Testiranje i analiza sustava za otkrivanje i sprječavanje napada

Podgoršek, Norman

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Norman Podgoršek

Testiranje i analiza sustava za otkrivanje i sprječavanje napada

ZAVRŠNI RAD

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Norman Podgoršek

JMBAG: 0016137116

Studij: Informacijski sustavi

Testiranje i analiza sustava za otkrivanje i sprječavanje napada

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Nikola Ivković

Varaždin, rujan 2022.

Norman Podgoršek

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Unutar ovog završnog rada obrađena je tema testiranja i analiziranja sustava za otkrivanje i sprječavanje napada. Najprije je opisano što to zapravo je sustav za otkrivanje i sprječavanje napada, kao i sam programski alat Snort koji je odabrani *Intrusion Detection System* (IDS) rada. Uz to obrađene su i funkcionalnosti Snort alata, kako je nastao, razlika u glavnim verzijama, načine rada Snorta, itd. Unutar praktičnog dijela rada odrađeno je postavljanje Snorta na Windows i Linux okruženja, struktura pravila Snorta, kao i pisanje vlastitih pravila u različitim okruženjima uz testiranje i analizu ostvarenog. EternalBlue je korišten kao primjer realne prijetnje sustavu, te je promatran način prevencije izvršenja ranjivosti. Rad je priveden kraju uz zaključak te korištene izvore.

Ključne riječi: analiza sustava za otkrivanje; testiranje sustava za otkrivanje; sprječavanje napada; Intrusion Detection System; IDS; Snort; Cisco; EternalBlue

Sadržaj

1. Uvod

Za početak promatranja sustava za otkrivanje i sprječavanje napada potrebno je znati što oni zapravo jesu, koja im je svrha i za što se koriste. U samoj srži, sustav za otkrivanje i sprječavanje napada (eng. *Intrusion Detection System*) je uređaj ili program koji je konfiguriran na željenoj mreži u svrhu praćenja zlonamjernih i neželjenih aktivnosti u obliku prometa na mreži [1]. Pomoću dobivenih povratnih informacija sustava za otkrivanje i sprječavanje napada moguće je odraditi kompletan snop aktivnosti od bilježenja do konkretnih mjera zaštite nadziranog sustava. Za potrebe rada promatra se alat Snort, jedan od mnogih NIDS (eng. *Network Intrusion Detection System*) alata koji se danas koriste za zaštitu osobnih i poslovnih sustava raznih veličina.

Snort je jedan od rijetkih sustava za otkrivanje i sprječavanje napada koji je „open source“ prirode, što znači da je programski kod javno dostupan za pregled i korištenje te svatko može napraviti svoju verziju programa i doprinijeti glavnoj verziji . Snort radi na osnovi pravila uz pomoću kojih se odrađuju željene akcije unutar mreže. Pravila mogu biti službena (Ciscova), kreirana od zajednice ili vlastita.

2. Sustavi za otkrivanje i sprječavanje napada

Sustavi za otkrivanje napada (eng. *Intrusion Detection System*) sami po sebi dolaze u raznim oblicima i konfiguracijama. Što znači, može biti program ili uređaj postavljen na željenoj mreži u svrhu praćenja zlonamjernih i neželjenih aktivnosti. [1] Kao što samo ime kaže, ovaj sustav je namijenjen samom otkrivanju, bilježenju i analizi dobivenog prometa nad promatranim sučeljem. Pakete koje program primi može označiti sumnjivima i to najčešće zbog košenja uspostavljenim pravilima. Sustavi također bilježe sve primljene pakete i izvještavaju administratora o mogućim prijetnjama putem konzole programa, grafičkog sučelja ili zapisa u vanjsku datoteku.

Sustavi za sprječavanje napada (eng. *Intrusion Prevention System*) su po mnogim karakteristikama slični sustavima za otkrivanje napada, no imaju iznimno ključnu dodatnu funkcionalnost odrađivanja željene aktivnosti ili akcije po zadobivenom i analiziranom prometu. Odnosno, u slučaju da sustav na promatranoj mreži ili na lokalnom računalu zapazi paket koji se ne slaže s postavljenim pravilima sustava (eng. *Signature Match*) ili ako se dogodi anomalija poput većeg od očekivanog broja paketa, mogu se zadati akcije koje pokreću određene mjere zaštite mreže. Moguće je implementirati prevenciju dolaska paketa na svoje željeno odredište, što podrazumijeva da je moguće postaviti da se takve situacije razriješe automatski bez eksplicitnog odobrenja administratora ili u slučaju da sustav nije siguran (te postoji prostor za pogreške) mogu se poslati administratoru na dodatan pregled.

Ovisno o željama ili potrebama budućeg klijenta koristio bi se jedan od navedenih sustava. Nakon odabira željenog sustava potrebno je odrediti hoće li sustav biti orijentiran na promatranje poslužitelja u sustavu (eng. *Host-based*) ili će promatrati zadanu mrežu (eng. *Network-based*).

2.1. Poslužiteljski i mrežno orijentirani sustavi

Poslužiteljski orijentirani sustavi ili programi izvršavaju se nad samim računalom korisnika na kojemu se nadzire promet. Također je moguće detaljnije kategorizirati poslužiteljske sustave na HIDS (eng. *Host-based Intrusion Detection System*) i HIPS (eng. *Host-based Intrusion Prevention System*). [2] Prednost individualnog, poslužiteljskog pristupa ovoj metodi jest taj da je dostupan uvid u dodatan promet koji se odvija lokalno na računalu za razliku od mrežnog pristupa gdje to nije moguće. No, nedostatak je zahtijevanje dodatnih resursa poput procesorskog vremena računala nad kojim se sustav pokreće, a ako je u pitanju veća mreža računala, poslužiteljski pristup nije skalabilan u usporedbi s mrežnim pristupom. Što znači da bi se za svako računalo trebalo individualno konfigurirati sustav koji bi nadzirao željeno računalo. Poslužiteljski pristup je generalno preferiran ako se nadzire manji broj sustava. Razlog tome je potreba velikog broja licenci za svako računalo nad kojim je potrebno primijeniti poslužiteljski pristup, kao i netemeljitost samog pristupa kad se radi o velikom broju računala.

Mrežno orijentirani sustavi mogu se izvršavati na posebno određenom uređaju unutar mreže koji pokreće sustav nadzora u obliku programa. Ipak, neki sustavi također mogu biti već integrirani kao proširenje ili značajka unutar mrežnih komponenti poput usmjernika. Mrežno orijentirani sustav nadolazeći promet preusmjerava sebi na analizu, testiranje i provjeru paketa. Nakon što je paket zaprimljen, prolazi kroz ustanovljena interna pravila te u slučaju da je prepoznato odstupanje ili anomalije u zadobivenim analiziranim paketima, paket koji nije klasificiran kao dozvoljen promet bit će odbačen te neće stići na svoju postavljenu destinaciju. S druge strane, paket za koji je sustav potvrdio ispravnost šalje usmjerniku koji dalje šalje promet odredišnom računalu. Naravno, dostupni su mnogi parametri koji se mogu koristiti kako bi se utjecalo na rigoroznost i učestalost provjera i odstupanja. Učinkovitost sustava mora biti testirana prije nego što je postavljen u pun pogon. Mrežno orijentirani sustavi mogu se detaljnije kategorizirati na NIDS (eng. *Network-based Intrusion Detection System*) i NIPS (eng. *Network-based Intrusion Prevention System*).

2.2. Načini detekcije prometa

Detekcija paketa unutar sustava, bilo da se koristi poslužiteljski ili mrežni pristup, može se odraditi na više načina.

Prva metoda je metoda pravila (eng. *Signature Detection System*) uz pomoću koje se može ustanoviti željena pravila unutar uređaja ili programa koja služe kako bi se jednoznačno odlučilo je li zadobiveni paket dozvoljen te može li nastaviti do svoje zadane destinacije ili ga je potrebno odbaciti s mreže. Ta komponenta tvori samu srž sustava za otkrivanje i sprječavanje napada. Set pravila za prihvat ili odbacivanje prometa dobavljaju se ovisno korištenom sustavu. Najčešće se biblioteke pravila preuzimaju od provjerenih izvora poput Cisca, no u mnogim slučajevima dostupna su i pravila koja uzdržava zajednica. Naravno, već kreiranoj listi mogu se dodati i vlastita, personalizirana pravila kako bi bilo moguće prilagoditi sustav vlastitim potrebama. Što bi značilo da ukoliko se baza pravila redovno osvježava (uz garanciju da je uvijek postavljena najnovija dostupna verzija pravila) ostvaruje se najažurnija sigurnost protiv napada na mrežu i sustav. Neki sustavi za otkrivanje i sprječavanje napada oko samih pravila orijentiraju metodu monetizacije svojeg proizvoda te se samim time naglašava izrazita važnost samih pravila za sustave otkrivanja i sprječavanja napada.

Također, određeni sustavi koriste i metodu anomalije (eng. *Anomaly-Based Detection System*) pomoću kojeg program analizira trenutno i uobičajeno stanje promatrane mreže. Zatim određuje kakav promet je, unutar ili van mreže, uobičajen kroz određeni vremenski period. Nakon analize koja se odrađuje unutar samog programa uspostavlja se kakav je promet dozvoljen, normalan i očekivan unutar sustava. Pomoću tih informacija može označiti određeni promet sumnjivim s različitom razinom sigurnosti (ili jačinom potencijalne prijetnje) te tako obratiti pozornost administratoru na problematičniji promet.

3. Osnovne informacije o Snortu

Snort je sustav „open source“ za sprječavanje napada u vlasništvu Cisca. Sadrži razne funkcionalnosti poput analize zadobivenog prometa u gotovo stvarnom vremenu i bilježenja dospjelih paketa na mreži te može razotkriti i djelovati protiv raznih napada, ranjivosti mreže i računala [3].

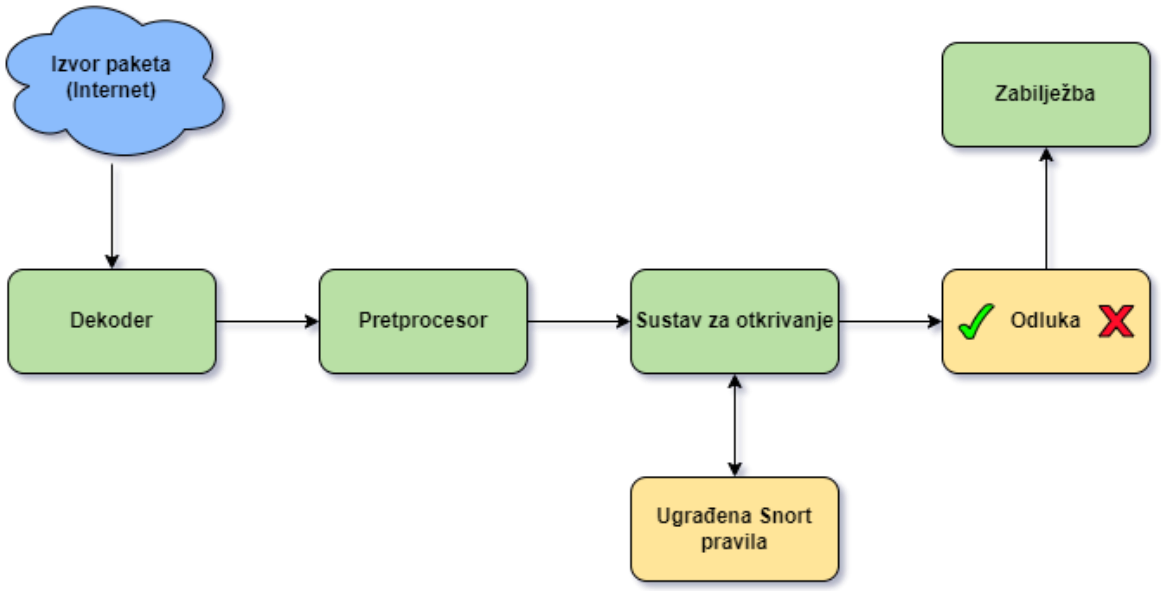
1998. godine, Snort je izdao vlasnik Sourcefirea Martin Roesch. U to vrijeme prozvan je nisko profilnim sustavom za sprječavanje napada koji je radio na Unix operacijskom sustavu i operacijskim sustavima baziranim na Unixu [4]. Napisan u C programskom jeziku, postao je popularan izbor sustava za sprječavanje napada zbog svojih značajki i mogućnosti prilagodbe sustavima klijenata. Snort je također od početka alat „open source“ prirode, što znači da je programski kod javno otvoren za pregled i doprinose. Prednost tog pristupa je da svatko može napraviti svoju „verziju“ programa tako da napravi novi *branch*, napravi željene doprinose te opisaše njihov doprinos kako bi doprinijeli glavnoj verziji programa.

