrozid je uređaj, programska podrška ili softver-kao-usluga koji se koristi za upravljanje mrežnim prometom na temelju unaprijed zadanih pravila. Pravila za vatrozide unutar organizacija definiraju se na temelju pristupnih politika koje uspostavlja pojedina organizacija. Vatrozid se postavlja na granice mrežnih segmenata i na samu granicu između cjelokupne mreže pojedine organizacije i interneta, kao što je vidljivo na slici (Slika 1.1)[2]. Slika prikazuje vatrozid „External firewall“ koji dijeli internu mrežu organizacije od ostatka interneta prikazanog oblačićem „External network“. Vatrozid naziva „Internal firewall“ stavljen je na granicu triju mrežnih segmenata naziva „DMZ“, „EDMZ“ i „Internal network.“ te time kontrolira pristupanje pojedinim čvorovima unutar tih segmenata.

Vatrozid je prva linija obrane protiv napadača koji pokušavaju prodrijeti u mrežu organizacije te je i dan danas sastavni dio mreže i ključni uređaj za osiguravanje sigurnosti mreže. On omogućava filtriranje prometa koji prolazi kroz njega tako da dopušta isključivo mrežne pakete koji zadovoljavaju definirana vatrozid pravila, dok odbacuje sve ostale. To se u većini slučajeva koristi s namjerom sprječavanja malicioznog prometa i onemogućavanja jednostavnih pokušaja neovlaštenog ulaska i kretanja kroz mrežu [2]. Postavljanjem nekoliko vatrozida na različita mjesta unutar mreže postiže se segmentacija, čime se omogućava detaljnije filtriranje prometa te zaštita od potencijalnih unutarnjih prijetnji. Vatrozid se također koristi i u svrhu kontrole i ograničavanja pristupa pojedinim uređajima unutar mreže. Time se postiže jedno od glavnih pravila sigurnosti, a to je

razdvajanje nadležnosti. Zapisivanje detalja o dolaznim paketima u dnevničke datoteke omogućava naknadnu analizu te, u slučaju incidenata, pruža uvid u neželjene pristupe određenim objektima unutar mreže [3].



Slika 1.1 Primjer mreže s vatrozidima [7]

S obzirom na objekte koje štite, vatrozidi se dijele na mrežno bazirane, to jest one koji štite cjelokupne mreže te su najčešće hardverski izvedeni, i računalno bazirane, odnosno softverski izvedene, koji štite pojedina računala na kojima se nalaze. Filtracijska metoda je drugi kriterij podjele vatrozida, te se prema tome dijele na:

- vatrozide za filtriranje paketa (*Packet-filtering firewalls*)
- vatrozide s praćenjem povezanosti paketa (*Stateful inspection firewalls*)
- vatrozide s praćenjem TCP rukovanja (*Circuit-level gateway firewalls*)
- vatrozide na aplikacijskom sloju (*Proxy firewalls*)
- vatrozide sljedeće generacije (*Next-generation firewalls*)
- vatrozide sljedeće generacije usmjerene na prijetnje (*Threat-focused NGFWs*)
- virtualne vatrozide (*Virtual firewalls*)
- vatrozide u oblaku (*Cloud-native firewalls*) [3]

Vatrozidi s praćenjem povezanosti paketa, poznati i pod nazivom vatrozidi s dinamičkim filtriranjem paketa, uspoređuju dolazni i odlazni promet s definiranim pravilima uzimajući u obzir izvorišne i odredišne IP adrese, odredišni priključak i protokol, te stanje konekcije koje je zapisano u internu tablicu otvorenih konekcija. Zadana pravila dijele se u disjunktne skupove ovisno o tome evaluiraju li se nad paketima koji pripadaju novim ili već uspostavljenim konekcijama [3].

Kao i svaki drugi uređaj koji je dio mreže, vatrozid ima svoje slabosti i ranjivosti koje se mogu iskoristiti i na taj način kompromitirati mrežni sustav. Neke od popularnijih prijetnji u posljednje vrijeme, od kojih se teže zaštiti bez dodatnih metoda i tehnologija, uključuju unutarnje napade, (distribuirane) napade uskraćivanjem usluge, infekciju zlućudnim kodom, sporu implementaciju ažuriranja i lošu konfiguraciju pravila [3]. Adekvatna konfiguracija pravila za vatrozide postaje sve teža zbog rastuće kompleksnosti i veličine mreže uređaja koja čini sustav pojedinih organizacija.

U današnje vrijeme, uz samu funkcionalnost filtriranja prometa na temelju predodređenih kriterija, u vatrozid uređaje ugrađuju se dodatne funkcionalnosti s ciljem povećanja zaštite i detekcije napada. Neki od spomenutih funkcionalnosti su sustavi za detekciju i prevenciju upada (IDS/IPS) te virtualne privatne mreže (VPN) koje rade na principu stvaranja kriptiranog tunela između izvora i odredišta kroz javnu mrežu [2]. S ciljem smanjenja rizika od prijetnji spomenutih u prethodnom paragrafu sve češće se koriste vatrozidi nove generacije s ugrađenim sustavom za detekciju prijetnji, filtriranje URL-ova ovisno o geolokaciji, kontrolom rizičnih aplikacija i sličnim metodama prevencije i ublažavanja napada. Nadogradnja na potonju vrstu vatrozida su vatrozidi nove generacije usmjereni na prijetnje, čija je zadaća detekcija i otklanjanje prijetnji analizom konteksta i praćenjem objekata koji su najviše izloženi riziku od napada te identificiranjem prometa na aplikacijskoj razini [4].

2. Prevođenje i razvrstavanje vatrozid pravila

Prednost opisivanja sigurnosnih politika višom razinom apstrakcije i automatsko prevođenje istih u vatrozid pravila je lakše razumijevanje i održavanje politika i njihovih pripadnih pravila. Time se omogućuje fokusiranje na definiciju pristupnih politika i identificiranje kontrole pristupa koja se želi postići umjesto fokusiranja na tehničke detalje. Jedan od preglednih radova koji se bavi problematikom prevođenja apstraktnih sigurnosnih politika u vatrozid pravila je rad Systematic review of automatic translation of high-level security policy into firewall rules [18].

U radu se analizira 23 rada koji opisuju 17 različitih pristupa prevođenju apstraktnih politika. Metode su prvo klasificirane ovisno o razini apstrakcije, postojanju predefinirane baze znanja ili konverzije pravila, postojanju grafičkog korisničkog sučelja (GUI), tipu generiranih pravila, konceptu prema kojem su definirane politike i postojanju studije upotrebljivosti [18].

Dobiveni rezultati ukazuju na to da se prikupljeni pristupi najčešće fokusiraju na prevođenje politika zadanih sintaksom inspiriranom programskim jezicima ili u XML formatu odnosno politike definirane na nižoj razini apstrakcije. Većina pristupa ne uključuje ili uključuje parcijalnu bazu znanja, trećina rješenja uključuje GUI, a pola od analiziranih rješenja uključuje studiju upotrebljivosti [18].

Glavni nedostaci pristupa uključuju potrebu konstantnog ažuriranja baze znanja i kompleksnost definiranja i održavanja sigurnosnih politika [18].

Nakon samog generiranja vatrozid pravila iz zadanih sigurnosnih politika javlja se problem raspoređivanja dobivenih pravila na vatrozid uređaje koji se nalaze u mreži. Populiranje vatrozida pravilima je proces podložan pogrešakama ako se radi ručno međutim ovaj proces je relativno jednostavno automatizirati.

Jedan od radova koji se bave automatizacijom potonjeg procesa je rad Automatic Firewalls' Configuration using Argumentation Reasoning [16]. Navedeni rad predstavlja alat koji na automatiziran način konfigurira vatrozide generirajući konfiguracije bez sukoba s ciljem umanjenja ljudskih pogrešaka tijekom konfiguracije pravila i vatrozida, što je veoma slično temi kojom se bavi ovaj diplomski rad. Znanstveni rad koristi proces formalne verifikacije pomoću formalnog radnog okvira ArgoFiCo [16].

Rješenje se sastoji od tri modula od kojih je prvi zadužen za identifikaciju potencijalnih anomalija kao što su redundantnost, irelevantnost i sukobi između pravila te primjenu strategije za razrješavanje anomalija brisanjem ili prioritetiziranjem pravila. Drugi modul generira konfiguraciju za svaki od vatrozida na način da razmješta pravila više razine apstrakcije na vatrozid uređaje, a treći modul prevodi konfiguraciju u tehnički specifične detalje kao što su priključnice i IP adrese [16].

3. Korištene tehnologije

U sklopu ovog diplomskog rada korišteno je nekoliko različitih alata i tehnologija, pri čemu je najveći naglasak stavljen na iptables i IMUNES. Od ostalih tehnologija za podešavanje vatrozida upogonjen je program Uncomplicated Firewall (ufw). Za implementaciju alata korišten je programski jezik Python uz razne biblioteke jedna od kojih je biblioteka `ipaddress` koja je zaslužna za generiranje IP adresa iz zadanih raspona. Tehnologije su izabrane jer su besplatne, jednostavne za korištenje, imaju sve potrebne funkcionalnosti te se zajedno upotpunjuju i zadovoljavaju sve potrebne uvjete.

3.1. Iptables

Iptables je administracijski alat naredbenog retka koji dolazi unaprijed instaliran na većini distribucija operacijskog sustava Linux. Koristi se unutar jezgre Linuxa za konfiguraciju vatrozida te ne podržava IPv6 adrese, što znači da se može koristiti isključivo za podešavanje IPv4 adresa [5]. Služi za postavljanje i održavanje tablica prema kojima se filtrira i kontrolira mrežni promet vezan za računalo na kojem se iptables nalazi. Zahvaljujući tim tablicama svaki mrežni paket koji prođe kroz vatrozid će biti pregledan i uspoređen s vatrozid pravilima te prema tome proslijeđen ili odbačen [6].

