

Pregled sustava za detekciju/prevenciju upada

Duvnjak, Ilija

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:959976>

Rights / Prava: [Attribution 3.0 Unported](#)/[Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2025-03-18**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Ilija Duvnjak

**PREGLED SUSTAVA ZA
DETEKCIJU/PREVENCIJU UPADA**

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ilija Duvnjak

Matični broj: 36037-07-I

Studij: *Primjena informacijske tehnologije u poslovanju*

Pregled sustava za detekciju/prevenciju upada

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Igor Tomičić

Varaždin, rujan 2020.

Ilija Duvnjak

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski 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/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Razvijanjem informacijskih sustava kao i razvijanjem internetske infrastrukture te selidbom svih podataka na mreže nastala je potreba za zaštitom naših podataka, podataka o poslovanju te zaštitom cijelog sustava radi neometanog poslovanja. Kao solucija za zaštitu su napravljeni sustavi za detekciju i prevenciju upada. Danas postoje mnogi sustavi za detekciju/prevenciju upada neki od njih su komercijalni a neki su besplatni i otvorenog koda. U ovom radu obrađujemo najkorištenije sustave danas te kroz praktični dio rada postavljamo te konfiguriramo svoj sustav za detekciju upada Snort. Prikazujemo način na koji se pišu osobna pravila u tom sustavu te prikazujemo kako se pokreće program, budući da nema grafičko sučelje. I na kraju ga kroz virtualno okruženje testiramo. Na kraju rada pokazujemo kakve vrste pravila neki od tih napada pokreću.

Ključne riječi: Sustav, Detekcija, Prevencija, Snort, Informacijska sigurnost., IDS, IPS

Sadržaj

| | |
|--|-----------|
| 1. UVOD | 1 |
| 2. METODE I TEHNIKE RADA | 2 |
| 3. SUSTAVI ZA DETEKCIJU/PREVENCIJU UPADA | 3 |
| 3.1. POVIJEST SUSTAVA ZA DETEKCIJU/PREVENCIJU UPADA | 3 |
| 3.2. PODJELA SUSTAVA | 5 |
| 3.2.1. <i>Podjela sustava za detekciju upada</i> | 5 |
| 3.2.1.1. Sustav za detekciju upada baziran na hostu (HIDS) | 5 |
| 3.2.1.2. Sustav za detekciju upada bazirani na mreži(NIDS) | 6 |
| 3.2.1.3. Podjela sustava prema načinu otkrivanja napada | 7 |
| 3.2.2. <i>Sustav za prevenciju upada</i> | 9 |
| 3.2.2.1. Podjela sustava za prevenciju upada | 10 |
| 4. PREGLED BESPLATNIH I KOMERCIJALNIH SUSTAVA | 12 |
| 4.1. BESPLATNI SUSTAVI ZA DETEKCIJU/PREVENCIJU UPADA | 12 |
| 4.1.1. <i>SURICATA</i> | 12 |
| 4.1.2. <i>OSSEC</i> | 13 |
| 4.2. KOMERCIJALNI SUSTAVI ZA DETEKCIJU/PREVENCIJU UPADA | 14 |
| 4.2.1. <i>Cisco Stealthwatch</i> | 14 |
| 4.2.2. <i>SolarWinds Security Event Manager</i> | 15 |
| 4.3. USPOREDBA SLOBODNO DOSTUPNIH SUSTAVA | 17 |
| 4.3.1. <i>Pravila</i> | 17 |
| 4.3.2. <i>Izvođenje procesa</i> | 17 |
| 4.3.3. <i>Primjena</i> | 18 |
| 4.3.4. <i>Snort 3.0</i> | 18 |
| 5. PRAKTIČNI DIO RADA | 19 |
| 5.1. SNORT | 19 |
| 5.1.1. <i>Packet sniffer mode</i> | 20 |
| 5.1.2. <i>Packet logger mode</i> | 21 |
| 5.1.3. <i>Intrusion detection mode</i> | 22 |
| 5.2. PRAVILA | 24 |
| 5.3. INSTALACIJA I KONFIGURACIJA SNORTA | 26 |
| 5.3.1. <i>Instalacija i konfiguracija na Windowsu</i> | 26 |
| 5.3.2. <i>Instalacija i konfiguracija na Linuxu</i> | 29 |
| 6. TESTIRANJE SNORTA | 31 |
| 6.1. POSTAVLJANJE VIRTUALNOG OKRUŽENJA | 31 |
| 6.2. TESTIRANJE DETEKCIJE SNORTA | 32 |
| 6.2.1. <i>Testiranje pomoću alata NMAP</i> | 33 |
| 6.2.2. <i>Testiranje pomoću alata Legion</i> | 34 |
| 6.2.3. <i>Testiranje pomoću alata Nikto</i> | 35 |
| 7. ZAKLJUČAK | 37 |
| POPIS LITERATURE | 38 |
| POPIS SLIKA | 41 |