2.0 verzija programa je izdana 2003. godine. Cisco kupuje Sourcefire 2013. godine u što je uključen i Snort alat te nastavlja rad na alatu sve do danas. Nakon cijelog desetljeća u razvoju u siječnju 2021. godine Cisco izdaje „open source“ verziju Snort 3.0. Verzija 3.0 donosi kompletno novi mehanizam prepoznavanja prometa i brojne druge promjene [5]. Verzija 3.0 napisana je u C++ programskom jeziku za razliku od C programskog jezika koji se koristio u prijašnjim izdanjima. Snort 3.X i dalje dobiva česta ažuriranja od kojih je zadnja verzija 3.1.36.0 izdana 16.7.2022.

3.1. Životni ciklus paketa

Ciklus života paketa započinje kada paket dolazi s izvora (npr. preko Interneta ili drugog računala unutar lokalne mreže) na sučelje nad kojim je Snort aktivan. Zadobiveni paket se prvo šalje dekoderu na dešifriranje sadržaja samog paketa. Odnosno, dekoderi određuju koje sve protokole paket koristi. To se ne odnosi samo na glavni protokol korišten unutar paketa poput TLS, TCP, DNS, DHCP, već i na Ethernet vezane podatke, kao i verziju IP protokola (npr. verzija 4 ili 6). Dekoder zatim traži moguće greške i neočekivane vrijednosti unutar zaglavlja analiziranih polja. Paket se zatim šalje pretprocesoru. Pretprocesor ima ulogu pripremiti zadobiveni paket za daljnju obradu kako bi ostatak sustava mogao obraditi paket u smislenom obliku. Kroz takvo oblikovanje podataka pretprocesor također eliminira određene zlonamjerne načine zaobilaženja zaštite putem sustava za otkrivanje i sprječavanje napada. Pretprocesor također ima mogućnost spoznaje određenih napada poput fragmentacije IP adrese.

Nakon što je paket obrađen šalje se na sam sustav za detekciju. Sustav za detekciju je zadužen za provođenje ugrađenih pravila Snorta. Pravila mogu biti pribavljena od samog Cisca ili zajednice, no mogu se nadodati i vlastita pravila. Zatim će sustav, uz sve znane informacije, donijeti odluku ostaje li paket na mreži te dopušta li mu se prolaz do zadane destinacije ili se odbacuje u slučaju da se pokazao zlonamjernim ili sumnjivim. Na kraju i po zadobivanju prometa, odrađivanja analize i donošenja odluke bilježi se odluka unutar datoteke ili kroz ispis unutar konzole.



Slika 1: Životni ciklus paketa u Snortu (izrada autora 2022.)

3.2. Razlike u verzijama Snort 2 i Snort 3

Snort danas dolazi u 2 izdanja: Snort 2 i Snort 3. Snort 2 je izdan 2003. godine, te je napisan u programskom jeziku C, dok je Snort 3 izdan 2021. godine i napisan u C++ jeziku što znači da je sam programski kod čitkiji i omogućava lakše dodavanje novih značajki. Snort 3 je u gotovo svim slučajevima nadogradnja na Snort 2.

Među najbitnijim novitetima su višedretvenost procesa, odnosno mogućnost da svaka procesorska dretva ima sposobnost istovremenog izvršavanja nekoliko analiza paketa, dok se u Snort 2 verziji može se obrađivati samo jedan paket po procesu. Uz novu mogućnost poboljšanog dijeljenja memorijskog prostora, uočava se veliki doprinos poboljšanoj skalabilnosti Snorta unutar verzije 3 u svrhu primjene na većim sustavima.

Način pisanja pravila je također izmijenjen kako bi pisanje vlastitih pravila i pregled napisanoga bio pregledniji i jednostavniji za krajnje korisnike. Unutar Snort 3 koristi se Lua standard pisanja što znači da postojeća Snort 2 pravila ne funkcioniraju na Snort 3. No, Cisco je razvio alat kako bi omogućio lak prijenos uspostavljenih pravila unutar Snort 2 na novi Lua način pisanja pravila u Snort 3 nazvan Snort2lua. Snort2lua kao unos prima Snort 2 formatirano pravilo te ga pretvara u Lua standard, što je zapravo Snort 3 kompatibilno pravilo [6].

Snort 3 također nudi moderniju arhitekturu koja pokriva neke od nedostataka Snort 2 poput bolje analize HTTPS/2 i QUIC protokola te Internet of Things (IoT) uređaja [7]. Navedena poboljšana također omogućuju razna dodatna poboljšanja kao što su manja potreba za ponovnim pokretanjem Snort programa, olakšano dodavanje novih mogućnosti, itd.

3.3. Načini rada Snorta

Snort ima tri glavna načina rada programa: „Sniffer Logger“ i „Network Intrusion Detection mode“.

„Sniffer mode“ je način rada unutar kojeg Snort ispisuje informacije o nadolazećem paketu unutar mreže. Snort uzima zaglavlje i srž samog paketa te ga ispisuje unutar pokrenutog sučelja, najčešće komandne linije. „Sniffer mode“ ima mnogo sličnosti s alatom tcpdump koji također odrađuje taj zadatak no Snort prezentira zadobivene informacije na pregledniji i efikasniji način.

„Logger mode“ radi na sličan način kao i „Sniffer mode“, no umjesto da prikazuje zaglavlja i srž paketa unutar komadne linije primljeni promet pohranjuje unutar datoteke u okviru specificiranog direktorija tijekom pokretanja Snort-a koja se zadaje kao argument ili kao zadani direktorij za logove unutar Snort konfiguracijske datoteke.

„Network Intrusion Detection mode“ je način rada koji pokreće Snort program kao mrežno orijentirani sustav za sprječavanje napada. NIDS način rada je najčešći razlog pokretanja Snorta. Kroz NIDS način rada Snort analizira zaprimljene pakete u gotovo stvarnom vremenu. Svu obavljenu analizu zabilježenih paketa Snort sprema se unutar zadanog direktorija, te šalje upozorenja o potencijalno nepoželjnom prometu i odbacuje neželjene pakete po pravilima koja su mu unaprijed dodijeljena. Svaki od načina rada mogu se dodatno prilagoditi korisničkim željama uz razne parametre koji mijenjaju krajnji rezultat izvršavanja Snorta. Neke od takvih mogućnosti jesu: filtriranje prometa kojeg je potrebno zabilježiti, ograničavanje broja informacija koje se prikazuju unutar komadne linije, korištenje određenog željenog mrežnog sučelja, itd.

3.4. Opasnosti napada i alternative Snortu

Sustavi za otkrivanje i sprječavanje napada bitniji su nego ikad jer su informacije o korisnicima traženije i osjetljivije. Svake godine broj uspješnih napada na mreže svijeta raste uz razmjere napada koji su sve veći, a sigurnost korisnika, njihovih podataka kao i poslovnih tajna poduzeća postaju sve veći prioritet velikih i malih poduzeća. U 2021. godini procijenjeno je da uspješni napad na mrežu poduzeća načini prosječnu štetu od 4.24 milijuna Američkih dolara [8]. Takve ogromne štete tjeraju poduzeća da ozbiljno shvate digitalnu sigurnost jer kada nastane napad na korisničke podatke, ne gube se samo sredstva, već i korisničko povjerenje u njihove usluge. Mnoga najveća poduzeća već su implementirala „Zero Trust“ pristup sigurnosti podataka. „Zero Trust“ pristup sigurnosti podrazumijeva da svaka autentikacija, pristup informacijama te akcija odrađena unutar sustava se bilježi te se pregledava legitimnost interakcije kroz svaki korak komunikacije. Ipak, napadi na mrežu ne nastaju samo mrežnim probojem nego i na ljudskoj grešci. Ukradene vjerodajnice zaposlenika zaslužne su za oko 19% napada na sustav, krađa identiteta kroz phishing oko 16% te pogrešna konfiguracija sustava oko 15% [8].

Snort nije jedini sustav za otkrivanje i sprječavanje napada, već postoje i modernije solucije, svaka s drugačijim mogućnostima. Među modernijim sustavima za otkrivanje i sprječavanje napada koje su također „open source“, postoje Suricata i Zeek. Jedna od glavnih nedostatka Snorta je strma krivulja učenja programa. Tome pridonosi nedostatak grafičkog korisničkog sučelja (GUI) te neintuitivno pokretanje programa prosječnim korisnicima.

Suricata nudi korisnicima moćno i moderno korisničko sučelje uz koje je moguće lakše pregledati ključne i relevantne informacije, kao i mogućnost statistike zaprimljenog prometa na definiranom sučelju u stvarnom vremenu. Iako ima drugačiju strukturu rada programa, Suricata je kompatibilna s već napisanim Snort pravilima. Također je omogućena višedretvenost, koju Snort dobiva tek u 3.0 verziji, kao i ubrzavanje rada programa uz pomoć rada grafičke kartice računala (Hardware Acceleration) [9]. Također omogućava hvatanja više prometa od samih paketa poput certifikata, DNS zahtjeva i sl.

Zeek, bivši Bro, radi na drugačiji način od Snorta na način da dobivene pakete pretvara u događaje uz pomoću kojih program zatim može pratiti pakete koji se kose pravilima i anomalijama unutar poslanih paketa. Zeek koristi Bro-Script uz pomoću kojeg je moguće kreirati zadatke koji služe automatizaciji ključnim dijelovima sustava [10]. Nudi ogromnu fleksibilnost uz mnoge dostupne mogućnosti analize prometa, no to je ujedno i sam problem Zeeka jer, iako je platforma moderna, namijenjena je za zahtjevnije korisnike poput sistemskih administratora velikog snopa računala.

4. Postavljanje Snorta

4.1. Postavljanje Snorta na Windows sustavu

Kako bi se omogućilo uspješno pokretanje Snort sustava na računalu s Windows operacijskim sustavom, potrebno je instalirati biblioteku za snimanje primljenih paketa. Ovisno o tome unutar koje verzije Windowsa je potrebno implementirati Snort, postoje dvije mogućnosti. WinPcap je biblioteka koju je razvio Riverbed Technology, poznati po svojim mrežnim alatima poput Riverbed Modelera, a može se koristiti kao biblioteka za snimanje primljenih paketa. No, WinPcap je prestao dobivati redovna ažuriranja 2013. godine s verzijom 4.1.3, te se njegovo korištenje jedino preporučuje korisnicima na sustavima Windows 8 i ranije.

Za sustave koji pokreću Windows 10 sustav preporučuje se korištenje Npcap biblioteke. Npcap je kreirao Gordon Lyon, koji je također razvio Nmap alat, s prvom javnom verzijom 2016 godine. Alat i dalje dobiva nova ažuriranja s kojim je zadnja verzija 1.70 koja je izdana 25.6.2022. U ovome slučaju odabran je Windows 10 sustav, zbog čega je pogodnija Npcap biblioteka. Zatim se dohvaća najnovije Snort izdanje te se odrađuje jednostavan proces instalacije Snorta na Windows sustav. Kako bi se olakšao proces postavljanje Snort datoteka, kao i pokretanje samog Snort sustava, korišten je korijenski direktorij lokalnog diska C kao glavni folder Snorta.

Nakon instalacije no prije rada sa samim programom neophodno je dobaviti pravila uz pomoću kojih je moguće pokrenuti Snortov sustav za sprječavanje napada. Kako bi se dobavila službena pravila, potrebno je napraviti korisnički račun na Snort web stranici. Uz besplatno dostupna službena pravila, moguće je i dobaviti pravila kreirana od zajednice Snorta. Nakon toga, preuzeta pravila unose se u direktorije „preproc_rules“ za pretprocesorska pravila i „rules“ za pravila vezana uz sustav za otkrivanje.

Uz pravila dodana, potrebno je i napraviti nekolicinu promjena unutar Snort konfiguracijske datoteke, snort.conf, kako bi Snort mogao ispravno raditi unutar zadanog okruženja. Odnosno, jer je snort.conf predkonfiguriran za Unix (Linux) okruženje, potrebno je ispraviti razlike između dviju platformi poput različitih, nepotpunih ili nepostojećih putanja.

```
44  # Setup the network addresses you are protecting
45  ipvar HOME_NET 192.168.1.0/24
46
47  # Set up the external network addresses. Leave as "any" in most situations
48  ipvar EXTERNAL_NET !192.168.1.0/24
```

Slika 2: Adresa željene mreže (izrada autora 2022.)