Postoji pet predefiniranih tablica koje se sastoje od niza lanaca. Prva tablica se naziva *raw* te se koristi za označavanje paketa koje se ne želi pratiti. *Nat* tablica se koristi za prevođenje mrežnih adresa, tablica imena *mangle* služi za "specijalizirane izmjene paketa", a za postavljanje mrežnih pravila Mandatne kontrole pristupa (Mandatory Access Control) se koristi *security* tablica. Posljednja tablica se zove *filter* tablica u kojoj se najčešće smještaju sva pravila vezana uz vatrozide [5]. Konfiguracija mreže i učitani moduli odlučuju koje tablice će biti prisutne u jezgri [6].

Paket se uspoređuje s vatrozid pravilima redom kako se one nalaze u lancu. Kada paket zadovoljava uvjete jednog od pravila, prestaje prolaženje po lancu te se aktivira akcija zadana tim pravilom. U slučaju da se paket ne može povezati s nijednim pravilom odabire se akcija zadana generalnom politikom koja može biti ili propuštanje ili odbacivanje paketa.

Kao što je ranije spomenuto tablice se sastoje od lanaca koji sadrže listu pravila koja je u početku prazna sve dok korisnik pomoću naredbenog retka ili jednog od nekoliko dostupnih

grafičkih sučelja ne umetne pravila. Svaka od zadanih tablica obavezno sadrži predefinirane lance, a može sadržavati i korisničke lance.

Prilikom definiranja pojedinog pravila potrebno je definirati i akciju (*target*) koja određuje što će se dogoditi s paketom ako zadovoljava uvjete pravila. Akcija može biti korisnički lanac, u tom slučaju se usporedba paketa nastavlja vršiti s pravilima korisničkog lanca, a u slučaju da ni ovdje ne dođe do poklapanja, usporedba se nastavlja s listom pravila u prethodnom lancu. Osim korisničkih lanaca akcije mogu biti ugrađene akcije (built-in target) i ekstenzije akcija (target extensions) u koje su uključene *REJECT* i *LOG*.

Postoje četiri ugrađene akcije, a to su *ACCEPT* koja propušta paket, *DROP* koja odbacuje paket kao da nikada nije ni postojao, *QUEUE* prosljeđuje paket u korisnički prostor i *RETURN* što označava prestanak procesiranja u trenutnom lancu i povratak na prethodni. Ako paket odgovara uvjetima pravila čija je akcija jedna od ugrađenih, prekida se procesiranje te se izvodi odgovarajuća akcija.

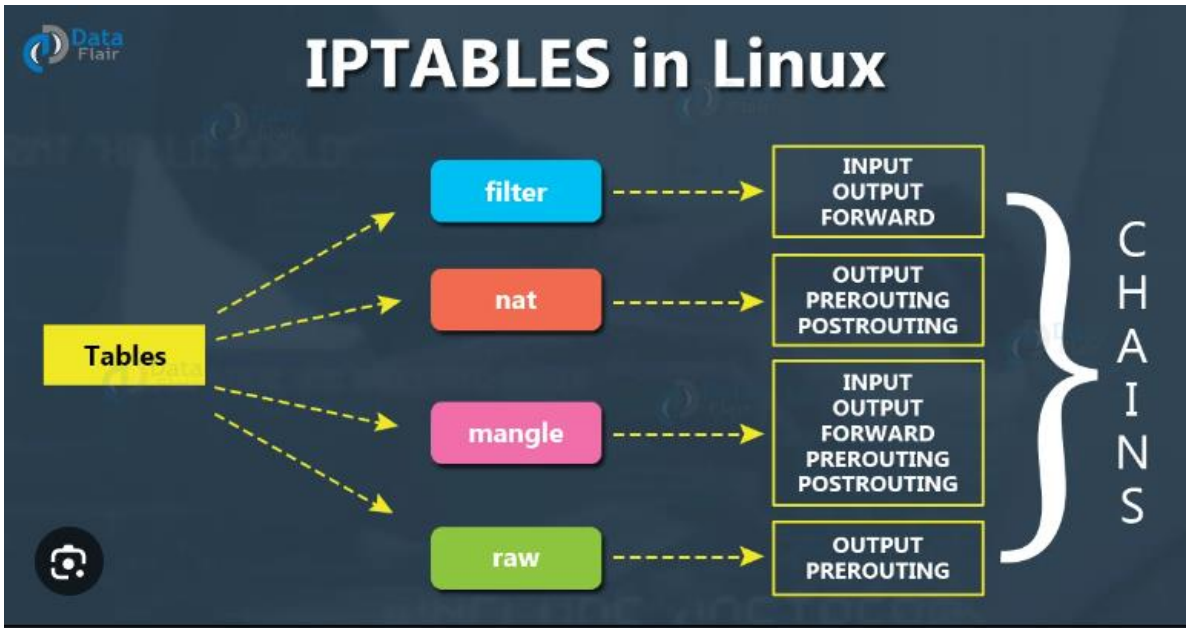
Filter tablica sadrži tri predefinirana lanca: *INPUT, OUTPUT* i *FORWARD* [5]. INPUT lanac sadrži pravila koja se odnose na sve dolazne pakete kojima je odredište računalo na kojem se nalazi vatrozid. Svi paketi koji prolaze kroz vatrozid, ali imaju izvorište i odredište negdje drugdje u mreži uspoređuju se s pravilima u FORWARD lancu, dok svi odlazni paketi prolaze kroz pravila OUTPUT lanca. Za sve pakete koji su dio dvosmjerne konekcije, kako bi se komunikacija mogla odvijati, potrebno je definirati vatrozid pravila za oba smjera. Srećom iptables prati stanje konekcije te se dodatno mogu definirati akcije za sve otvorene konekcije.

Vizualizacija navedenih tablica i njihovih pripadajućih predefiniranih lanaca prikazana je na slici (Slika 3.1).

Iptables tehnologija ima mnogo opcija za definiranje vatrozid pravila. Jedan osnovan primjer konfiguracije vatrozid pravila pomoću iptables tehnologije slijedi u nastavku:

```
iptables -I INPUT -s 10.0.0.0 -j DROP
```

Ovaj primjer definira vatrozid pravilo koje odbacuje sve dolazne pakete s izvorištem u IP adresi 10.0.0.0., točnije parametrom `-I` se označava da će se pravilo umetnuti na početak lanca koji je zadan neposredno nakon parametra, u ovom slučaju se radi o lancu INPUT. Parametar `-s` označava da će se neposredno nakon njega zadati IP adresa odredišta paketa, te naposljetku, parametrom `-j` se zadaje akcija na koju se "skače" u slučaju da paket zadovolji ostale uvjete pravila [8].



Slika 3.1 Pregled tablica i lanaca s kojima raspolaže iptables alat [9]

Od ostalih zanimljivijih parametara, mogu se izdvojiti sljedeći: `-d` nakon čega slijedi IP adresa te oni zajedno označavaju odredište paketa, slično tome zastavica `-o` označava ime izlaznog sučelja kroz koje će se proslijediti paket te će se ta dva parametra rijetko kada naći zajedno u istom pravilu. Suprotno tome, ako se želi zadati ulazno sučelje kroz koje će paket doći u vatrozid, koristit će se zastavica `-i`. Još jedan zanimljiv parametar koristi se za definiciju protokola transportnog sloja koji se koristi za slanje paketa, a zadaje se zastavicom `-p` [8].

Osim navedenog, iptables parametri nude pregršt konfiguracijskih opcija, a u nastavku slijedi kratki pregled osnovnih postavki. Svako pravilo se može dodati u lanac na kraj ili početak, može se izbrisati, izlistati ili izmijeniti isto kao i pojedini lanci. Može se zadati nekoliko razina bilježenja dnevničkih zapisa za svaki od lanaca, te se mogu zadati specifične izvorišne i odredišne priključnice [8].

3.2. IMUNES

Simulatori i emulatori mreže su veoma korisni alati kada se želi eksperimentirati s raznim topologijama mreže ili se educirati o njenom funkcioniranju bez da je potrebno uložiti mnogo resursa u postavljanje fizičke mreže što osobito nije optimalno ako nije unaprijed poznato kako će krajnji raspored mreže izgledati. IMUNES (Integrated Multiprotocol Network Emulator/Simulator) je emulator i simulator mreže otvorenog koda koji se koristi