1. Uvod

Kako se razvija informacijska tehnologija i povećava se količina podataka na mrežama i na internetu, nastala je potreba za zaštitom svojih podataka i podataka o poslovanju radi boljeg poslovanja. Kako bi se organizacije zaštitile od upada ili napada, napravljeni su sustavi koji detektiraju kada netko želi pristupiti našim podacima ili kada netko pristupa našoj mreži. Kao i svugdje u svijetu postoje sustavi koji se plaćaju, koji obično imaju više funkcionalnosti, ali postoje i programi otvorenog koda koji su slobodno dostupni i koji se razvijaju unutar zajednice.

Ovaj rad opisuje povijest sustava za detekciju/prevenciju upada te njihove podjele. Nakon toga se opisuje nekoliko slobodno dostupnih programa. U ovom radu smo opisali programe Suricata i OSSEC te Snort u praktičnom dijelu rada. To su programi koji se često koriste danas. Nakon toga smo opisali 2 programa komercijalne prirode Cisco Stealthwatch i Solarwinds security event manager. Zatim, kao praktični dio rada, smo pobliže objasnili jedan program za detekciju upada Snort, opisali smo način na koji se instalira konfigurira te smo ga na temelju par testova pomoću Kali linuxa testirali. Koristili smo alate već postavljene unutar Kali Linuxa Nmap, Legion te Nikto.

2. Metode i tehnike rada

U ovom radu će se prvo teorijski obraditi povijest te podjela sustava kao i neki od sustava koji se najviše koriste danas, a nakon toga će se u praktičnom dijelu pobliže opisati te postaviti sustav za detekciju upada Snort. Za praktični dio rada će se koristiti Oracle VM VirtualBox unutar kojega ćemo postaviti 3 virtualna računala za testiranje, svaki od njih će imati drugačiji operacijski sustav. Prvo računalo će imati instaliran Ubuntu i bit će naše računalo pomoću kojeg pokrećemo Snort te će se prikazati kako postaviti te konfigurirati Snort na Ubuntu. Drugo računalo će imati metasploitable2 OS koje ćemo iskoristiti u zadnjem testu. Treće računalo će imati instaliran Kali operacijski sustav s alatima za testiranje pomoću kojeg ćemo testirati Snort i njegovu sposobnost detekcije upada.

3. Sustavi za detekciju/prevenciju upada

U ovome poglavlju opisani su opći pojmovi vezani za temu kao što su otkrivanje napada i sprječavanje napada. Također, navedena je i povijest sustava za detekciju/prevenciju upada, a objašnjene su i različite vrste sustava za detekciju/prevenciju.

3.1. Povijest sustava za detekciju/prevenciju upada

Sustavi za detekciju upada (IDS) i sustavi za prevenciju upada (IPS) su započeli na Stanfordovom institutu za razvoj (SRI) kao pokušaj da razviju ekspertni sustav za detekciju upada (engl. *Intrusion Detection Expert System*) (IDES). IDES je koristio statističku detekciju anomalija, potpise te profile korisnika i poslužitelja kako bi detektirao maliciozna mrežna ponašanja. IDES je mogao detektirati ako se protokol ne koristi za ono za što je namijenjen te je također mogao detektirati napad uskraćivanja usluge (DOS). [2][40]

Početak 2000-ih IDS je postao najbolja praksa u sigurnosti. Kroz 90-te su se koristili vatrozidovi koji su bili tada efektivni za prijetnje tih godina. Vatrozidovi procesiraju podatke brzo te samo imaju sposobnost reagiranja bazirano na portovima, protokolima i IP adresama. Ranih 2000-ih su se na internetu pojavile nove prijetnje kao SQL injekcije i XSS napadi. Navedene vrste napada su postale sve popularnije, napadi bi prošli kroz vatrozid te se zato počeo primjenjivati IDS. [2]

Dok je IDS postajao sve popularniji tih godina, malo koja organizacija je imala IPS. Zbog bojaznosti da bi IPS mogao blokirati bezopasne poruke ili podatke od kupaca ili korisnika. IDS za razliku od IPS-a nije blokirao maliciozne podatke, nego kada je otkrio nešto što je on smatrao malicioznim, slao je upozorenje administratoru organizacije kako bi administrator mogao pregledati zapise paketa i odlučiti ako je paket maliciozan ili ne. [1][40]