Prvo je potrebno unutar konfiguracijske datoteke unijeti IP adresu mreže koju je potrebno zaštititi. Navedenu IP adresu se može pribaviti na nekoliko načina, no najjednostavniji je preko naredbe „ipconfig /all“. Naredba prikazuje sva sučelja trenutno dostupna računalu. Na kraju IP adrese dodaje se sufiks /24 zato što se navedena IP adresa nalazi u C potklasi, što podrazumijeva Subnet masku 255.255.255.0.

```
104  var RULE_PATH c:\Snort\rules
105  var SO_RULE_PATH ../so_rules
106  var PREPROC_RULE_PATH c:\Snort\preproc_rules
```

Slika 3: Putanja pravila (izrada autora 2022.)

Unutar snort.conf datoteke usmjeravaju se varijable „RULE_PATH PREPROC_RULE_PATH“ da upućuju na ispravnu putanju kako bi ostatak konfiguracijske datoteke mogao uspješno inicijalizirati korištena pravila. „SO_RULE_PATH“ nije potrebno podešavati za Windows okruženje programa.

```
184  # Configure default log directory for snort to log to.
185  # For more information see snort -h command line options (-l)
186  config logdir: c:\Snort\log
```

Slika 4: Direktorij logova (izrada autora 2022.)

Ukoliko bi tijekom korištenja Snorta bilo potrebno da uz prikaz podataka o paketima budu i zabilježeni u nespecificiranom direktoriju, zapis se sprema u zadani direktorij koji se ovdje definira.

```
246  # path to dynamic preprocessor libraries
247  dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
248
249  # path to base preprocessor engine
250  dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
```

Slika 5: Putanje za pretprocesor (izrada autora 2022.)

Također je potrebno usmjeriti Snort prema ispravnim putanjama za „dynamicpreprocessor“ i „dynamicengine“ kako bi pretprocesorska komponenta mogla ispravno raditi uz priložene biblioteke.

```
113  var WHITE_LIST_PATH c:\Snort\rules
114  var BLACK_LIST_PATH c:\Snort\rules
```

Slika 6: Varijabla whitelist i blacklist (izrada autora 2022.)

```
511    whitelist $WHITE_LIST_PATH\whitelist.rules, \
512    blacklist $BLACK_LIST_PATH\blacklist.rules
```

Slika 7: Pozivanje varijabli whitelist i blacklist (izrada autora 2022.)

Iako ih se neće koristiti, također se moraju definirati putanje na „whitelisti“ i „blacklisti“. „Whitelist“ služi da se označi IP adrese željenima ili neželjenima. U slučaju da je kreiran unos IP adrese na „blacklisti“, paketi s navedenom IP adresom bit će odbačeni čim dođu na promatrano sučelje, preskačući sam proces analize paketa. U slučaju da je IP adresa dodana na „whitelist“, paketi s navedenom IP adresom bit će odobreni u slučaju da Snort smatra da je nadolazeći promet neželjen. Također treba provjeriti postojanje same „Whitelist“ datoteke u slučaju da nije prisutna. Datoteka se ne nalazi u novijim Snort izdanjima no potrebna je za uspješnu inicijalizaciju sustava te ju je potrebno ručno kreirati. Moguće je iskoristiti „Blacklist“ datoteku uz podešavanje parametra imena datoteka i sadržaja datoteke. Nanesene promjene unutar Snort.conf datoteke se sprema, te je Snort sada uspješno podešen za Windows okruženje i spreman za rad.

4.2. Postavljanje Snorta na Linux sustavu

Snort je moguće postaviti na Linux distribucije Fedora, Centros i FreeBSD pomoću ugrađenih programa za upravljanje paketa (eng. *Packet manager*), te mogu „povući“ Snort instalaciju i sve ostale programe o kojima Snort ovisi kako bi ispravno funkcionirao. Za ostale distribucije, Snort se može preuzeti i instalirati ravno iz izvora. Prije nego što se preuzme sam Snort, potrebno je preuzeti i postaviti DAQ biblioteku (eng. *Data Acquisition Library*). DAQ ima sličnu ulogu kao WinPcap i Npcap biblioteke na Windows sustavu, odnosno služi za uvoz i izvoz mrežnih paketa no dozvoljava novu razinu fleksibilnosti personalizacije. Pridobivene programske pakete DAQ i Snort potrebno je konfigurirati te zatim instalirati. Nakon uspješne instalacije, rezultat je Snort direktorij s već poznatom strukturom datoteka. Prije nego što je moguće započeti sa samim radom, potrebno je dodati pravila koji NIDS način rada Snorta provodi te podesiti Snort konfiguracijsku datoteku.

Kako bi se mogla nabaviti Ciscova pravila, potrebno je napraviti Snort korisnički račun. Nije potrebno ponovno preuzimati ponovno pravila ako su već dostupna od prijašnjih postavljanja, jedino u slučaju nove verzije pravila. Kao i u Windows sustavu, dodaju se direktoriji „rules“ i „preproc_rules“ u korijenski direktorij Snorta. Snort konfiguracijska datoteka je pisana uz Linux okruženje na umu, tako da u usporedbi s Windows okruženjem, većina parametra i putanja je postavljeno uz relativne putanje, te nije potrebno raditi promjene. Kako bi se postavila adresu koju je potrebno zaštiti, potrebno ju je saznati putem naredbe „ip addr“. „Ip addr“ je ekvivalent „ipconfig“ naredbi na Windows sustavu, odnosno ispisuje sva računalu vidljiva sučelja i njihove pojedinosti . Ključna, internetska IP adresa se nalazi pod „inet“. Zadobivenu IP adresu je potrebno unijeti u konfiguracijsku datoteku te je sve spremno za rad sa Snortom u Linux okruženju.

```
2: enp0s3: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
ate UP group default qlen 1000
    link/ether 08:00:27:d9:05:61 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 82723sec preferred_lft 82723sec
    inet6 fe80::223d:324b:95a0:506/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
norman@norman-VirtualBox:~/snort-2.9.20/rules$
```

Slika 8: Provjera adrese na sučelju (izrada autora 2022.)

```
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 10.0.2.15/24
46
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET !10.0.2.15/24
```

Slika 9: Adresa željene Linux adrese (izrada autora 2022.)

5. Praktični rad sa Snortom

Kroz praktični dio korištena su dva računala (stolno i prijenosno računalo) spojena unutar lokalne mreže, povezana putem D-LINK mrežnog preklopnika. Oba računala koriste Windows 10 operacijski sustav.

Programi korišteni su sam Snort kao sustav za otkrivanje i sprječavanje napada, Npcap biblioteka za snimanje paketa unutar Windows okruženja, Windows konzola, te program za pregled prometa Wireshark za detaljniju analizu snimljenog mrežnog prometa.

Za EternalBlue ranjivost korištena je prva javna verzija Windows 7 operacijskog sustava te Oracle VM VirtualBox za virtualizaciju navedenog sustava. Na računalu napadača korišten je Kali Linux, Linux distribucija za analizu digitalne i mrežne sigurnosti te izvršavanja raznih napada u svrhu iskorištavanja ranjivosti. Unutar Kali Linuxa korišten je alat Metasploit, program za iskorištavanje raznih ranjivosti nad željenim računalima ili sustavima.

5.1. Načini rada Snorta

Prije nego što je moguće zadati Snortu način rada, potrebno je znati nad kojim sučeljem ga je potrebno postaviti. Prilikom pozicioniranja unutar direktorija „/Snort/bin“ koristi se naredba „snort -W“ kako bi se ispisao popis svih Snortu dostupnih sučelja.

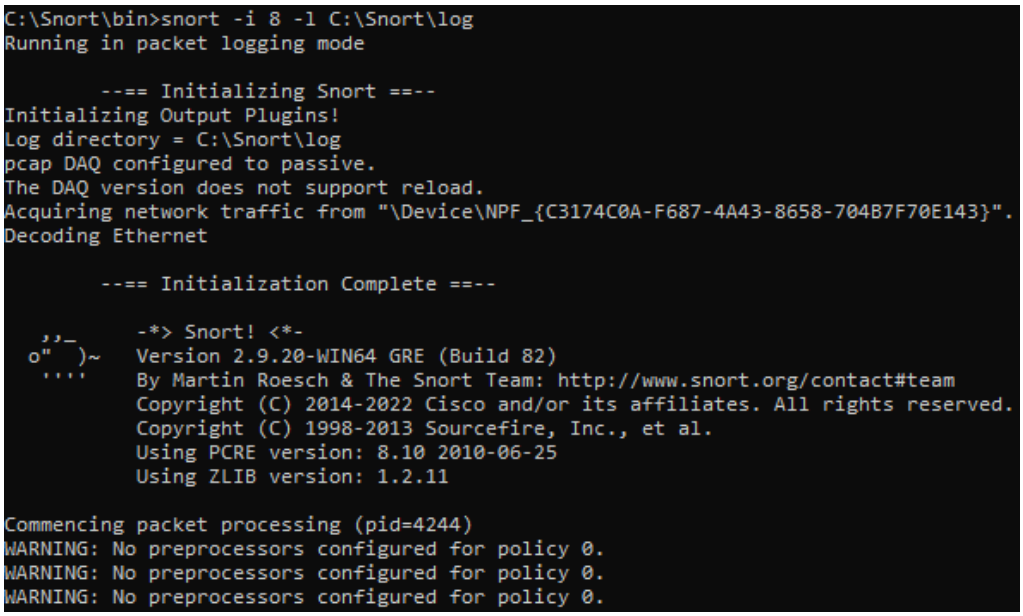
```
C:\Snort\bin>snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address        IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{259271DD-186E-4C96-A704-F784F3F0CEDD}      WAN Miniport (Network Monitor)
    2   00:00:00:00:00:00       disabled        \Device\NPF_{4DDF057A-F59B-4687-A70A-8EEFEB39646E}      WAN Miniport (IPv6)
    3   00:00:00:00:00:00       disabled        \Device\NPF_{D2FDC390-8409-4F82-B2FE-7530792B15C0}      WAN Miniport (IP)
    4   00:15:5D:5D:64:3C       172.29.0.1      \Device\NPF_{27FFA654-926C-4162-8C7C-079E686B4364}      Hyper-V Virtual Ethernet Adapter #2
    5   00:15:5D:E5:F5:36       172.31.32.1     \Device\NPF_{CE3A7AC9-DE36-449F-BC0D-18A18C041FA3}      Hyper-V Virtual Ethernet Adapter
    6   00:1A:7D:DA:71:13       169.254.184.115 \Device\NPF_{1C9FCBC3-C543-430B-9A67-AE0B0CD65485}      Bluetooth Device (Personal Area Network)
    7   0A:00:27:00:00:18       192.168.56.1    \Device\NPF_{FC21C3CB-75C3-4639-A8A9-0E3075AE855A}      VirtualBox Host-Only Ethernet Adapter
    8                           192.168.1.13    \Device\NPF_{C3174C0A-F687-4A43-8658-704B7F70E143}      Killer E3100G 2.5 Gigabit Ethernet Controller
    9   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture
   10   00:FF:15:26:3D:62       169.254.65.39   \Device\NPF_{15263D62-6E5D-47AC-9921-4654A35EF53B}      TAP-Windows Adapter V9
   11   00:00:00:00:00:00       169.254.126.147 \Device\NPF_{FC85ADFC-5D34-43A1-B953-E7DDED030F2A}      Wintun Userspace Tunnel
```

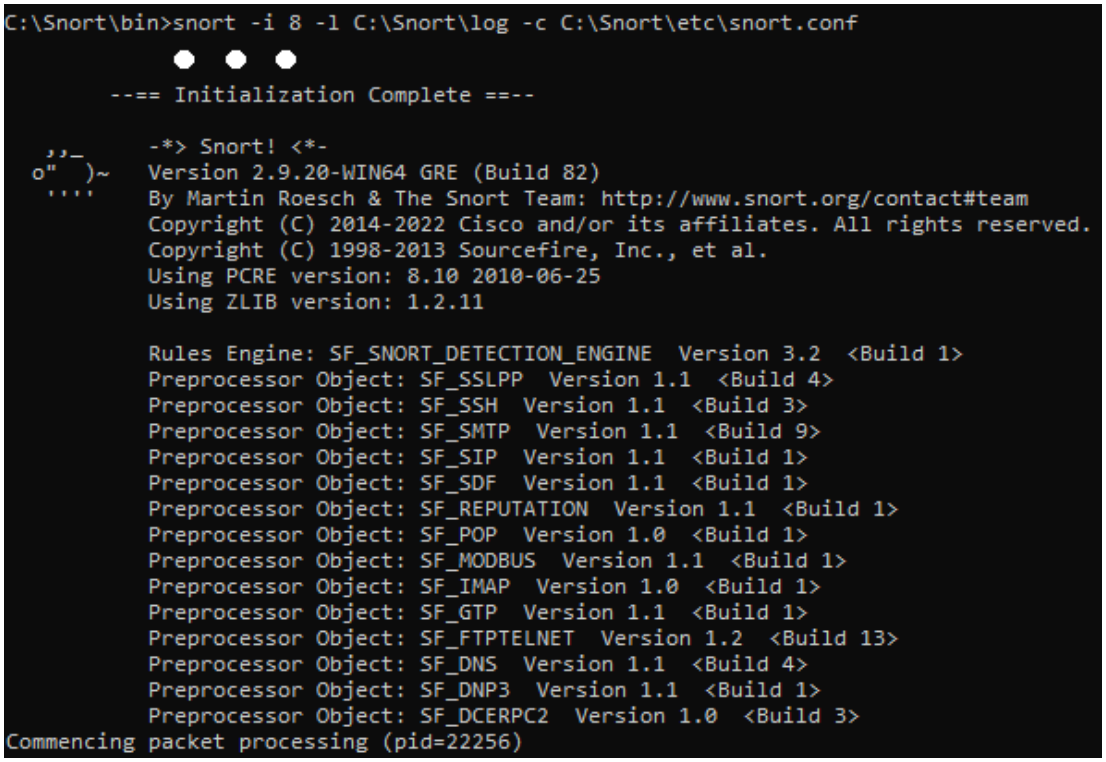
Slika 10: Popis sučelja (izrada autora 2022.)