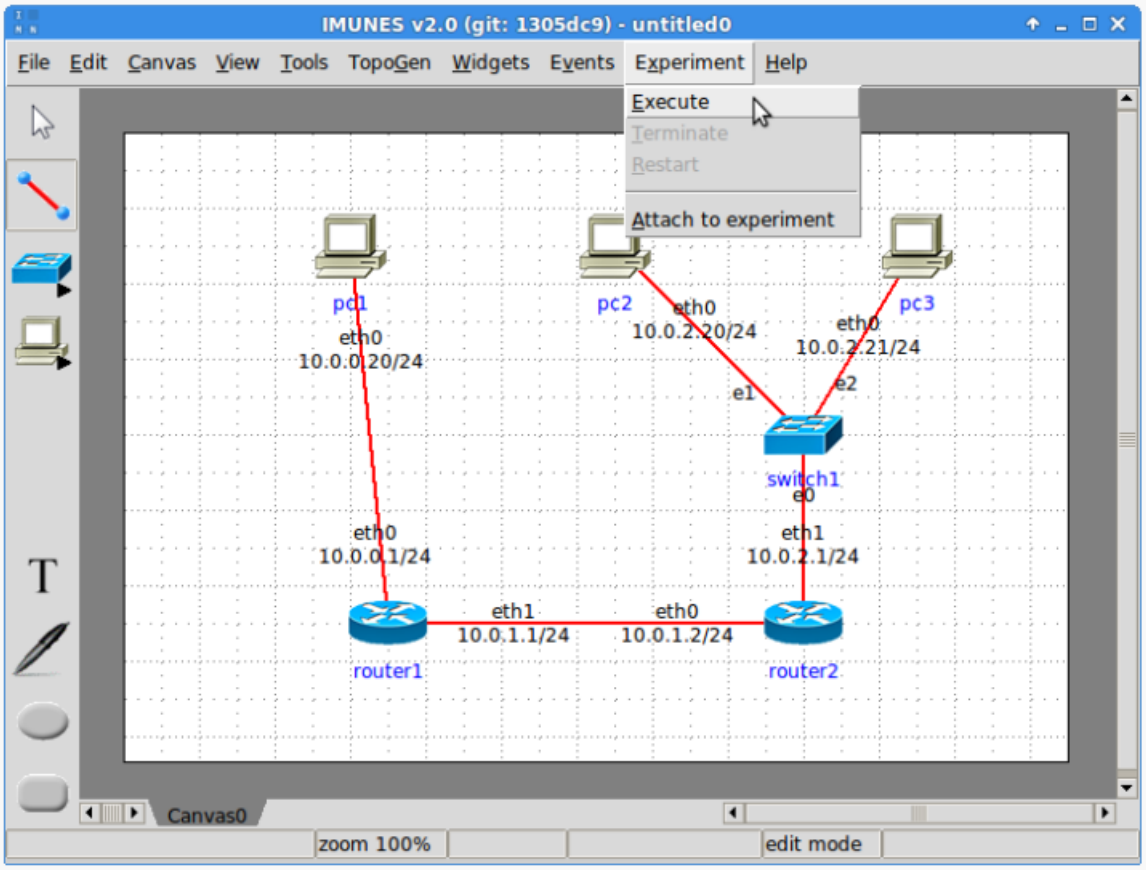
u svrhu edukacije i eksperimentiranja s mrežom i njenim segmentima, uređajima, protokolima i ostalim mrežnim artefaktima tako što se na temelju zadane topologije mreže pokrenu virtualni čvorovi koji su formirani kroz posebne modifikacije jezgre ako se nalaze na FreeBSD operacijskom sustavu ili Docker spremnike ako su na Linuxu [10]. IMUNES je jedan od rijetkih rješenja otvorenog koda koje može simulirati mrežu i promet nad velikim brojem čvorova te je zbog toga koristan tijekom edukacije, istraživanja i testiranja mrežne topologije kao i proučavanja ponašanja mreže i pojedinih čvorova u raznim scenarijima.

IMUNES korisniku omogućava konfiguraciju IPv4 i IPv6 adresa, MAC adresa, IPsec-a i priključnica svakog pojedinog čvora kojih može biti čak više stotina. Simulacija se pokrene u tek nekoliko sekundi, a svi virtualni čvorovi se ponašaju kao što bi se ponašali i fizički čvorovi i to u realnom vremenu [11]. Osim potpuno funkcionalnog naredbenog retka, na čvorovima se mogu instalirati i koristiti razni alati poput Wiresharka koji služi za nadziranje i analizu prometa, internetskog preglednika i klijenta elektroničke pošte. Osim toga može se koristiti gotovo svaki alat dostupan putem naredbenog retka Linux operacijskog sustava. Od takvih alata su u kontekstu ovog rada najzanimljiviji programski jezik Python, mrežni alati poput netcat-a i ping-a te iptables tehnologija. Tijekom dizajniranja topologije mreže IMUNES podržava stvaranje nekoliko osnovnih vrsta mrežnih uređaja poput osnovnog i NAT usmjeritelja, mrežnog prekidača, korisničkih računala, poslužitelja te vanjskih konekcija i sučelja čijom kombinacijom se može postići gotovo svaka mrežna topologija.

Postoji nekoliko alternativnih simulatora mreže od kojih su neki Graphical Network System 3 čija je glavna razlika mogućnost kombinacija virtualnih i fizičkih uređaja unutar topologije, Cisco Packet Tracer koji se koristi u sklopu tečajeva koje nudi Cisco, Putty čija je glavna svrha konfiguracija mreže i mnogi drugi [12]. Glavne značajke koje odjeljuju IMUNES od sličnih rješenja su simulacija u stvarnom vremenu, mogućnosti kreiranja topologije s više stotina pa čak i nekoliko tisuća potpuno virtualnih čvorova bez potrebe kompleksne fizičke infrastrukture, jednostavno grafičko korisničko sučelje vidljivo na slici (Slika 3.2), prenosivost generiranih eksperimenata te činjenica da je tehnologija besplatna i otvorenog koda. IMUNES se trenutno koristi za potrebe izvođenja laboratorijskih vježbi i tijekom predavanja na Fakultetu elektrotehnike i računarstva te za testiranje proizvoda u tvrtki Ericsson Nikola Tesla [13].

Zbog skalabilnosti i simulacije mreže u stvarnom vremenu virtualni čvorovi se generiraju u krnjem obliku što ograničava mogućnosti tih čvorova. Tijekom pokušaja simulacije kompleksnih topologija i čvorova ovaj nedostatak bi mogao uvelike utjecati na mogućnosti

simulacije zadane mreže. Prednost koju pojedini konkurenti imaju nad IMUNES-om je rješavanje tog problema između ostalog i kombiniranjem virtualnih i fizičkih čvorova tijekom simulacije.



Slika 3.2 Prikaz grafičkog korisničkog sučelja alata IMUNES [14]

Dvije komande koje omogućavaju interakciju između virtualnih čvorova i operacijskog sustava domaćina su alati himage i hcp čijom upotrebom se olakšava implementacija simulacijskog alata. Alat himage putem naredbenog retka domaćina omogućava izvođenje naredbi na virtualnim čvorovima. Sintaksa je vrlo jednostavna te je prikazana u sljedećem primjeru:

```
himage moj_pc pwd
```

Alat se pokreće ključnom riječi himage nakon čega slijedi ime virtualnog čvora iza čega se nastavlja naredba pisana programskim jezikom Bash. Zadana naredba će se izvesti na virtualnom čvoru imena moj_pc, a rezultat izvođenja naredbe će se ispisati u nastavku naredbenog retka domaćina. Dok se himage koristi za pokretanje naredbi na virtualnim čvorovima, hcp se koristi za kopiranje datoteka s jednog virtualnog čvora na drugi, s domaćina na virtualne čvorove i obrnuto. Osnovna sintaksa i dva primjera korištenja alata hcp prikazani su na ispisu (Ispis 3.1).

```
hcp [cp_command_options] [vi_hostname1:]filename
[vi_hostname2:]filename
hcp moj_pc:skripta.sh .
hcp datoteka.txt moj_pc:
```

Ispis 3.1 Osnovna sintaksa hcp alata

Nakon ključne riječi hcp i neobaveznih parametara dolazi datoteka koja se kopira zadana opcionalnim nazivom virtualnog čvora i nazivom datoteke iza čega slijedi lokacija gdje će se kopirana datoteka nalaziti zadana istim principom kao i izvorna datoteka. Ako ime virtualnog čvora nije zadano, pretpostavlja se da je odredište datotečni sustav domaćinskog operacijskog sustava [15].

Prilikom generiranja topologije mreže pomoću grafičkog sučelja tehnologije IMUNES popis generiranih mrežnih uređaja zajedno s njihovim IP i MAC adresama, rutama, sučeljima, međusobnim vezama i ostalom konfiguracijom se spremaju u tekst datoteku s ekstenzijom .imn. O skripti za automatsko generiranje .imn datoteka će biti više riječi u sljedećem poglavlju.

4. Opis izrađenih alata

U sklopu ovog rada razvijena su dva alata kojima je zadaća iz zadanih sigurnosnih politika i opisa topologije mreže izgenerirati vatrozid pravila te ih rasporediti po vatrozid uređajima i naposljetku ispitati valjanost dobivenih pravila u odnosu na zadane politike. Ulazni podaci se zadaju u obliku datoteke formata json, a izlazni podaci se spremaju u nekoliko skriptnih datoteka koje se mogu pokrenuti koristeći skriptni jezik Bash te u jednu tekst datoteku s ekstenzijom .imn pripremljenu za korištenje s tehnologijom IMUNES koja predstavlja topologiju mreže. Glavnina aplikacija pisana je programskim jezikom Python uz nekoliko manjih skripti koje su pisane jezikom Bash. Također, za ulazne datoteke je korišten format json uz iznimku jedne tekst datoteke.

Jedan od značajnijih algoritama korištenih u ovom radu, algoritam za raspoređivanje vatrozid pravila na vatrozid uređaje, preuzet je iz znanstvenog rada Automatic Firewalls' Configuration using Argumentation Reasoning [16] te dodatno modificiran izbacivanjem dijelova koji su u kontekstu ovog projekta bili suvišni.

4.1. Alat za generiranje vatrozid pravila

Prvi od dva alata koristi se za generiranje vatrozid pravila. Ovaj alat na svom ulazu očekuje popis sigurnosnih politika i opis mreže te na izlazu daje vatrozid pravila pisana iptables sintaksom. Ovim alatom se olakšava kreiranje vatrozid pravila koja su tehničke prirode iz sigurnosnih politika koje su najčešće zadane na apstraktnijoj razini. Time se pojedincima i zaposlenicima zaduženim za pisanje i održavanje vatrozid pravila olakšava i ubrzava posao te se smanjuje mogućnost pogrešne konfiguracije.

Mreža kojom raspolaže pojedina organizacija može doseći velike razmjere. Dok je relativno jednostavno odrediti pristupne politike, pretvaranje politika u vatrozid pravila i njihovo raspoređivanje po vatrozid uređajima je često dugotrajan i kompleksan posao. Osobe zadužene za postavljanje vatrozida i njihovih pravila moraju razumjeti tehničke detalje mreže i zahtjeve svakog čvora i veze koji se nalaze u mreži. Ovim alatom se automatizira cjelokupan proces prevođenja apstraktnih pristupnih politika u vatrozid pravila te njihovo razmještanje po vatrozidima u mreži.