Potrebno je razmotriti koje sučelje je potrebno promatrati, odnosno preko kojeg sučelja se računalo spaja s izvorom prometa kojeg je potrebno promatrati, u ovom slučaju internet. Zatim, prilikom pokretanja Snorta, potrebno je dodati argument -i kao i broj odabranog sučelja, npr. „-i 8“.



Slika 11: Ne unesena pretprocesorska pravila (izrada autora 2022.)

Također, prilikom pokretanja bilo kojeg načina rada, moguće je primijetiti upozorenja „WARNING: No preprocessors configured for policy 0.“. Upozorenja upućuju na činjenicu da pretprocesorska pravila nisu uspješno unesena, te da ih treba redefinirati unutar same komandne linije uz parametar -c i putanju do Snort.conf datoteke.



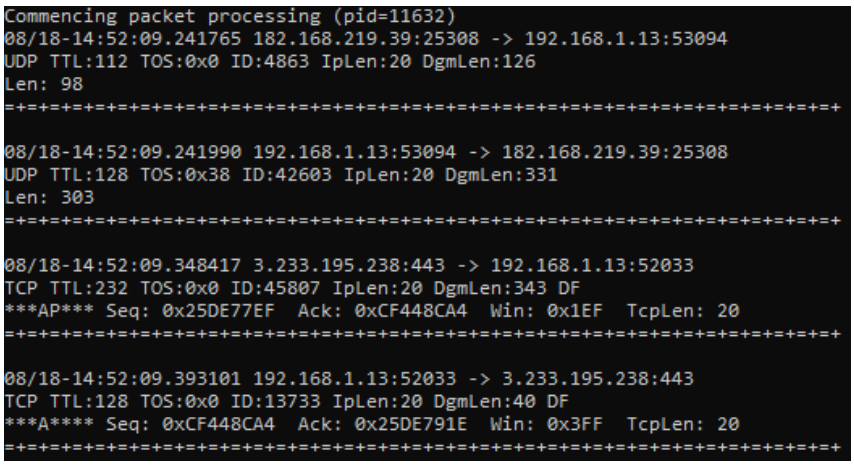
Slika 12: Unesena pretprocesorska pravila (izrada autora 2022.)

Nakon što se definira uz parametar -c putanja prema željenoj konfiguracijskoj datoteci, Snort povlači konfiguracijsku datoteku, te provjerava ispravnost svih unesenih parametara. U slučaju greške, dobiva se ispis na kojoj liniji unutar konfiguracijske datoteke je nastala greška. Ako greška nije nastala, pretprocesorska pravila se uspješno učitavaju te se upozorenje više neće pojavljivati.

5.1.1. Sniffer

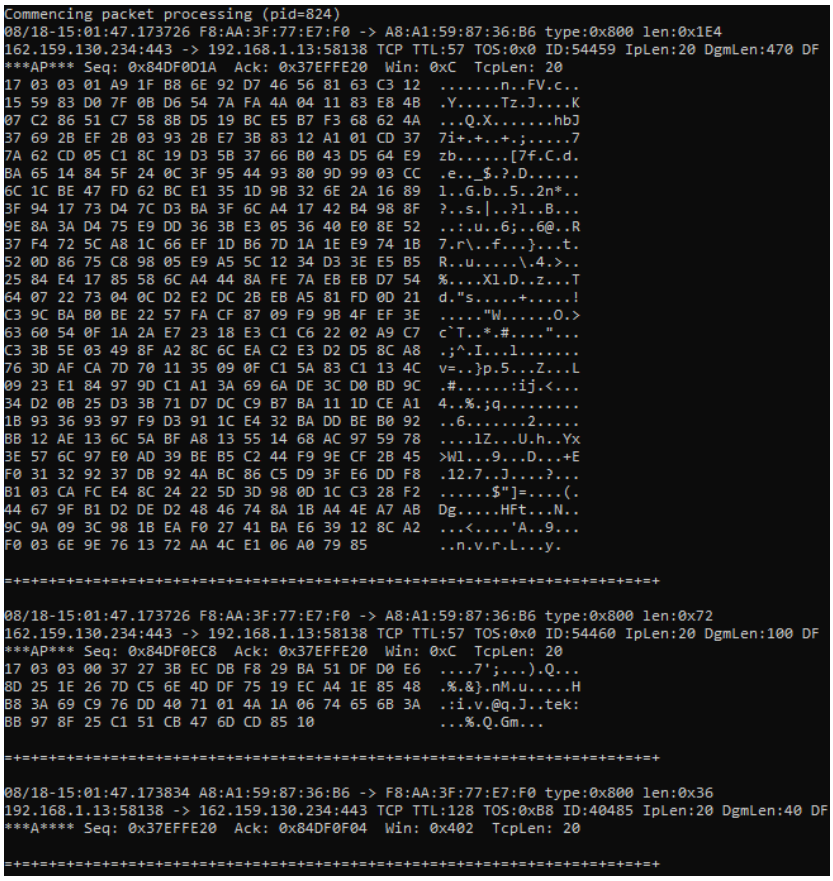
Unutar Sniffer načina rada Snorta ispisuje se nadolazeći promet nad promatranim sučeljem kroz komandnu liniju. Također postoji mogućnost i ispisa loga prema putanji zadanoj unutar konfiguracijske datoteke ili vlastiti direktorij. Prilikom pokretanja Snorta unutar komandne linije, postoji nekoliko parametara uz koje je moguće prilagoditi ispis prema zadanim potrebama.

Sniffer način rada se pokreće tako da se doda argument -v prilikom pokretanja programa. Tako se pokreće osnovni način rada pomoću kojeg Snort bilježi pakete TCP, IP, UDP i ICMP protokola.



Slika 13: Ispis Sniffer načina rada (izrada autora 2022.)

Moguće je vidjeti točan datum i vrijeme primitka i analize paketa, nad kojim sučeljem je zaprimljen paket i od kojeg pošiljatelja, koji protokol koristi, te ostale parametre poput TTL (eng. *Time To Live*). Ako je potrebno proširiti upit i uključiti pakete koje pripadaju prometu aplikacija dodajemo argument -d. Tako se može nadzirati i promet koje pokrenute aplikacije generiraju unutar sustava računala. No, u slučaju da je potrebno bilježiti pakete na još detaljniji način, koristimo -e parametar prilikom pokretanja Snorta. On omogućuje dodavanje pregleda zaglavlja od sloja veze podataka.

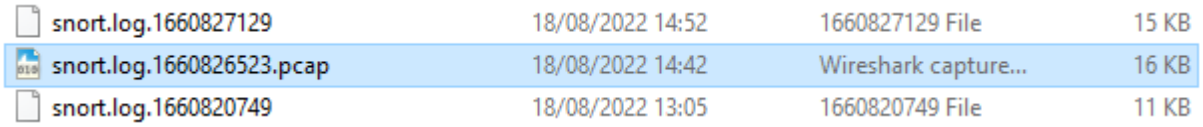


Slika 14: Detaljniji ispis sadržaja paketa (izrada autora 2022.)

5.1.2. Logger

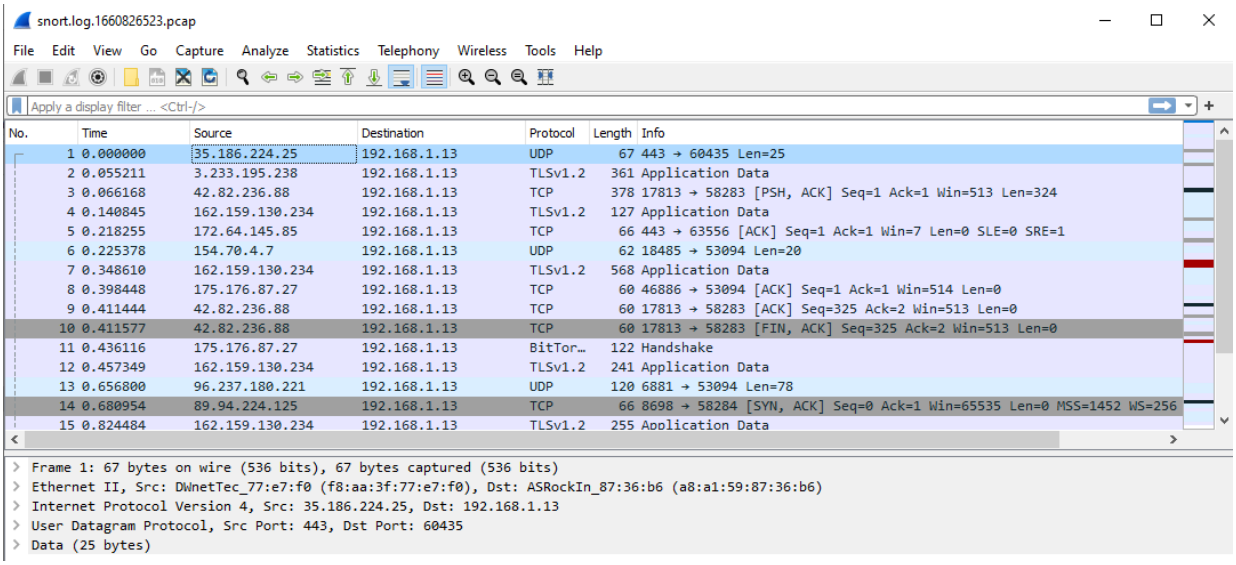
Logger način rada programa vrlo je sličan Sniffer načinu rada, no uz dodatnu komponentu zapisivanja svih paketa koji bi se ispisali na ekranu. Logger način rada se pokreće tako da se jednostavno doda uz sve ostale željene parametre -l argument te puna putanja do direktorija. Unutar Logger načina rada moguće je također koristiti iste parametre kao i kod Sniffer načina rada.

Nakon što se završi snimanje paketa pritiskom na CTRL+C kombinaciju tipki, zabilježeni paketi se spremaju u specificiranu putanju.



Slika 15: Ispravljanje nastavka datoteke (izrada autora 2022.)

Logovi se spremaju unutar datoteke bez funkcionalnog nastavka. Svaki od log datoteka završava s vremenskim žigom kada je snimanje paketa završilo u „Epoch“ obliku prikaza. Ukoliko se doda vlastiti nastavak.pcap na log datoteku, moguće je otvoriti log datoteku u pregledniku prometa po želji, npr. Wireshark.



Slika 16: Prikaz uhvaćenog prometa unutar Wiresharka (izrada autora 2022.)

Uz pomoću alata kao što je Wireshark moguće je na pregledan i jednostavan način detaljnije pregledati sav snimljeni promet, uz sve dodatne mogućnosti i informacije koje takav program omogućava.