Alat ima nekoliko sastavnih dijelova koji se izvode slijedno. Prvi modul je zadužen za parsiranje podataka iz ulaznih datoteka te spremanje tih podataka u podatkovne strukture s kojima će manipulirati ostatak aplikacije. Sljedeći modul uz pomoć biblioteke ipaddress generira podatke koje korisnik nije zadao, ali su nužno potrebni za osiguravanje pravilnog funkcioniranja programskog rješenja. To obuhvaća IPv4 i IPv6 adrese te MAC adrese za svako sučelje u mreži. Spomenuta sučelja ne postoje u ulaznim datotekama te će se također generirati u sklopu ovog alata. Nakon toga slijede dva modula koji čine srž aplikacije. Njihova zadaća je konstrukcija vatrozid pravila iz svih raspoloživih podataka te raspodjela generiranih pravila na vatrozid uređaje.

Aplikacija raspolaže s tri ulazne datoteke. U prvoj datoteci su zadani mrežni segmenti zajedno s njihovim pripadnim čvorovima te apstraktne sigurnosne politike. Može se smatrati da su podaci zadani u formatu temeljenom na grafovima što je najlogičniji način spremanja topologije mreže te se može iskoristiti na efikasan način. Primjer mrežnih segmenata prikazan je u ispisu (Ispis 4.1).

```
"network_segments": {
        "0": [
            "admin-developer:0:0",
            "None:0:0"
        ],
        "1": [
            "None:0:1",
            "None:0:1#1"
        ],
        "2": [
            "None:0:2",
            "None:0:2#1"
        ]
    }
```

Ispis 4.1 Primjer dijela ulazne datoteke sa zadanim mrežnim segmentima

Iz primjera je očito da je zadana mreža s tri mrežna segmenta od kojih svaki sadrži po dva čvora. Može se primijetiti da u ulaznoj datoteci nisu zadane IP i MAC adrese niti slični podaci koji dodatno opisuju pojedine čvorove što znači da će se navedeni podaci morati generirati unutar alata kako bi se mogla kreirati vatrozid pravila.

Primjer pristupnih politika prikazan je u ispisu (Ispis 4.2):

```
"firewall-rule-498f8f02-28fd-42ab-bb3a-1ad321803fd3": {
```

```
        "allow": true,
        "from_objects": [
            "admin-
developer:0:0>CUSTOM:/a:linux:administration_tools#1b57ff86-
72bc-4e88-96ab-ddee135b06e7"
            ],
        "idn": "firewall-rule-498f8f02-28fd-42ab-bb3a-
1ad321803fd3",
        "to_objects": [
            "None:0:2#1>cpe:/o:centos:centos#c618953e-3ad3-
4a5a-a77a-1cbf5a2caba2"
            ]
    }
```

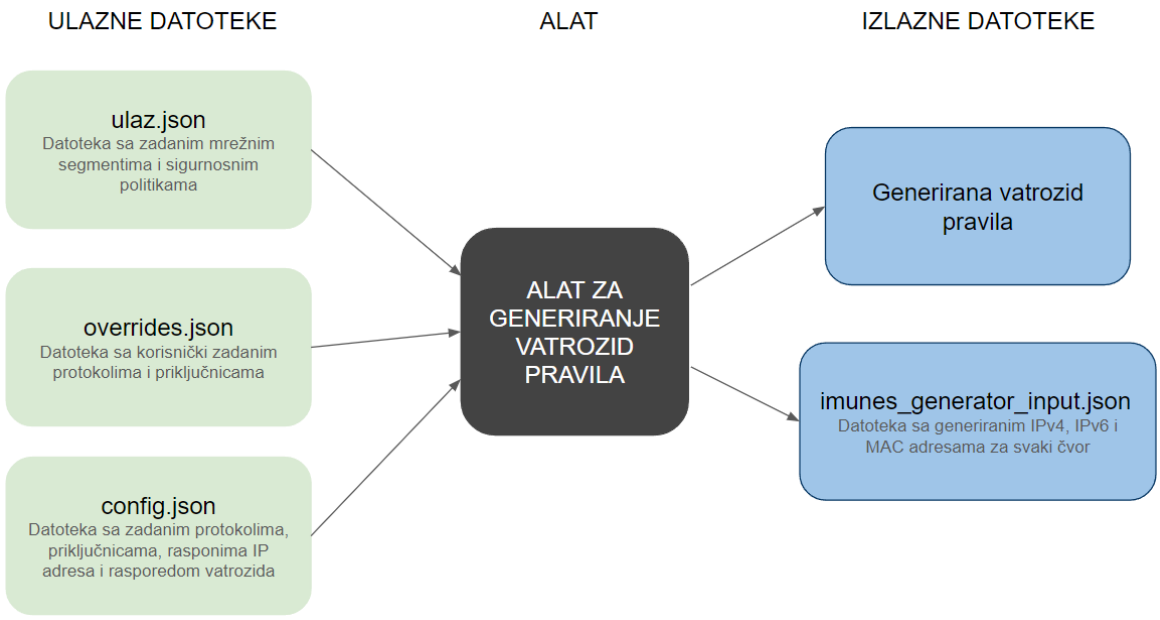
Ispis 4.2 Primjer dijela ulazne datoteke za zadanom pristupnom politikom

Može se zaključiti da se ovom pristupnom politikom želi omogućiti pristup aplikaciji `centos` na računalu naziva `None:0:2#1` iz aplikacije `administration_tools` koja se nalazi na računalu `admin-developer:0:0`. Kao i u primjeru mrežnih segmenata i ovdje se može primijetiti nedostatak svih potrebnih informacija kao na primjer informacije o protokolima i priključnicama koje koriste pojedine aplikacije. Sve to se zadaje u drugoj ulaznoj datoteci.

Važno je primijetiti da svaka pristupna politika može obuhvaćati više izvorišnih i odredišnih objekata. Pristupne politika s kojima može raspolagati ovaj alat pripadaju osrednjoj razini apstrakcije. Politike bi se mogle zadati na još apstraktnijem sloju, na primjer moglo bi se zadati "Računalo `admin-developer:0:0` aplikacijom `administration_tools` može pristupiti aplikaciji `centos` na računalu `None:0:2#1`" međutim to bi značilo da se u alat mora integrirati umjetna inteligencija s mogućnošću procesiranja prirodnog jezika što nije zadano u sklopu ovog diplomskog rada.

Sljedeća ulazna datoteka obuhvaća sve podatke koji dodatno omogućavaju generiranje vatrozid pravila. U njoj su definirani pretpostavljeni protokoli i priključnice za svaku aplikaciju obuhvaćenu sigurnosnim politikama, rasponi IP adresa za svaki mrežni segment te raspored vatrozid uređaja unutar mreže u odnosu na mrežne segmente. Ova datoteka se treba pažljivo definirati jer nedostatak bilo koje informacije može dovesti do neuspješnog izvođenja programa. Cilj je da se ova datoteka kontinuirano nadopunjuje informacijama kako se pojavi potreba za njihovim postojanjem. Na taj način ova datoteka služi kao baza znanja odnosno izvor domenskog znanja koji će rasti te će s vremenom trebati sve manje

unosa. Posljednja ulazna datoteka koju alat može parsirati sadrži podatke o specifičnim protokolima i priključnicama korištenim u pojedinim objektima pristupnih politika te se time nadjačavaju protokoli i priključnice zadane u prethodnoj datoteci. Navedene ulazne datoteke te struktura cijelokupnog alata prikazani su na slici (Slika 4.1).



Slika 4.1 Struktura alata za generiranje vatrozid pravila

Modul za konstruiranje vatrozid pravila iz svake sigurnosne politike ekstrahira izvorišne i odredišne parove aplikacija i računala te za svaki izvorišni i odredišni par kombinirajući podatke iz ulaznih datoteka puni listu vatrozid pravila. Pravila se spremaju u instance klase `FirewallRule` gdje svaka instanca obuhvaća jedno vatrozid pravilo. Parametrima klase se mogu zadati ulazno i izlazno sučelje, izvorišna i odredišna IP adresa, lanac na koji se "skače" u slučaju da paket zadovoljava uvjete pravila, protokol i odredišnu priključnicu. Osim toga svaka instanca sadrži izvorišni i odredišni objekt te set vatrozid uređaja na kojima bi se pravilo trebalo nalaziti.

Svaka instanca klase će se prilikom ispisa transformirati u iptables sintaksu te će svakom pravilu biti dodijeljen lanac FORWARD i akcija ACCEPT koja se izvršava ako paket zadovoljava postavljeno pravilo. Uz sve to je potrebno i odrediti politiku koja će se izvoditi u slučaju kada nijedno pravilo ne odgovara paketu, međutim to će se dodati u zasebnu datoteku koja vrijedi na razini cjelokupne mreže te se time osigurava konzistentnost krovne politike. Prema zadanim postavkama politika za sve pakete koji ne zadovoljavaju nijedno pravilo odbacuje takve pakete stoga je zadana akcija koja je pridružena svakom generiranom pravilu ACCEPT, međutim one se mogu jednostavno promijeniti modificiranjem

sigurnosnih politika i datoteke s predodređenim pravilima. U trenutnoj implementaciji alata jedini korišteni lanac je FORWARD zbog toga što se očekuje da vatrozid uređaji neće biti prethodno postavljeni u mreži već se definiraju zasebnom datotekom. S obzirom na to da se pristupne politike odnose na topologiju mreže koja je već zadana, nema smisla definirati politike za uređaje koji još ne postoje u mreži.