5.1.3. Network Intrusion Detection System

NIDS načina rada je ono po čemu je Snort najviše poznat, unutar kojeg Snort analizira nadolazeći promet nad određenim sučeljem u gotovo stvarnom vremenu. NIDS način rada zahtijeva da su stavke poput pravila i konfiguracijske datoteke uspješno postavljene prije nego što se može započeti izvršavati NIDS način rada. Nakon što su pribavljena službena pravila i postavljena Snort konfiguracijska datoteka u skladu s vlastitim okruženjem, mogu se nadodati i vlastita pravila kako bi prilagodili NIDS korisnikovim potrebama.

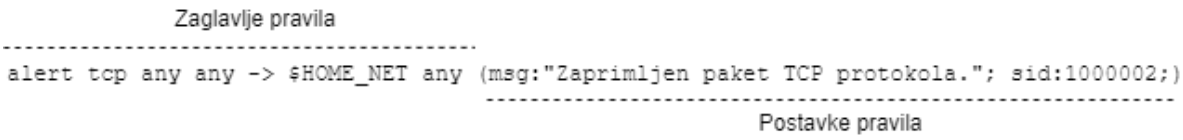
NIDS se koristi kada nije potrebno promatrati, analizirati i bilježiti cjelokupni promet koji dođe na promatranu mrežu, već je važnost pojedinačnog paketa određena po tome okida li se zadano pravilo po njegovom dolasku na sučelje.

Pravila mogu biti Ciscova službena pravila, pravila Snort zajednice koja imaju mnogo dodatnih pravila za blokiranje raznih ranjivosti i propusta sustava kako bi se sustav dodatno zaštitio. Također je moguće pisanje vlastitih pravila ukoliko dođe do potrebe za prilagođenim prijetnjama.

5.2. Pisanje vlastitih pravila

Uz Ciscova službena pravila kao i javno dostupna pravila, mogu se kreirati i vlastita, lokalna pravila kako bi omogućili veću fleksibilnost i prilagodbu zahtjevima korisnika. Vlastita pravila se uvoze unutar „local.rules“ datoteke, direktorija za pravila Snorta.

Prije nego što je moguće napisati vlastita pravila, potrebno je dobro poznavati njihovu strukturu. Pravila se sastoje od zaglavlja pravila i postavki pravila.



Slika 17: Anatomija Snort pravila (izrada autora 2022.)

Svako pravilo se sastoji od dva dijela: zaglavlje pravila i opcije pravila. Unutar zaglavlja pravila definira se glavna akcija koju je potrebno izvršiti nad paketom, te na koji protokol se akcija odnosi. Zatim se definira izvorišna i odredišna adresa i port. Unutar opcija pravila dodaju se svi ostali potrebni parametri, poput sadržaja poruke, sadržaja paketa kojeg se pretražuje kao i ID pravila.

5.2.1. Zaglavlje pravila

Unutar zaglavlja pravila prvo je potrebno definirati akciju koju je potrebno izvršiti kada se naleti na zadani paket. Mogu se koristiti parametri poput „alert“ koji pokreću zadani proces obavještavanja korisnika o paketu, „log“ uz kojeg se samo bilježi nadolazeći paket, „pass“ uz koji ignorira zaprimljeni paket i drugi. Također postoji i nekoliko načina za odbacivanje paketa, ovisno o potrebama. Primjerice, „drop“ parametar odbacuje i bilježi odbačeni promet, te „sdrop“ koji odbacuje nadolazeći promet no ne bilježi odbačeni promet.

Zatim je potrebno definirati na koji protokol se odnosi pravilo. Postoje četiri IP protokola koje Snort podržava odnosno ima mogućnost analize za sumnjive ili zlonamjerne aktivnosti: TCP, UDP, ICMP i sam IP [11]. Ovisno o vrsti neželjenog prometa s kojim se susreće, koristi se jedan od četiri protokola.

Svaki paket ima svoje izvorište i odredište. Izvorište podrazumijeva računalo ili poslužitelja koje je poslalo paket, a odredište je računalo ili poslužitelj koje prima poslani paket. Izvorište i odredište imaju IP adresu i port. Unutar ovih parametara moguće je definirati od kojih pošiljatelja je potrebno promatrati pakete ili nad kojim primateljima je potrebno odraditi analizu prometa. Moguće je i postaviti „any“ vrijednost nad bilo kojom IP adresom i portom, što znači da ako se stavi za izvorište parametre any any, očekuje se promet s bilo koje IP adrese i porta. Također je moguće koristiti IP adrese definirane unutar

Snort konfiguracijske datoteke poput „$HOME_NET“ za vlastitu adresu. Samom pravilu je moguće nadodati znak „<>“ kako bi Snort pratio obje strane komunikacije, što podrazumijeva poslane i primljene pakete na promatranom sučelju.

5.2.2. Postavke pravila

Unutar postavki pravila, moguće je dodati mnogo dostupnih parametara za hvatanje različitih vrsta prometa. Jedan od njih je i „msg“ parametar koji dopušta ispis željenog teksta unutar konzole te u sam log koji se generira. Željeni tekst se unosi u par dvostrukih navodnika. Zadnji parametar unutar pravila je SID, „Signature ID“, te uz njega jednoznačno se određuje nadodano pravila uz drugih. Što se tiče vlastitih pravila, prvih 100 vrijednosti su rezervirane, a vrijednosti od 100 - 999999 su za službena Snort pravila što znači da pri pisanju vlastitih pravila treba krenuti od vrijednosti 1000000 na dalje.

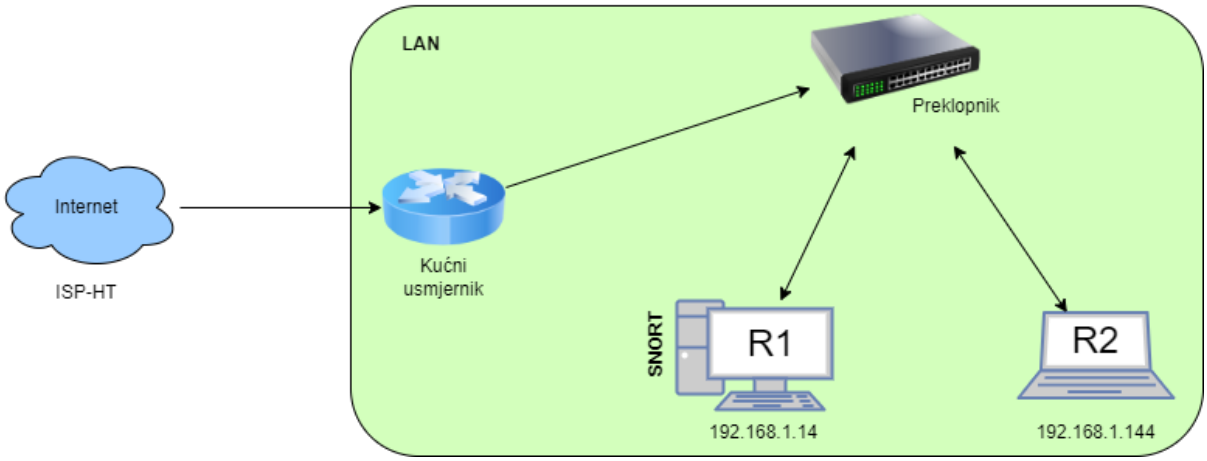
Postoje i druge postavke pravila koje su korisne u hvatanju određenog prometa, ovisno o tome s kakvim problemima se susreće. Pomoću parametra „content“ Snort provjerava sadržaj samog zaprimljenog paketa. Uz „content“, moguće je postaviti i da ako se uneseni string i sadržaj paketa podudaraju da se nadolazeći paket ukloni s mreže. Ako je potrebno ograničiti veličinu sadržaja paketa koji se prima te tako spriječiti „buffer overflow“ napad na mrežu može se koristiti „Dsize“. On omogućuje zadavanje maksimalne veličine unutar definiranja pravila. „Dsize“ prima „int“ vrijednost, te je moguće koristiti i operatore veće od i manje od (>, <) za veću fleksibilnost parametra.

Postoje i alati uz pomoću kojih postoji mogućnost blokiranja prometa web stranica u skoro stvarnom vremenu. Ta funkcionalnost može se postići uz React. React omogućuje fleksibilnu reakciju na promet koje je pravilo uhvatilo. Najčešći način korištenja je blokiranje pristupa željenim stranicama određenim korisnicima na mreži uz „block“ parametar. No, postoji mogućnost korištenja „msg“ ili „warn“ parametra kako bi se administratora obavijestilo o mogućim problematičnim web stranicama pomoću zadane string vrijednosti.

Također postoje parametri „rev“ i „classtype“ koji ne utječu na izvršenje samog pravila već su tu za kategorizacijske potrebe. Uz „rev“ (revision) se definira verziju pravila nad kojim se trenutno radi, a u slučaju da dođe do poboljšanja pravila moguće je inkrementirati verziju. „Classtype“ se koristi za kategoriziranje pravila u skupine zbog lakše organizacije skupina pravila.

5.3. Testiranje vlastitih pravila

Za testiranje IDS funkcionalnosti Snort-a, povezana su dva računala: R1 i R2, preko Cat5E žice (Ethernet) na preklopnik.



Slika 18: Topologija za testiranje pravila (izrada autora 2022.)

5.3.1. Ping

Kako bi se moglo znati vide li se računala R1 i R2 uopće unutar lokalne mreže treba odraditi ping test. Pingovi koriste ICMP protokol za slanje paketa, te je to vrsta prometa koju je moguće uhvatiti preko Snort-a.

```
alert icmp 192.168.1.144 any -> $HOME_NET any (msg:"Zaprimljen ping od
racunala."; sid:1000001;)
```

Pomoću ovog pravila odrađuje se akcija alert obavještavanja o nadolazećem paketu. U slučaju paketa ICMP protokola, provjerava se je li pošiljatelj računalo R2, odnosno 192.168.1.144, s bilo kojeg porta, te je li pošiljatelj početna adresa, $HOME_NET (192.168.1.14) računala R1. U slučaju da je pravilo zadovoljeno poruka se ispisuje i zabilježava. Snort se pokreće uz:

```
snort -i 9 -c c:\Snort\etc\snort.conf -l C:\Snort\log -A console
```

Uz Snort pokrenut na računalu R1, započinje se proces pinga na R2.

```
C:\Windows\system32>ping 192.168.1.14

Pinging 192.168.1.14 with 32 bytes of data:
Reply from 192.168.1.14: bytes=32 time<1ms TTL=128
Reply from 192.168.1.14: bytes=32 time<1ms TTL=128
Reply from 192.168.1.14: bytes=32 time<1ms TTL=128
Reply from 192.168.1.14: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Slika 19: Uspješan ping (izrada autora 2022.)

Paketi su uspješno došli do odredišta, te primili odgovor od računala na adresi. Ako se pogleda računalo R2 moguće je vidjeti da se pravilo uspješno izvršilo četiri puta koliko je i ping poslan. Unutar konzole moguće je vidjeti datum i vrijeme analize paketa, sve do pete decimale sekunde. Nakon toga se ispisuje SID (Signature ID) pravila, pošto je broj veći od 1000000 odmah govori da se radi o vlastito kreiranom pravilu. Na kraju se ispisuje protokol paketa te smjer prometa, u ovom slučaju od računala R2 prema računalu R1.