Pseudokod prevođenja vatrozid pravila iz apstraktne pristupne politike prikazan je u ispisu (Ispis 4.3).

```
foreach policy in security_policies do
    foreach source in policy.source do
        foreach destination in policy.destination do
            rules.add(source, destination, policy.action,
policy.protocol, source.port)
        end
    end
end
```

Ispis 4.3 Algoritam za prevođenje sigurnosne politike u vatrozid pravilo

Raspoređivanje pravila na zadane vatrozid uređaje implementirano je na temelju rada Automatic Firewalls' Configuration using Argumentation Reasoning [16] koje radi na jednostavnom principu pronalaženja svih putova od jednog računalnog čvora do drugog te ako je zadana akcija propuštanje paketa onda se pravilo kopira na svaki vatrozid koji se nalazi na tom putu, a ako se paket ne smije propustiti onda se pravilo dodaje samo na vatrozid najbliži izvornom računalu. Time se osigurava propuštanje paketa do odredišta kroz sve potrebne vatrozide i odbacivanje onih paketa koji ne zadovoljavaju pravila na samom početku prije nego prodru dublje u mrežu. Pseudokod spomenutog algoritma prikazan je u ispisu (Ispis 4.4).

```
foreach rule in ruleset do
    foreach p in paths(rule.source, rule.dest) do
        if rule.action==“allow" then
            foreach fw in firewalls_in_path(p) do
                fw.rule.add(rule)
            end
       end
       else most
upstream(firewalls_in_path(p)).rule.add(rule)
   end
```

```
end
```

Ispis 4.4 Algoritam za raspoređivanje pravila na vatrozid uređaje

Naposljetku alat sprema sva izgenerirana pravila u izlazne datoteke s ekstenzijom .sh tako da kreira datoteku za svaki vatrozid koji se nalazi u mreži te raspoređena pravila zapiše u pripadnu datoteku. Primjer jedne takve datoteke vidljiv je na slici (Slika 4.2). Kako bi se pravila kopirala na fizičke vatrozid uređaje u mreži potrebno je svaku datoteku kopirati na pripadajući vatrozid te ju izvršiti. Ovim putem svaki vatrozid uređaj ima pripadajuću datoteku sa svim vatrozid pravilima koja se odnose na taj i samo taj vatrozid. Time se olakšava posao korisnika koji ne mora razmišljati o odvajanju pravila na potrebne vatrozide već ovaj alat to radi za njega.

imunes > fw_assignments > $ C.sh

```
1   iptables -A FORWARD -p tcp -s 10.0.3.2 -d 10.0.2.3 --dport 389 -m state --state NEW -j ACCEPT
2   iptables -A FORWARD -p tcp -i eth0 -d 10.0.3.3 --dport 443 -m state --state NEW -j ACCEPT
3   iptables -A FORWARD -p tcp -s 10.0.2.3 -d 10.0.3.2 --dport 25 -m state --state NEW -j ACCEPT
4   iptables -A FORWARD -p tcp -s 10.0.3.3 -d 10.0.2.3 --dport 389 -m state --state NEW -j ACCEPT
5   iptables -A FORWARD -p tcp -s 10.0.3.2 -d 10.0.2.3 --dport 25 -m state --state NEW -j ACCEPT
6   iptables -A FORWARD -p tcp -s 10.0.3.3 -d 10.0.2.3 --dport 636 -m state --state NEW -j ACCEPT
7   iptables -A FORWARD -p tcp -s 10.0.1.3 -d 10.0.3.3 --dport 22 -m state --state NEW -j ACCEPT
8   iptables -A FORWARD -p tcp -i eth0 -d 10.0.3.3 --dport 80 -m state --state NEW -j ACCEPT
9   iptables -A FORWARD -p tcp -i eth0 -d 10.0.3.2 --dport 25 -m state --state NEW -j ACCEPT
10  iptables -A FORWARD -p tcp -s 10.0.3.2 -d 10.0.2.3 --dport 636 -m state --state NEW -j ACCEPT
11  iptables -A FORWARD -p tcp -s 10.0.1.3 -d 10.0.3.2 --dport 22 -m state --state NEW -j ACCEPT
```

Slika 4.2 Primjer generirane datoteke s vatrozid pravilima

Osim datoteka s pravilima alat generira i jednu datoteku json formata u koju sprema IPv4, IPv6 i MAC adrese generirane unutar alata koje će se kasnije koristiti kao ulazni podaci za alat kojim se ispituje valjanost dobivenih vatrozid pravila.

Neke od zadanih politika opisuju pristup pojedinih računala prema internetu, a s obzirom na to da se internet kao cjelina ne može opisati jednom IP adresom i priključnicom, bilo je potrebno unijeti promjene u algoritme koje će rješavati specifične slučajeve pristupa internetu ili s interneta. Naime, čvor internet se ne prevodi u specifične IP adrese kao ostali čvorovi već se objekt prevodi u ulazno ili izlazno sučelje te se tako zadaje u pravilu.

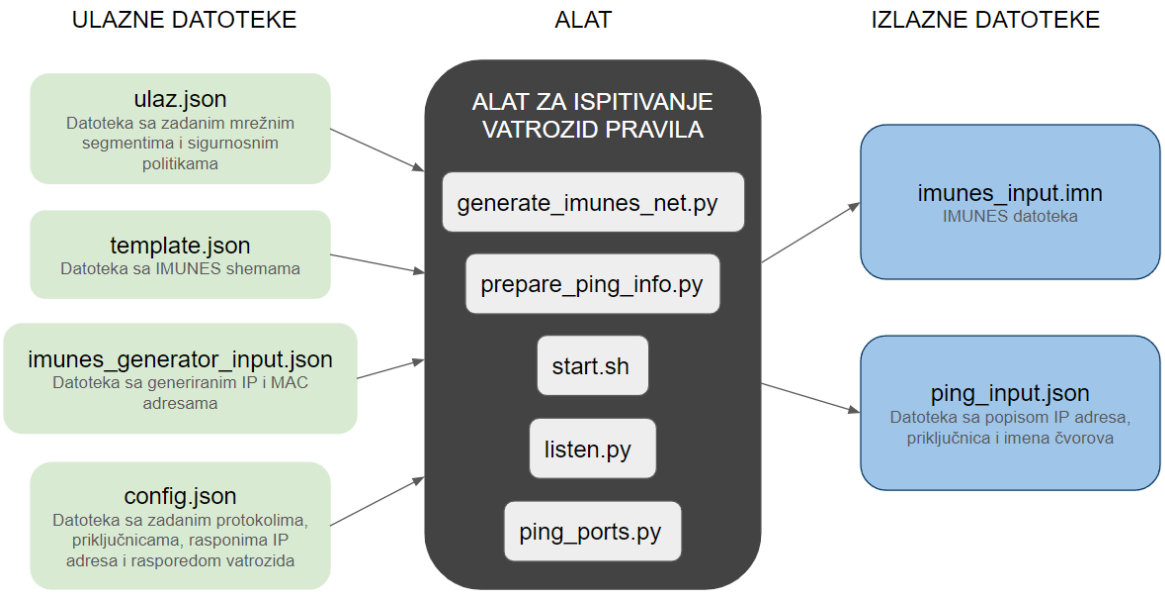
Glavni izazovi prilikom izrade alata bili su identificiranje svih potrebnih sučelja u mreži i dodjela IP adresa uređajima i sučeljima te sakupljanje potrebnih podataka o protokolima i priključnicama.

Razvijeni alat na brz i jednostavan način koristeći ulazne datoteke prevodi apstraktne pristupne politike u vatrozid pravila, raspoređuje ih po zadanim vatrozid uređajima te ih

sprema u datoteke spremne za prenošenje i izvođenje na fizičkim vatrozid uređajima. Dodatno, kreira datoteke s generiranim mrežnim podacima te tako omogućava ispitivanje valjanosti generiranih vatrozid pravila.

4.2. Alat za testiranje dobivenih pravila u odnosu na ulazne politike

Ovim alatom se ispituje valjanost vatrozid pravila generiranih alatom opisanim u prethodnom potpoglavlju. Ulazni podaci su dvije datoteke koje zadaje korisnik, a one su također i ulazne datoteke u prethodni alat. Riječ je o datoteci s popisom sigurnosnih politika i mrežnih segmenata te o datoteci s podacima o protokolima, priključnicama i rasponima IP adresa. Osim njih, na ulazu se očekuje i datoteka formata json koju generira prethodni alat u kojoj su zapisane generirane IPv4, IPv6 i MAC adrese svakog pojedinog računala i sučelja te datoteka koja sadrži sheme za generiranje IMUNES datoteke u koju je potrebno ubaciti mrežne podatke generirane ovim alatom. Navedene ulazne datoteke i struktura alata prikazani su na slici (Slika 4.3).



Slika 4.3 Struktura alata za ispitivanje ispravnosti vatrozid pravila

Alat služi za testiranje ispravnosti vatrozid pravila u odnosu na zadane sigurnosne politike. S obzirom na to da prethodni alat generira pravila za arbitrarne politike i topologiju mreže, bilo bi nepraktično provjeravati ispravnost dobivenih rezultata ručno. Stoga je stvoren ovaj alat kako bi se na automatiziran način simulirala zadana mreža, prekopirala vatrozid pravila na pripadajuće vatrozid uređaje te simulirao promet s raznih računala na razne priključnice.

Praćenje kretanja simuliranog prometa kroz vatrozid uređaje promatraču daje informaciju o ispravnosti vatrozid pravila te upućuje na eventualne greške u vatrozid pravilima, politikama ili konfiguraciji mreže i čvorova.

Alat se sastoji od sljedećih komponenata:

- Skripta pisana programskim jezikom Python koja na temelju prethodno spomenutih ulaznih datoteka generira tekst datoteku s ekstenzijom .imn koja će kasnije biti pokrenuta tehnologijom IMUNES (`generate_imunes_net.py`)
- Virtualna mašina s instaliranim programom IMUNES
- Python skripta za ekstrakciju IP adresa, priključnica i imena čvorova (`prepare_ping_info.py`)
- Bash skripta za kopiranje i pokretanje vatrozid pravila i simulacije prometa (`start.sh`)
- Python skripta za osluškivanje dolaznih zahtjeva (`listen.py`)
- Bash skripta za slanje zahtjeva (`ping_ports.py`)

Skripte su napisane na način da automatizirano izvršavaju što više zadataka te je potrebna minimalna intervencija korisnika u obliku premještanja datoteka na virtualnu mašinu, pokretanje eksperimenta u IMUNES tehnologiji i izvođenja skripti. Zbog izbora korištenih programskih jezika i virtualne mašine alat je maksimalno prenosiv na sve popularnije operacijske sustave.

Za ispravno funkcioniranje alata prvo je potrebno generirati vatrozid pravila i json datoteku pokretanjem prethodnog alata. Nakon toga se pokreće skripta za generiranje IMUNES datoteke imena `generate_imunes_net.py` koja će također stvoriti i pomoćne datoteke za lakšu izvedbu skripti za simulaciju prometa. Slijedi izvođenje skripte koje priprema podatke iz pomoćne skripte naziva `prepare_ping_info.py`. Sve potrebne datoteke se tada mogu kopirati s domaćinskog operacijskog sustava na virtualnu mašinu te se jednostavnim izvođenjem skripte za kopiranje i pokretanje datoteka provodi kopiranje vatrozid pravila i simulacija prometa.

Skripta koja generira IMUNES datoteku se sastoji od tri modula. Prvi modul, slično kao i u prethodnom alatu, parsira ulazne datoteke kako bi skripta imala sve potrebne podatke za daljnji rad te ih sprema u adekvatne podatkovne strukture. Drugi modul priprema, generira i formatira sve potrebne podatke kako bi ih sljedeći modul mogao umetnuti u sheme za razne

vrste virtualnih čvorova s kojima raspolaže IMUNES te se naposljetku sheme sklapaju u jednu cjelinu koja se zapisuje u datoteku i time nastaje IMUNES datoteka.

Ovaj alat može na taj način kreirati četiri različite vrste virtualnih čvorova. Dvije od četiri vrste čvorova predstavljaju čvorove računala i poslužitelja. Za njihovo kreiranje je potrebno zadati index i ime čvora, IPv4 i MAC adrese te statičke rute. Osim toga, skripta generira sva potrebna sučelja prema drugim čvorovima u mreži te zapisuje sve međusobne veze. Treća vrsta virtualnog čvora je mrežni prekidač koji povezuje sva računala pojedinog segmenta s četvrtom vrstom čvora, a to je usmjeritelj. Za generiranje usmjeriteljskih čvorova potrebno je, uz prethodno spomenute podatke zadati i IPv6 adrese pojedinih sučelja, a za kreiranje mrežnih prekidača je potrebno samo povezati sučelja s pripadajućim čvorovima.

Kako bi se vatrozid pravila mogla koristiti na virtualnim čvorovima potrebno je nakon pokretanja eksperimenta u IMUNES-u kopirati sve potrebne datoteke na virtualne čvorove te ih ispravnim naredbama pokrenuti. U tu svrhu se koriste alati `himage` i `hcp`, a kako bi se maksimizirala automatizacija te naredbe su skupljene u bash skriptu prikazanu u kodu na slici (Slika 4.4).

Skripta na početku provjerava da li je moguće pokretati naredbe na virtualnim čvorovima s domaćinskog operacijskog sustava, nakon čega se definira direktorij u koji će se spremati datoteke s vatrozid pravilima. Potom se na svaku datoteku s generiranim vatrozid pravilima dodaje ispis datoteke u kojima su zapisana dodatna pravila na razini cjelokupne mreže poput zadane politike za pakete koji ne odgovaraju niti jednom pravilu i pravila za propuštanje paketa vezanih uz postojeće konekcije odnosno pravila koja omogućavaju dvosmjerne konekcije. Dobivene datoteke se kopiraju na pripadajuće virtualne čvorove. Naposljetku se kopiraju i izvode datoteke za simulaciju prometa.

Na 32. liniji navedene skripte svakom vatrozidu se na kraj lanca FORWARD dodaje pravilo pisano iptables sintaksom kojim se onemogućava sav promet odnosno svi paketi koji ne zadovolje ostala pravilo u tom lancu će biti odbačena. Ovo pravilo je dodano zbog problema s postavljanjem zadane politike. Naime, postavljanjem politike koja odbacuje sve pakete koji ne zadovolje nijedno pravilo u lancu spontano se promijene statičke IP rute te se efektivno onemogući svaki promet kroz taj vatrozid. Zbog ovog problema i nedostatka adekvatnog rješenja, napravljeno je zaobilazno rješenje tako da se zadana politika postavi u ACCEPT te se na kraj FORWARD lanca na svakom vatrozidu doda pravilo koje odbacuje svaki paket.

Time se postigao isti učinak, međutim ovo rješenje nije najbolja praksa jer se otvara prostor za pogreške u slučaju kada se treba nadopuniti lanac novim pravilima.

imunes > imunes_scripts > $ start.sh

```
#! /bin/sh

if test $(id -u) -ne 0; then
    echo "Try: sudo $0"
    exit 2
fi

# copy firewall rules
fw_assignments_folder=fw_assignments
fw_output_folder=fw_output
rm -r $fw_output_folder
mkdir $fw_output_folder

for fw in $(ls $fw_assignments_folder)
do
    # check if the experiment is running
    himage ${fw%.*} hostname > /dev/null 2>&1
    if test $? -ne 0; then
        echo "Experiment not executed. Run cleanupAll and try again"
        exit 2
    fi

    # create file for each firewall node and copy firewall rules into it
    cat template_fw.sh > temp.txt
    cat $fw_assignments_folder/$fw >> temp.txt
    cat temp.txt > $fw_output_folder/$fw

    # copy each file to its designated firewall node and execute it
    hcp $fw_output_folder/$fw ${fw%.*}:
    himage ${fw%.*} chmod u+x $fw
    himage ${fw%.*} ./$fw
    himage ${fw%.*} iptables -A FORWARD -p tcp -m state --state NEW -j DROP
done

# ping all ports on all nodes
for node in $(cat ping_input.json | jq .nodes | jq -r @sh)
do
    host=$(echo $node | xargs echo)
    hcp ping_ports.py $host:
    hcp ping_input.json $host:
    hcp listen.py $host:

    for port_string in $(cat ping_input.json | jq .ports | jq -r @sh)
    do
        port=$(echo $port_string | xargs echo)
        himage $host python listen.py $port &
    done

    himage $host python ping_ports.py
done
```

Slika 4.4 Bash skripta za kopiranje i pokretanje vatrozid pravila i simulacije prometa

Izvorna ideja simulacije prometa na virtualnim čvorovima je uključivala pregledavanje dnevničkih zapisa u svrhu potvrde ispravnog rada vatrozida i njegovih pripadnih pravila, međutim u procesu postavljanja dnevničkih zapisa došlo je do nekoliko problema, od nepostojanja usluge koja upravlja zapisivanjem događaja u dnevnike na IMUNES-ovim

usmjeriteljima do spontanog mijenjanja statičkih IP ruta nakon izvršavanja pojedinih iptables naredbi.

Zbog kompleksnosti i nepoznate veličine problema odlučeno je da će se napraviti zaobilazno rješenje koje će na svakom virtualnom čvoru tipa računalo ili poslužitelj postaviti jednostavan program za osluškivanje koji će na svaki zahtjev na predviđenim priključnicama odgovoriti kako bi pošiljatelj znao da je zahtjev prošao kroz sve potrebne vatrozide i stigao do svog odredišta te uz to po primitku odgovora verificira ispravnost pravila napisanih za dvosmjerne veze. Ovo je neefikasno rješenje koje usporava cjelokupni postupak testiranja i validacije vatrozid pravila te nepotrebno troši računalne resurse i uvodi potrebu za dodatnim skriptama, zbog toga bi se u budućnosti trebalo zamijeniti postavljanjem i analiziranjem dnevničkih zapisa.

Program za osluškivanje je jednostavna Python skripta koja otvara zadanu priključnicu za osluškivanje te na njoj osluškuje dolazne zahtjeve. Kada zaprimi zahtjev, skripta pročita poslanu poruku te tu istu poruku pošalje natrag pošiljatelju nakon čega zatvara konekciju i prekida osluškivanje na zadanoj priključnici. Kod echo poslužitelja za osluškivanje prikazan je na slici (Slika 4.5).

imunes > imunes_scripts > listen.py > ...