```
Commencing packet processing (pid=25524)
08/20-21:11:05.023724  [**] [1:1000001:0] Zaprimljen ping od racunala. [**] [Priority: 0] {ICMP} 192.168.1.144 -> 192.168.1.14
08/20-21:11:06.034805  [**] [1:1000001:0] Zaprimljen ping od racunala. [**] [Priority: 0] {ICMP} 192.168.1.144 -> 192.168.1.14
08/20-21:11:07.043949  [**] [1:1000001:0] Zaprimljen ping od racunala. [**] [Priority: 0] {ICMP} 192.168.1.144 -> 192.168.1.14
08/20-21:11:08.052371  [**] [1:1000001:0] Zaprimljen ping od racunala. [**] [Priority: 0] {ICMP} 192.168.1.144 -> 192.168.1.14
```

Slika 20: Zaprimljen ping promet (izrada autora 2022.)

Log je spremljen na željenom odredištu te ako se promijeni ekstenzija datoteke u .pcap, moguće je log otvoriti u bilo kojem programu za analizu prometa i paketa, npr. Wireshark. Wireshark nudi mnogo detaljnije informacije o svakom paketu.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.144	192.168.1.14	ICMP	74	Echo (ping) request id=0x0001, seq=44/11264, ttl=128 (no response found!)
2	1.011081	192.168.1.144	192.168.1.14	ICMP	74	Echo (ping) request id=0x0001, seq=45/11520, ttl=128 (no response found!)
3	2.020225	192.168.1.144	192.168.1.14	ICMP	74	Echo (ping) request id=0x0001, seq=46/11776, ttl=128 (no response found!)
4	3.028647	192.168.1.144	192.168.1.14	ICMP	74	Echo (ping) request id=0x0001, seq=47/12032, ttl=128 (no response found!)

Slika 21: Ping promet unutar Wiresharka (izrada autora 2022.)

5.3.2. UDP i DNS

Snort podržava protokol UDP što znači da omogućava hvatanje paketa DNS protokola. DNS promet se najčešće provodi na sučelju broj 53. U slučaju da je potrebno odraditi zabranu pristupa određenim web stranicama unutar mreže, pa tako i DNS poslužiteljima zadanih web stranica, moguće je koristiti Snort kako bi se napravilo pravilo koje se pokreće nad pojavom zadanih DNS paketa. Iduća pravila definirana su unutar local.rules datoteke:

```
1. alert udp any any <> $HOME_NET 53 (content:"youtube"; within:100; msg:"YouTube DNS promet detektiran!"; sid:1000008; rev:1;)
2. alert udp any any <> $HOME_NET 53 (content:"steamcommunity"; within:100; msg:"Steam DNS promet detektiran!"; sid:1000009; rev:1;)
3. alert udp any any <> $HOME_NET 53 (msg:"Ostali DNS promet detektiran!";sid:90000002;)
```

Sintaksa govori da se pošalje alert kada paket UDP protokola s bilo koje adrese i bilo kojeg sučelja dođe preko sučelja 53 na promatranu mrežu. Zatim unutar content polja definira se sadržaj kojeg se pretražuje unutar svakog UDP paketa na sučelju 53. U slučaju prosječnog DNS paketa, koji su generalne veličine oko 50-75 bajtova, što znači da je moguće postaviti parametar „within“ i postaviti njegovu vrijednost na 100 kako bi pretražili samo prvih 100 bajtova primljenog paketa za uneseni „content string“. Svako pravilo mora imati unikatan „Signature ID“, te „revision“ nije potreban no poželjno ga je napomenuti u slučaju da se radi o novijoj ili starijoj verzija pravila. Za potrebe testa odabrani su YouTube i Steam kao pokusne web stranice uz dodatno pravilno za ostali UDP/DNS promet.

```
Commencing packet processing (pid=22168)
08/24-00:38:54.130031  [**] [1:1000009:1] Steam DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:52498
08/24-00:38:54.130031  [**] [1:90000002:0] Ostali DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:52498
08/24-00:38:54.142138  [**] [1:1000009:1] Steam DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:51818
08/24-00:38:54.142138  [**] [1:90000002:0] Ostali DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:51818
08/24-00:38:54.153894  [**] [1:1000009:1] Steam DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:53986
08/24-00:38:54.153894  [**] [1:90000002:0] Ostali DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:53986
08/24-00:38:54.942629  [**] [1:1000008:1] YouTube DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:64745
08/24-00:38:54.942629  [**] [1:90000002:0] Ostali DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:64745
08/24-00:38:54.954705  [**] [1:1000008:1] YouTube DNS promet detektiran! [**] [Priority: 0] {UDP} 1.1.1.1:53 -> 192.168.1.10:60678
```

Slika 22: Zaprimljen DNS promet (izrada autora 2022.)

Prilikom početka testa, otvorene su web stranice YouTube i Steam te se navedeni promet odmah prikazao unutar Snort konzole.

```
1.471675    1.1.1.1    192.168.1.10    DNS    138 Standard query response 0x4b9f AAAA steamcommunity.com SOA ns1.valvesoftware.com
1.471675    1.1.1.1    192.168.1.10    DNS    138 Standard query response 0x4b9f AAAA steamcommunity.com SOA ns1.valvesoftware.com
2.260410    1.1.1.1    192.168.1.10    DNS    365 Standard query response 0x2f61 A www.youtube.com CNAME youtube-ui.l.google.com A
2.260410    1.1.1.1    192.168.1.10    DNS    365 Standard query response 0x2f61 A www.youtube.com CNAME youtube-ui.l.google.com A
```

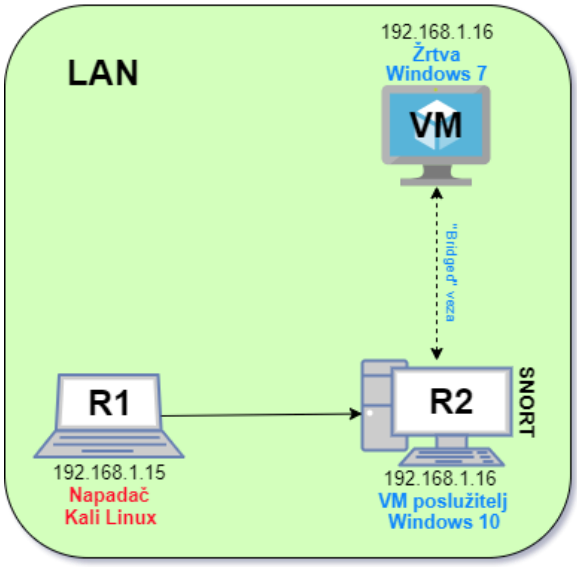
Slika 23: DNS promet unutar Wiresharka (izrada autora 2022.)

Promet je moguće otvoriti uz Wireshark, te promotriti detalje primljenih paketa.

5.4. EternalBlue ranjivost

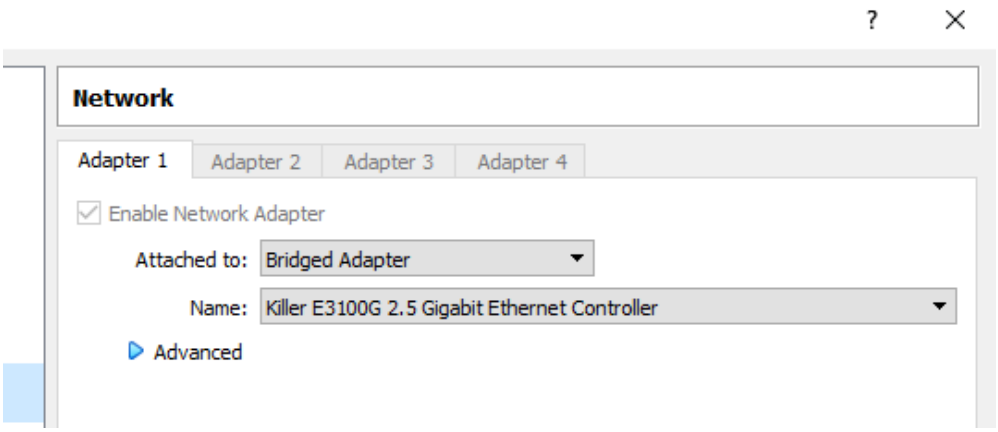
EternalBlue je računalna ranjivost koju je razvila Američka NSA (National Security Agency), no ista je procurila u javnost od strane hakerske skupine „Shadow Brokers“ travnja 2017, samo mjesec dana nakon što je Windows kreirao ažuriranja za ranjivost. Što znači da vrlo velik broj računala nije imalo najnovija ažuriranja te su bili ranjivi napadu [12]. EternalBlue koristi „SMBv1“ (Server Message Block verzija 1) protokol kao ulaznu točku napada; protokol koji je namijenjen kao način komunikacije i dijeljenje podataka između uređaja poput drugih računala i printera. Radi na način da tijekom uspostave „SMBv1“ komunikacije također se šalje i zlonamjeran paket uz pomoću kojeg je moguće dobiti povišene privilegije na ciljanom sustavu. Sigurnosna ažuriranja bila su potrebna za sve tada dostupne verzije Windowsa: Windows Vista, 7, 8.1, 10, Sever 2008, Sever 2012, Sever 2016. No zbog vrlo opasne prirode ranjivosti, Microsoft je ažurirao i više nepodržane verzije Windowsa kao Windows XP, Server 2003 i 8.

Kako bi se testirao pravi napad na mrežu, odabrana je EternalBlue ranjivost zbog njegove vrlo široke i opasne primjene, što može ozbiljno oštetiti starije i neodržavane sustave.



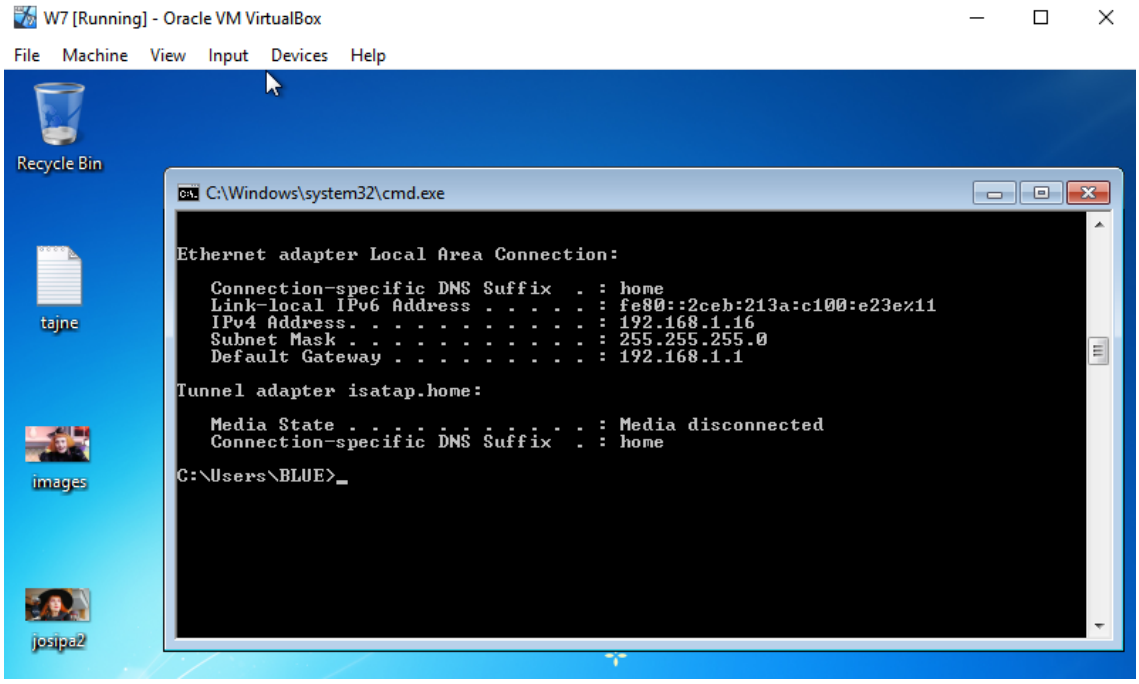
Slika 24: Topologija za testiranje EternalBlue ranjivosti (izrada autora 2022.)

Računalo R1 je napadač na virtualno okruženje Windows 7 koje se pokreće na računalu R2. Virtualno okruženje Windows 7 je verzija prvog službenog izdanja operacijskog sustava iz 2009. godine tako da je ranjiv na EternalBlue ranjivost i pogodan za testiranje.



Slika 25: Bridged način rada (izrada autora 2022.)

Računalo R2 i virtualno okruženje Windows 7 povezani su preko „bridged“ veze, što znači da će virtualno okruženje dijeliti isto fizičko sučelje kao i računalo R2 te po prirodi i njegovu IP adresu.



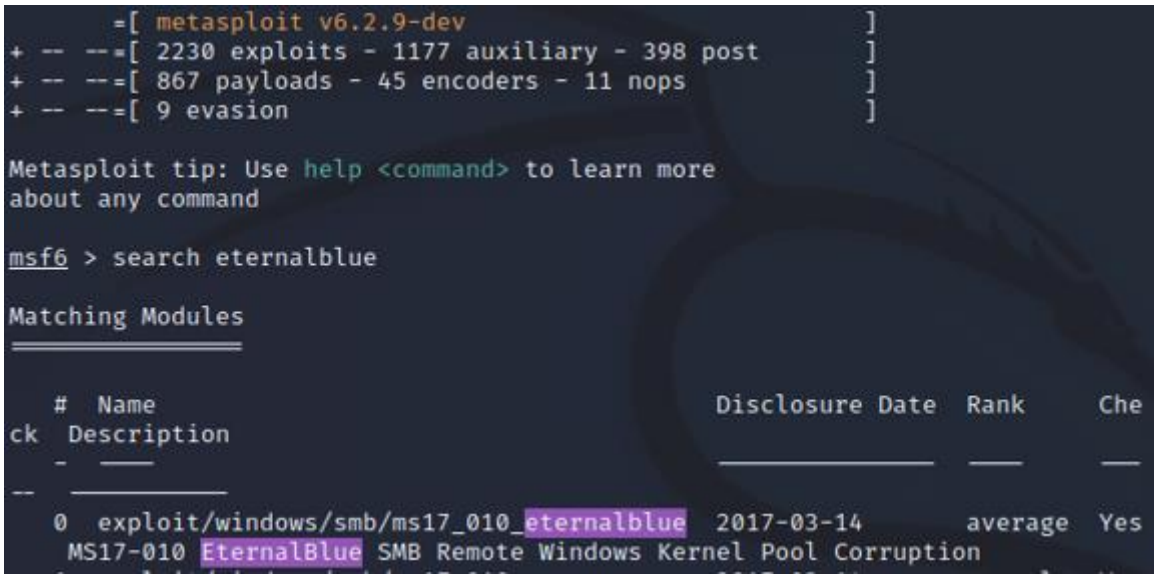
Slika 26: Osjetljive datoteke i ipconfig na računalu žrtvi (izrada autora 2022.)

Unutar Windows 7 sustava se potvrđuje zadobivanje iste IP adresu. Bezazlen korisnik može sadržavati osjetljive podatke već na samoj pozadini. U svrhu pokusa, kreirano je nekoliko datoteka za koje je cilj napadača preuzeti na svoje računalo.

Kako bi se napad spriječio, potrebno je definirati pravilo unutar Snorta kako bi takav nepoželjni promet odbacio. Na svu sreću, Cisco je vrlo brzo otkrio način kako odbaciti tu vrstu prometa te su kreirali pravilo koje se koristi tijekom ovog pokusa. Signature ID pravila je 1-41978. Snort se pokreće na računalu R2 da promatra vlastito sučelje 192.168.1.16.

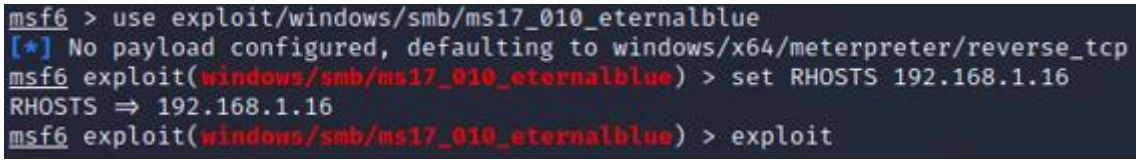
Kako bi se pokrenuo proces napada potreban je sustav koji može iskoristiti tu ranjivost. Korišten je Kali Linux, svjetski poznata Linux distribucija za analizu digitalne i mrežne sigurnosti te iskorištavanja ranjivosti.

Unutar Kalija pokreće se program Metasploit alat pomoću kojeg će se odrađivati penetracija sustava.



Slika 27: Pretraživanje EternalBlue modula (izrada autora 2022.)

Prvo je potrebno pronaći modul vezan za ranjivost EternalBlue te se koristi ključna riječ search za pronalazak svih dostupnih modula.

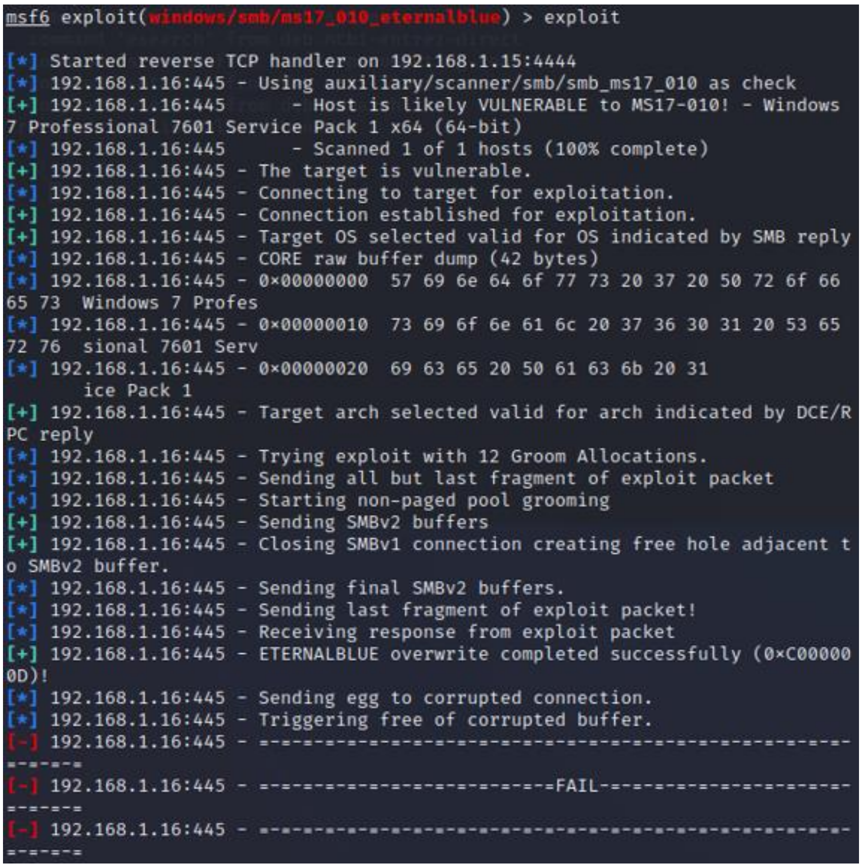


Slika 28: Pokretanje EternalBlue modula na adresu (izrada autora 2022.)

Zatim se koristi ključna riječ „use“ za odabir pronađenog modula. Modul se tada učitava u Metasploit, te je potrebno zadati sve parametre potrebne za izvršavanje ranjivosti. U slučaju EternalBlue ranjivosti, pošto su računala već na istoj lokalnoj mreži, jedino je potrebno proslijediti IP adresu uređaja na mreži. To se odrađuje uz ključnu riječ „set RHOSTS“ te se unosi IP adresa računala žrtve.

5.4.1. Uspješno blokiranje EternalBlue ranjivosti

U slučaju da je Snort pokrenut i ispravno konfiguriran na poslužitelju pokretanje modula ranjivosti EternalBlue neće se uspješno izvršiti do kraja te će naići na grešku tijekom izvođenja.

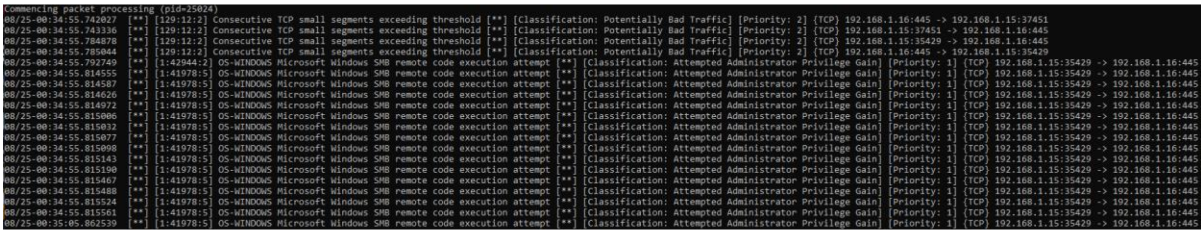


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.1.15:4444
[*] 192.168.1.16:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.16:445      - Host is likely VULNERABLE to MS17-010! - Windows
7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.16:445      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.16:445 - The target is vulnerable.
[*] 192.168.1.16:445 - Connecting to target for exploitation.
[+] 192.168.1.16:445 - Connection established for exploitation.
[+] 192.168.1.16:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.16:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.1.16:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66
65 73  Windows 7 Profes
[*] 192.168.1.16:445 - 0×00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65
72 76  sional 7601 Serv
[*] 192.168.1.16:445 - 0×00000020  69 63 65 20 50 61 63 6b 20 31
       ice Pack 1
[+] 192.168.1.16:445 - Target arch selected valid for arch indicated by DCE/R
PC reply
[*] 192.168.1.16:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.16:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.16:445 - Starting non-paged pool grooming
[+] 192.168.1.16:445 - Sending SMBv2 buffers
[+] 192.168.1.16:445 - Closing SMBv1 connection creating free hole adjacent t
o SMBv2 buffer.
[*] 192.168.1.16:445 - Sending final SMBv2 buffers.
[*] 192.168.1.16:445 - Sending last fragment of exploit packet!
[*] 192.168.1.16:445 - Receiving response from exploit packet
[+] 192.168.1.16:445 - ETERNALBLUE overwrite completed successfully (0×C00000
0D)!
[*] 192.168.1.16:445 - Sending egg to corrupted connection.
[*] 192.168.1.16:445 - Triggering free of corrupted buffer.
[-] 192.168.1.16:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
=-=-=-=
[-] 192.168.1.16:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-FAIL-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=
[-] 192.168.1.16:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
=-=-=-=
```

Slika 29: Neuspješni EternalBlue modul (izrada autora 2022.)

Pogreška se dogodila jer je Snort uspješno blokirao pokušaj uspostavljanja povišenih privilegija pristupa. To je moguće zaključiti jer unutar Snort konzole, kao i logove, pokušaj napada je uspješno zabilježen.



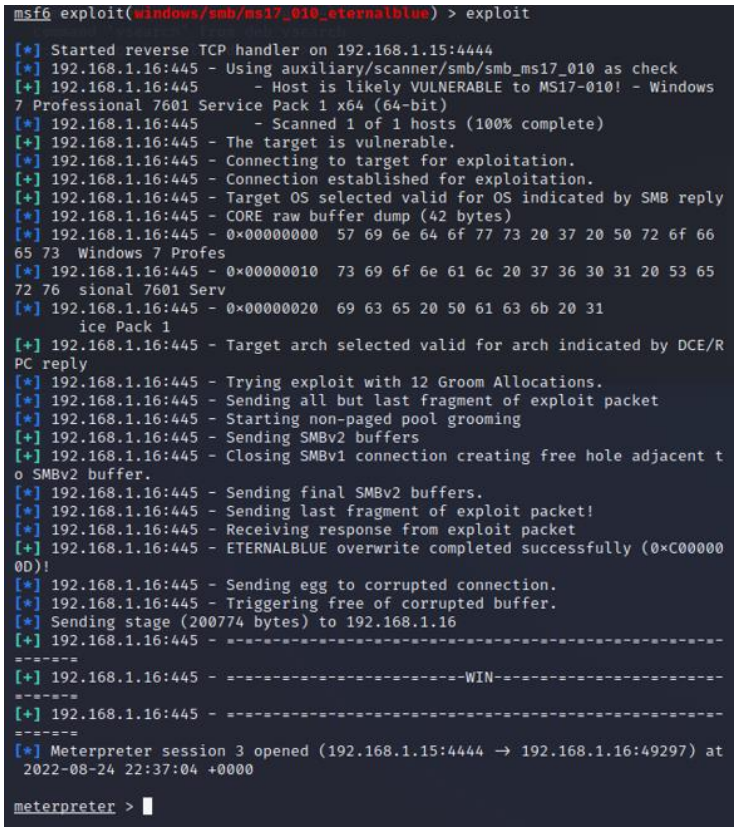
```
Commencing packet processing (pid=25024)
08/25-00:34:55.742027  [**] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.16:445 -> 192.168.1.15:37451
08/25-00:34:55.743336  [**] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.15:37451 -> 192.168.1.16:445
08/25-00:34:55.784878  [**] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.785044  [**] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 192.168.1.16:445 -> 192.168.1.15:35429
08/25-00:34:55.792749  [**] [1:42944:2] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.814555  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.814587  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.814626  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.814972  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815006  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815032  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815077  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815098  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815143  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815190  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815467  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815488  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815524  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:34:55.815561  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
08/25-00:35:05.862539  [**] [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.1.15:35429 -> 192.168.1.16:445
```

Slika 30: Uhvaćeni i spriječen EternalBlue (izrada autora 2022.)

Također je moguće zaključiti da Metasploit ne pokušava samo jednom izvršiti EternalBlue modul nego mnogo puta; u slučaju da je uspjeh ovisan o manjem postotku uspješnosti izvedbe modula ranjivosti. Kao i uz sve Snort logove, moguće je otvoriti snimljeni promet unutar Wiresharka kako bi se dobilo više informacija o pokušaju napada.