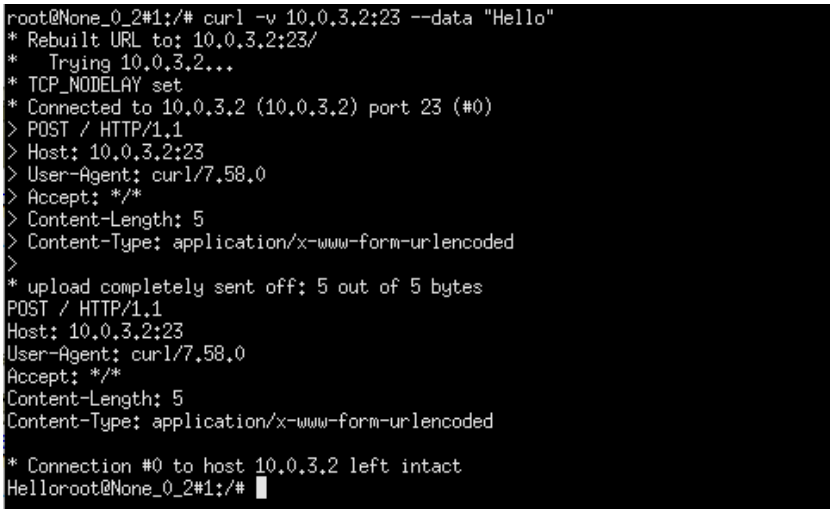
```
1   import sys
2   import socket
3
4   port = int(sys.argv[1])
5   s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6   try:
7       s.bind(('', port))
8   except:
9       s.close()
10      exit(2)
11  s.listen(9)
12  connection, address = s.accept()
13  data = connection.recv(1024)
14  connection.sendall(data)
15
16  s.close()
```

Slika 4.5 Python skripta za osluškivanje dolaznih zahtjeva

Zahtjev se može poslati na nekoliko načina, a jedan od jednostavnijih je korištenjem alata naredbenog retka naziva curl kojem se zadaje IP adresa i priključnica odredišta. Dodatno, mogu se zadati podaci koji će se poslati u zahtjevu korištenjem parametra `--data`. Primjeri zahtjeva i odgovora zajedno s ispisom dodatnih informacija u konzolu prikazani su na

slikama (Slika 4.6 i Slika 4.7). U oba primjera se može primijetiti da je poslana poruka „Hello“ stigla do odredišta te je reflektirana pošiljatelju gdje je i zaprimljena.

Slika 4.6 Primjer ispisa skripte za osluškivanje



Slika 4.7 Primjer poslanog zahtjeva alatom curl

Ovaj alat na automatiziran način stvara IMUNES datoteku za simulaciju mreže i prometa iz zadanog opisa mreže te kopira generirana vatrozid pravila u virtualne čvorove te uz pokretanje pomoćnih skripti priprema okruženje za postavljanje vatrozid pravila i izvodi simulaciju prometa.

5. Rezultati

Kako bi se simulacija prometa što više automatizirala programi za osluškivanje i slanje zahtjeva prema predviđenim priključnicama sakupljeni su u skripte. Skripte za osluškivanje (Slika 4.5) pokrenute su na svim priključnicama i virtualnim čvorovima koji postoje u ulaznoj datoteci odmah nakon prepisivanja vatrozid pravila u skripti prikazanoj u prethodnom poglavlju na slici (Slika 4.4).

Slično tome, skripte za slanje zahtjeva prema preodređenim priključnicama kopirane su na sve virtualne čvorove i pokrenute tek nakon što se izvedu svi programi za osluškivanje. Kod takve skripte prikazan je u nastavku (Slika 5.1).

imunes > imunes_scripts > ping_ports.py > ...

```
1   import json
2   import os
3   import subprocess
4
5   INPUT_FILE = "ping_input.json"
6
7
8   def ping_all(nodes, ports):
9       FNULL = open(os.devnull, 'w')
10      for ip_address in nodes:
11          for port in ports:
12              subprocess.call("curl {}:{} --data 'Hello'
                --connect-timeout 1".format(ip_address, port),
                shell=True, stdout=FNULL)
13
14
15  def main():
16      content = ""
17      with open(INPUT_FILE, "r") as file:
18          content = json.load(file)
19
20      ip_addresses = content["ip_addresses"]
21      ports = content["ports"]
22
23      ping_all(ip_addresses, ports)
24
25
26  main()
```

Slika 5.1 Skripta za slanje zahtjeva

Skripte za osluškivanje omogućuju prikupljanje podataka o dolaznom prometu na predviđenim priključnicama te analizu raznih vrsta paketa. S druge strane, skripte za slanje zahtjeva omogućuju simulaciju različitih vrsta paketa i provjeru povratnih odgovora.

Umjesto ručnog generiranja i slanja prometa, automatizacija tog procesa omogućuje bržu i konzistentniju simulaciju u raznim uvjetima te olakšava validaciju pravila.

Nakon pokretanja skripte analizom ispisa može se zaključiti koji od zahtjeva su zadovoljili pravila pa su stoga propušteni kroz vatrozid uređaje i stigli do svog odredišta koje je potom natrag poslalo odgovor na zaprimljeni zahtjev. Primjer dijela ispisa nakon pokretanja skripti vidljiv je na slici (Slika 5.2). Prikazane su dvije različite vrste ispisa. Prvi ispis prikazuje uspješno uspostavljenu vezu s priključnicom 443 na čvoru s IP adresom 10.0.3.2. Može se primijetiti da je uspješno poslano pet bajtova. Drugi primjer ispisa opisuje situaciju kada vatrozid odbaci zahtjev te se pokušaj konekcije prekida nakon zadanog vremenskog ograničenja. Pošiljatelj zahtjeva neće dobiti povratnu informaciju od vatrozida da je paket odbačen te to može zaključiti isključivo po nedostatku bilo kakvog odgovora.

```
* Rebuilt URL to: 10.0.3.2:443/
*   Trying 10.0.3.2...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to 10.0.3.2 (10.0.3.2) port 443 (#0)
> POST / HTTP/1.1
> Host: 10.0.3.2:443
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 5
> Content-Type: application/x-www-form-urlencoded
>
} [5 bytes data]
* upload completely sent off: 5 out of 5 bytes
{ [17 bytes data]
100   155    0   150  100     5  11538    384 --:--:-- --:--:-- --:--:-- 22142
* Connection #0 to host 10.0.3.2 left intact
* Rebuilt URL to: 10.0.1.1:143/
*   Trying 10.0.1.1...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connection timed out after 1001 milliseconds
* stopped the pause stream!
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
* Closing connection 0
curl: (28) Connection timed out after 1001 milliseconds
```

Slika 5.2 Primjer ispisa nakon slanja zahtjeva

Analizom ovih ispisa i usporedbom sa zadanim sigurnosnim politikama korisnik donosi zaključke o ispravnosti generiranih pravila te po potrebi mijenja neke od njih.

6. Diskusija

Tijekom izrade alata i analizom generiranih vatrozid pravila zaključeno je da ovaj alat na adekvatan način procesira i nadopunjava sigurnosne politike te iz njih stvara vatrozid pravila koja se mogu unijeti u stvarni fizički sustav i na prikladan način koristiti u mrežnom sustavu za filtriranje mrežnog prometa i kontrolu pristupa pojedinim mrežnim čvorovima. Također, jednostavnim algoritmom sva pravila se raspoređuju na odgovarajuće vatrozid uređaje, što dodatno minimizira potrebu za korisničkom intervencijom.

Dobivena pravila su testirana na virtualnom mrežnom sustavu u kojem je provedena simulacija prometa. Zaključeno je da su pravila izvedena iz politika na odgovarajući način te se mrežni paketi pravilno filtriraju, odnosno propuštaju se oni koji bi trebali stići do svog odredišta prema zadanim sigurnosnim politikama, dok se odbacuju oni za koje nije zadana nijedna politika. Time je također ispitana ispravnost alata za testiranje, koji uz manje probleme i kreativna zaobilazna rješenja uspijeva zadovoljiti postavljene zahtjeve.

Uspoređujući vrijeme potrebno za ručno pretvaranje politika u pravila i njihovo raspoređivanje po vatrozidima s vremenom koje je potrebno ovom alatu, očito je da je vremenska ušteda značajna te raste s veličinom i kompleksnošću zadanih mrežnih sustava. Osim vremenskog aspekta, važno je istaknuti konzistentnost i točnost koju donosi ovaj alat. Budući da je izlaz alata uvijek isti za iste ulazne podatke te da alat uvijek prolazi set predefiniranih koraka, svaka greška koja se pojavi u generiranim pravilima je greška u dizajnu alata, a ne propust. To znači da je ukupni broj grešaka koje će se pojaviti tijekom stvaranja vatrozid pravila puno manji u odnosu na broj grešaka koje bi nastale ručnim pretvaranjem politika.

Tijekom izrade ovog projekta pojavilo se nekoliko problema koji su izrazito utjecali na kompleksnost i jasnoću alata za testiranje pravila. Velika količina isprobanih naredbi vezanih uz tehnologiju iptables na virtualnim čvorovima nije funkcionirala prema očekivanom ponašanju. Štoviše jedini skup iptables naredbi koji je uvijek funkcionirao točno su generirane naredbe za postavljanje vatrozid pravila. Postavljanje zadane politike za pakete koji ne zadovoljavaju nijedno od vatrozid pravila je rezultiralo spontanom promjenom statičkih IP ruta na vatrozidima nakon određenog vremenskog intervala.

Također, samim postavljanjem zadane politike se efektivno onemogućio sav promet između čvorova unutar različitih segmenata odnosno promet koji mora proći kroz jedan ili više vatrozid uređaja unatoč postojanju vatrozid pravila koja omogućavaju upravo takve pakete.

Osim toga, usprkos brojnim pokušajima nije bilo moguće pravilno instalirati i pokrenuti servis zadužen za stvaranje i ažuriranje dnevničkih zapisa o svim događanjima unutar pojedinih vatrozida. Unatoč nepostojanju vidljivih grešaka nakon pokretanja servisa i omogućavanja zapisivanja, zapisi nisu bili uspješno stvoreni.

7. Budući rad

Razvijeni alati otvaraju mnoge mogućnosti za nadogradnju, od implementacije umjetne inteligencije za procesiranje apstraktnijih sigurnosnih politika do proširenja mogućnosti filtriranja na ostale razine modela međusobnog povezivanja otvorenih sustava.

Jedan od korisnijih dodataka, koji bi bio najkorisniji organizacijama koje tek postavljaju fizičku mrežu, ali i onim koje žele provjeriti da im je trenutni mrežni sustav optimalno postavljen, je implementacija algoritma za generiranje rasporeda vatrozid uređaja. Točnije, ovaj algoritam generira optimalan raspored vatrozid uređaja unutar mreže i svrstavanje računalnih čvorova u segmente tako da su čvorovi sa sličnim pristupnim politika u istim segmentima čime se postiže maksimalna kontrola pristupa.

Alat, u svojoj trenutnoj implementaciji, razumije isključivo sigurnosne politike niže razine apstraktnosti koje su zadane u formatu temeljenom na grafovima. Bilo bi korisno, osobito za korisnike koji se tek počinju upoznavati s funkcionalnostima vatrozida, kada bi alat mogao obraditi politike zadane na najvišoj apstraktnoj razini, odnosno politike zadane prirodnim jezikom kojim komuniciramo u svakodnevnom životu. U tu svrhu trebao bi se implementirati algoritam za obradu prirodnog jezika. Osim toga, umjetna inteligencija bi se mogla iskoristiti za validaciju zadanih pristupnih politika te upozorenje na potencijalne propuste ili nekonzistentnosti.

Logičan sljedeći korak u nadogradnji razvijenih alata je implementacija grafičkog korisničkog sučelja koje bi od krajnjih korisnika sakrilo pravu implementaciju alata i olakšalo definiranje ulaznih podataka i promjenu konfiguracije s ograničenim dijelovima koji se mogu mijenjati, čime se minimizira vjerojatnost uvođenja promjena koje uzrokuju prijevremeni prestanak izvršavanja programa.

Osim navedenih dodataka mogla bi se implementirati podrška za generiranje vatrozid pravila namjenjena ne samo iptables tehnologiji već alternativnim rješenjima poput nftables-a koja je nasljednik iptables-a te sličnim tehnologijama.

Zaključak

U sklopu ovog diplomskog rada razvijena su dva alata kojima je zadaća korisnicima olakšati postavljanje i održavanje vatrozid uređaja te time održavati sigurnost unutar mrežnog sustava. Taj zadatak nastoje obaviti prevođenjem apstraktnih sigurnosnih politika u konkretna vatrozid pravila, razmjestiti ih na predodređene vatrozid uređaje te ispitati ispravnost generiranih pravila u odnosu na zadane politike.

Korisnik je dužan pružiti popis pristupnih politika, opis topologije mreže i neke osnovne informacije o mrežnom sustavu poput raspona IP adresa i rasporeda vatrozid uređaja. Dodatno, po potrebi se može upotpuniti baza znanja umetanjem softverskih alata, protokola i priključnica korištenih u mreži te se mogu zadati specifične priključnice za pojedini softver. Na temelju navedenih informacija, alati su sposobni generirati vatrozid pravila i sve ostale datoteke potrebne za omogućavanje i provođenje testiranja.

Ovi alati u velikoj mjeri ubrzavaju postupak postavljanja i konfiguracije mreže vatrozid uređaja te umanjuju mogućnost pogrešaka koja se inače pojavi tijekom ručnog postavljanja pravila. Automatizirani proces osigurava konzistentnost i točnost te štedi vrijeme i smanjuje rizik. Osim toga, omogućavaju jednostavno eksperimentiranje s raznim topologijama i konfiguracijama mreže te stjecanje praktičnog znanja o funkcionalnostima vatrozida te razumijevanje njihove uloge unutar sustava bez potrebe implementacije fizičkog mrežnog sustava što je skup i dugotrajan proces.

IMUNES je izrazito korisna tehnologija za ispitivanje raznih pretpostavki o ponašanju i interakciji kompleksnih mrežnih sustava. Ova tehnologija olakšava i ubrzava postavljanje mreže i eksperimentiranje sa raznim topologijama i konfiguracijama. Također, pomaže u shvaćanju načina na koji mrežni sustavi funkcioniraju. Unatoč tome, pokušaji prilagođavanja samih Docker spremnika na kojima počiva cijeli IMUNES sustav može biti izazov te često rezultira neuspjehom.

Tehnologija iptables pokazala se jednostavnom za korištenje kada je u pitanju definiranje vatrozid pravila. Sintaksa je jednostavna i jasna te je samo potrebno detaljnije proučiti tablice i lance koji sadrže pravila. Usprkos tome, većina problema na koje se naišlo tijekom izrade ovog rada uzrokovana je izvršavanjem pojedinih iptables naredbi, međutim uzrok problema nije pronađen.

Literatura

[1] Fortinet, *What is Network Traffic?*, Fortinet. Poveznica: https://www.fortinet.com/resources/cyberglossary/network-traffic; pristupljeno 21. svibnja 2023.

[2] Check Point, *What is a Firewall?*, Check Point Software Technologies. Poveznica: https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/#:~:text=A%20Firewall%20is%20a%20network,network%20and%20the%20public%20Internet; pristupljeno 20. svibnja 2023.

[3] Yasar K., Lutkevich B., *firewall*, TechTarget, (2023, travanj). Poveznica: https://www.techtarget.com/searchsecurity/definition/firewall; pristupljeno 20. svibnja 2023.

[4] Cisco, *What Is a Firewall?*, Cisco Systems. Poveznica: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html; pristupljeno 20. svibnja 2023.

[5] ArchLinux, *iptables*, ArchWiki, (2023, svibanj). Poveznica: https://wiki.archlinux.org/title/iptables#:~:text=iptables%20is%20a%20command%20line,console%20and%20graphical%20front%2Dends; pristupljano 24. svibnja 2023.

[6] Eychenne H., *iptables*, die.net. Poveznica: https://linux.die.net/man/8/iptables; pristupljeno 24. svibnja 2023.

[7] Martins F., *Controlling Security Policies in a Distributed Environment*, ResearchGate, (2004, svibanj). Poveznica: https://www.researchgate.net/figure/A-two-firewall-tiered-network-architecture_fig1_2882120; pristupljeno 24. svibnja 2023.

[8] Linode, *A Tutorial for Controlling Network Traffic with iptables*, Linode, (2010, srpanj). Poveznica: https://www.linode.com/docs/guides/control-network-traffic-with-iptables/; pristupljeno 25. svibnja 2023.

[9] DataFlair, *What are Iptables in Linux?*, DataFlair. Poveznica: https://data-flair.training/blogs/wp-content/uploads/sites/2/2022/04/iptables-in-linux.webp; pristupljeno 27. svibnja 2023.

[10] IMUNES, *Introduction*, IMUNES Manual, (2018, studeni). Poveznica: http://imunes.net/dl/guide/node2.html; pristupljeno: 27. svibnja 2023.

[11] IMUNES, *About*, IMUNES Manual, (2018, studeni). Poveznica: http://imunes.net/about; pristupljeno 27. svibnja 2023.

[12] I-Medita, *Top 10 Most Popular Network Simulation Tools*, I-Medita, (2021, siječanj). Poveznica: https://www.imedita.com/blog/top-10-list-of-network-simulation-tools/; pristupljeno 28. svibnja 2023.

[13] IMUNES, *Integrated Multiprotocol Network Emulator/Simulator*, IMUNES, (2004). Poveznica: http://imunes.net/; pristupljeno 28. svibnja 2023.

[14] Linkletter B., *IMUNES on Linux*, Open Source Routing and Network Simulation blog, (2015, kolovoz). https://www.brianlinkletter.com/2015/08/imunes-on-linux/; pristupljeno 28. svibnja 2023.

[15] IMUNES, *Advanced Usage*, IMUNES Manual, (2018, studeni). Poveznica: http://imunes.net/dl/guide/node5.html#SECTION00580000000000000000; pristupljeno 28. svibnja 2023.

[16] Karafili E., Valenza F., *Automatic Firewalls' Configuration using Argumentation Reasoning*. Znanstveni rad. University of Southampton, Politecnico di Torino, 2020.

[17] Verizon, *Data Breach Investigations Report*, Verizon, (2022). Poveznica: https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf; pristuplljeno 11. lipnja 2023.

[18] Kovačević I., Štengl B., Groš S., *Systematic review of automatic translation of high-level security policy into firewall rules*. 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, (2022), str. 1063-1068

Sažetak

Podešavanje i ispitivanje vatrozida na temelju zadanih politika visoke razine apstrakcije

Vatrozid je uređaj ili programska podrška koji omogućuje nadziranje i upravljanje mrežnim prometom između računalnih mreža ili njihovih segmenata. Upravljanje se vrši na temelju apstraktnih pristupnih politika koje su prevedene u niz pravila za vatrozid te se svaki mrežni paket uspoređuje s navedenim pravilima te shodno tome propušta ili odbacuje. U ovom radu se opisuje programsko rješenje koje na automatiziran način prevodi apstraktne politike zadane u obliku modela temeljenog na grafu, uz prisutnost opisa računalne mreže, u niz pravila za vatrozide te ih razmješta na vatrozid uređaje zadane u mreži. Dodatno, razvijen je alat za testiranje ispravnosti generiranih pravila u odnosu na zadane politike te njihove raspodjele po vatrozidima u mreži.

Vatrozid, apstraktne pristupne politike, vatrozid pravila, automatsko podešavanje

Summary

Configuration and Testing of Firewalls Based on Given High-Level Policies

A firewall is a device or software that enables monitoring and management of network traffic between computer networks or their segments. Management is based on abstract access policies that are translated into a set of firewall rules, and each network packet is compared against these rules to determine whether it should be allowed or dropped. This paper describes a software solution that automatically translates abstract policies, defined in the form of a graph-based model along with a network description, into a set of firewall rules and distributes them to the specified firewall devices in the network. Additionally, a tool has been developed to test the correctness of generated rules with respect to the defined policies and their distribution across the firewalls in the network.

Firewall, abstract access policies, firewall rules, automated configuration

Skraćenice

IDS	Intrusion Detection System	sustav za otkrivanje upada
IPS	Intrusion Prevention System	sustav za sprječavanje upada
VPN	Virtual Private Network	virtualna privatna mreža
NAT	Network Address Translation	prijevod mrežne adrese
MAC	Media Access Control	kontrola pristupa medijima