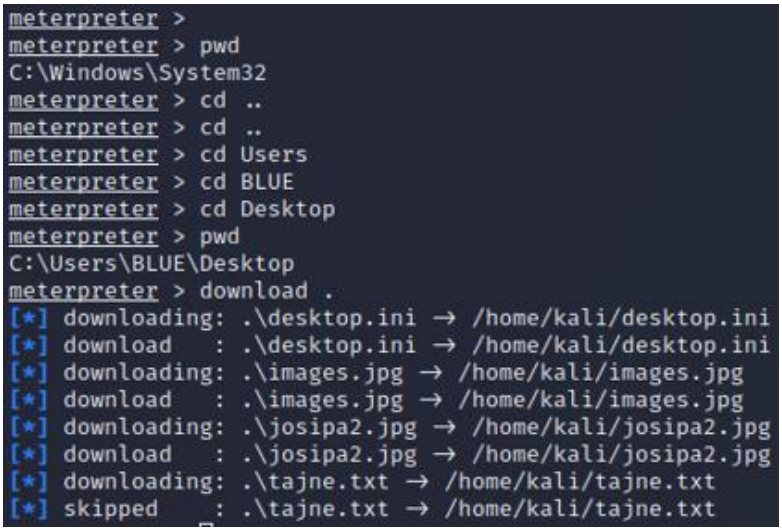
5.4.2. Neuspješno blokiranje EternalBlue ranjivosti

U slučaju da Snort nije pokrenut nad željenom mrežom ili nije ispravno konfiguriran modul EternalBlue će se uspješno izvršiti te će se ostvariti povišeni pristup računalu žrtve.



Slika 31: Uspješan EternalBlue modul (izrada autora 2022.)

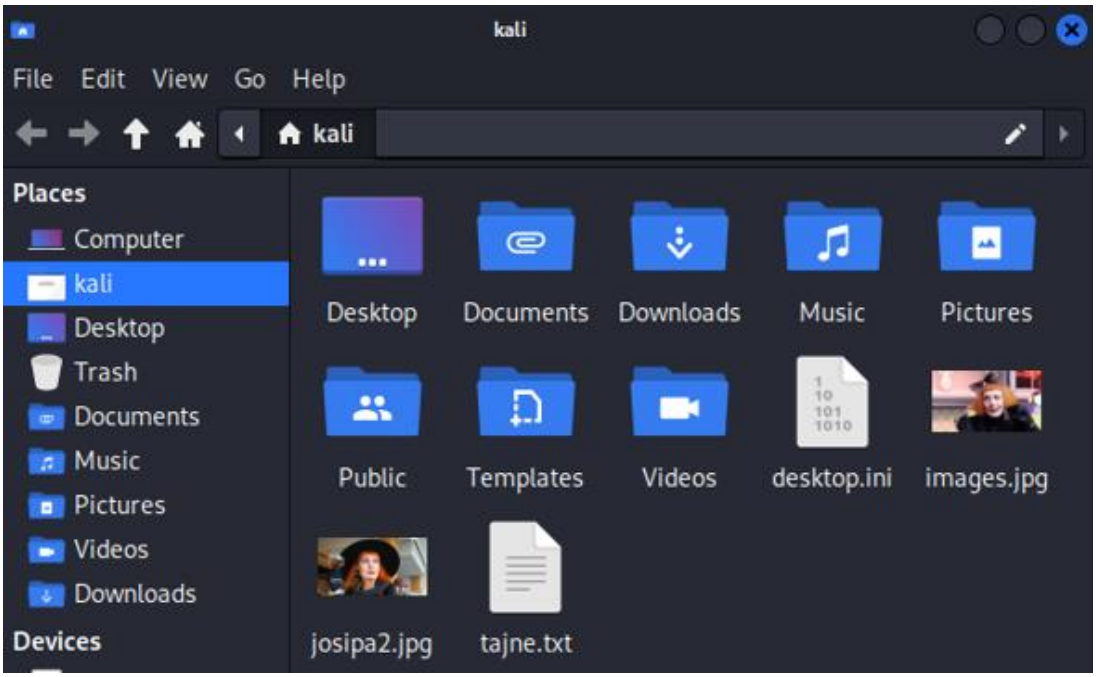
Modul odmah pri početku izvršavanja uspješno identificira metu kao ranjivu, te nastavlja dalje s izvršavanjem dok se uspješno ne izvrši.



Slika 32: Pozicioniranje i preuzimanje datoteka (izrada autora 2022.)

Nakon uspješnog izvršavanja pozicionira se u sustav datoteka računala žrtve i to unutar direktorija System32 koji se nalazi u sistemskom direktoriju Windows. Kroz ovaj način pristupa dostupna je mogućnost izvršavanja bilo kakve radnje nad računalom žrtve te žrtva nema nikakav pokazatelj da se zlonamjerne aktivnosti događaju u pozadini. Neke od jednostavnijih mogućih akcija jesu „cat“ za prikazivanje sadržaja datoteka, „pwd“ za provjeravanje trenutne putanje i „cd“ za promjenu trenutno pozicioniranog direktorija. No također su dostupni i „download“ uz pomoću kojeg je moguće preuzeti željene datoteke s računala žrtve i „upload“ za prijenos datoteka s računala napadača na računalo žrtve, zatim „execute“ za izvršavanje naredbi i akcija nad računalom žrtve, „hashdump“ uz pomoću kojeg zlonamjeran korisnik može dobiti vjerodajnice trenutnog korisnika te čak i mogućnost korištenja web kamere žrtve u zlonamjerne svrhe.

Nakon toga se pozicionira uz naredbu „cd“ na korijen diska, unutar direktorija trenutnog korisnika pa do direktorija radne površine unutar kojega su ciljane datoteke. Pomoću „pwd“ naredbe potvrđuje se trenutna putanja, te uz „download“ naredbu preuzimaju se datoteke s radne površine računala žrtve. „Download“ naredbi je dodan parametar „.“ kako bi poručili da je želja povući sve programe iz trenutnog direktorija odnosno s radne površine žrtve. Prijenos datoteka započinje te u vrlom kratkom vremenu prijenos je dovršen, a datoteke su uspješno prenesene na računalo napadača.



Slika 33: Potvrda primitka datoteka (izrada autora 2022.)

Dogodi li se ovakva situacija u poslovnom okruženju, šteta može biti kobna za poduzeće, zbog čega vrijedi investirati u ispravno postavljen sustav za otkrivanje ili sprječavanje napada.

5.5. PulledPork, Snorpy i Snowl GUI

Snort je moguće proširiti kroz razne alate i dodatke poput PulledPork, Snorpy i GUI-a za Snort. PulledPork je aplikacija za upravljanje pravilima unutar Snorta, kao i njemu sličnom no modernijem alatu Suricata. PulledPork ima razne korisne mogućnosti za poboljšanje Snort iskustva, no najbitnije su automatizirano preuzimanje najnovijih verzija pravila, checksum provjera legitimnosti preuzetih pravila, preuzimanje pravila iz drugih repozitorija, itd.

Snorpy je web stranica uz pomoću koje je moguće putem grafičkog sučelja kreirati pravilo po potrebama. Sintaksa Snort pravila može biti kompleksna na prvu te Snorpy pomaže s razumijevanjem sintakse i mogućnosti odabira određenih parametara preko padajućeg izbornika u usporedbi s korištenjem dokumentacije. Kao primjer, rekreirano je pravilo za praćenje DNS prometa unutar Snorpyja.

Slika 34: Snorpy web aplikacija (izrada autora 2022.)

Snort također ima i nekoliko dostupnih grafičkih sučelja. Snowl je najpopularnije komercijalno grafičko sučelje za Snort. Snowl nudi moderno i intuitivno korisničko sučelje s pomoću kojim se može znatno olakšati rad sa Snortom. Također nudi mogućnosti vizualizacije prikupljenih logova, automatizirane radnje po pojedinačnim prijetnjama te jednostavnu instalaciju [13]. Snort je također imao i „open source“ grafička sučelja, što znači da je bilo tko imao mogućnost koristiti i doprinijeti projektu u svrhu poboljšanja iskustva no nažalost više se ne održavaju s novim izdanjima Snorta.

6. Zaključak

Sustavi za otkrivanje i sprječavanje napada su bitniji nego ikad za male i velike primjene, a cijena informacija i podataka je sve veća kako sustavi sadrže sve više informacija o korisnicima. Kako programi i aplikacije postaju sve kompleksniji, tako i mogućnost velikih ranjivosti unutar sustava raste. Svako poduzeće treba ozbiljno shvatiti mrežnu sigurnost. Ako već ne za prevenciju prijetnji, imati sustav koji prijavljuje bilo kakav sumnjivi promet unutar mreže može biti od iznimne važnosti. Ovisno o veličini sustava nad kojim se želi postaviti sustav, može se koristiti poslužiteljski ili mrežni pristup od kojeg se poslužiteljski koristi kada je potrebno nadzirati manji broj računala, a mrežni ako imamo veći sustav s kompleksnijom topologijom.

Tu dolazi Snort kao jedan od starijih, no nimalo manje relevantnih IPS-a. No Snort je potrebno i ispravno konfigurirati budući da sustavi za otkrivanje i sprječavanje nisu iznimno korisni ako se ne poprime pravila koja je potrebno provoditi. Ciscova pravila i pravila zajednice daju vrlo snažan temelj sigurnosti mreže, no u slučaju da je potrebno pisati vlastita pravila moguće ih je napisati po volji uz opsežne mogućnosti Snorta. Uz pomoć Cisca i Snorta, vrlo kobne i rasprostranjene ranjivosti poput EternalBlue i Log4j mogu se brzo zakrpati unutar željenog sustava samo uz preuzimanje najnovijih pravila. No, nije potrebno sve raditi ručno; PulledPork omogućuje automatizirano preuzimanje pravila što osigurava da je implementirana najnovija dostupna zaštita. Snorpy web sučelje je moguće koristiti za pojednostavljenje procesa pisanja vlastitih pravila.

Bez obzira koristi li se Snort, Suricata ili Zeek, praćenje i ograničavanje prometa unutar mreže nije jedini način neželjenog pristupa sustavu; pristup se može steći i krađom vjerodajnica ili čak i ljudskom greškom, no imati osiguranu mrežu je ključni dio svakog odjela sigurnosti poduzeća, velikog ili malog.

Popis literature

[1] Intrusion detection system, [Na internetu] Dostupno na: https://en.wikipedia.org/wiki/Snort_(software) [Pristupljeno 28-lip-2022]

[2] IDS vs IPS, [Na internetu], Dostupno na: https://cybersecuritykings.com/2021/05/25/ids-vs-ips-tips-on-nids-hids-nips-and-hips/ [Pristupljeno 13-kol-2022]

[3] Snort - Open-Source Network Intrusion Detection & Prevention System, [Na internetu] Dostupno na: https://www.talosintelligence.com/snort [Pristupljeno 07-kol-2022]

[4] Snort and the Value of Detecting the Undetectable, [Na internetu], Dostupno na

https://www.techopedia.com/2/28294/security/snort-and-the-value-of-detecting-the-undetectable [Pristupljeno 11-kol-2022]

[5] Snort 3 Adoption, [Na internetu], Dostupno na: https://secure.cisco.com/secure-firewall/docs/snort-3-adoption [Pristupljeno 13-kol-2022]

[6] The Snort Team, Snort 3 User Manual, 3.1.29. izd. Cisco: Cisco Talos Intelligence Group

[7] Comparing Snort 2 and Snort 3 on Firepower Threat Defense – Cisco, [Na internetu], Dostupno na: https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/217617-comparing-snort-2-and-snort-3-on-firepow.html [Pristupljeno 14-kol-2022]

[8] Cost of a Data Breach Report 2022, [Na internetu], Dostupno na: https://www.ibm.com/security/data-breach [Pristupljeno 21-kol-2022]

[9] Top 6 Free Network Intrusion Detection Systems (NIDS) Software in 2022 | UpGuard, [Na internetu], Dostupno na:

https://www.upguard.com/blog/top-free-network-based-intrusion-detection-systems-ids-for-the-enterprise [Pristupljeno 21-kol-2022]

[10] 2021 Open Source IDS Tools: Suricata vs Snort vs Bro (Zeek), [Na internetu], Dostupno na: https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview [Pristupljeno 21-kol-2022]

[11] Writing Snort Rules, [Na internetu], Dostupno na: https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm#rule_header

[12] EternalBlue, [Na internetu], Dostupno na:

https://en.wikipedia.org/wiki/EternalBlue [Pristupljeno 22-kol-2022]

[13] Snowl, Snort-GUI, [Na internetu], Dostupno na:

https://snowl.io/ [Pristupljeno 23-kol-2022]

Popis slika