Tijekom ovog vremena potpisi su bili pisani da detektiraju načine eksploatacije, a ne slabosti. Za svaku slabost moglo je biti stotinjak različitih načina za iskorištavanje, zato su dobavljači pisali 100 ili više eksploatacijskih potpisa. Kad je jedan od tih poznatih potpisa prošao na mrežu, IDS bi poslao upozorenje administratoru. U to vrijeme dobavljači su se hvalili s brojem potpisa koje su imali u bazi podataka misleći da s time imaju bolju ponudu od ostalih dobavljača. [1]

Kada je krajem 2005. godine većina organizacija počela prihvaćati IPS, više dobavljača je počelo pružati podršku za njega. Također su se prestali hvaliti s količinom potpisa koje su imali u svojoj bazi podataka. Budući da je IPS ugrađen, klijenti su se bojali da će velika količina potpisa u bazi podataka znatno usporiti mrežu jer bi svaka veza morala biti provjerena na potpis eksploatacije. Zato su dobavljači počeli razvijati samo jedan potpis koji bi riješio svaku ranjivost bez obzira koliko exploita je s tom ranjivosti bilo povezano. [1][2]

Danas dobavljači i dalje odabiru one najvažnije i najrelevantnije potpise koji se nose s trenutnim prijetnjama kao i na neke starije prijetnje koje bi napadači mogli i dalje koristiti. Sustavi za detektiranje i prevenciju upada se konstantno mijenjaju te će se i dalje mijenjati kako napadači budu mijenjali svoje taktike i tehnike za ulaske u mreže.

Sigurnosne tvrtke koje su imale sustave za detekciju i prevenciju upada su pojačali sustave s 1 i 2 Gbps na 5 Gbps i s time su dobili mogućnost nadzora segmentiranih mreža te web farmi. Povećavajući brzinu na 5 Gbps je omogućilo puno veći kapacitet upravljanja protokom podataka na uređaju te je omogućilo nadgledanje puno više mrežnih segmenata nego prije. Danas su te protočnosti još puno veće od 40 Gbps do 60 Gbps. [1]

Sve se više i više klijenata počelo prebacivati na IPS. Do tada je tehnologija za prevenciju upada bila usklađenija nego prije. Tehnologija je bila bolja u odlučivanju što je promet koji nije štetan te samim time nije blokirala bezazlene podatke. Stoga su ljudi počeli koristiti sve više IPS način rada.

2011. godine se pojavila nova prijetnja na tržištu „Advanced Persistent Threat“ (APT). To su bili dokumenti koji su u sebi sadržavali malware. Budući da sustavi za detekciju i prevenciju nisu bili konfigurirani da se nose s ovom prijetnjom, ubačen je mod pješčanika ili emulacija u sustav. Pješčanik (Sandbox) je omogućavao da se malware nađe prvi dan na način da su se dokumenti i datoteke automatski otvarale u pješčaniku. Kada bi se otkrilo nešto maliciozno, administratoru bi bilo upućeno upozorenje. Ti pješčanici koristili su kontrolne sume. Svaka datoteka ima svoju jedinstvenu sumu te sume su nizovi znakova izrađeni od brojeva i slova koji se mogu koristiti za provjeru točnosti te integriteta datoteka i tekstnih poruka. Danas se ta tehnologija koristi u vatrozidovima nove generacije. [1]

3.2. Podjela Sustava

U ovome dijelu opisane su osnovne podjela za detekciju i prevenciju upada.

3.2.1. Podjela sustava za detekciju upada

Sustavi za detekciju upada se dijele na osnovu toga gdje je napad detektiran. Tako imamo sustave bazirane na hostu (engl. *Host-based intrusion detection Systems*) i sustave bazirane na mreži (engl. *Network-based intrusion detection Systems*).

3.2.1.1. Sustav za detekciju upada baziran na hostu (HIDS)

Sustav za detekciju upada baziran na hostu (HIDS) je sustav koji nadzire i analizira unutarnje dijelove računalnog sustava. Nadzire i mrežne pakete na mrežnim sučeljima unutar toga računalnog sustava. Sustav za detekciju upada baziran na hostu nadzire sve ili samo dijelove dinamičkog ponašanja i stanja računalnog sustava, sve zavisno o tome kako je postavljen i konfiguriran.

Osim dinamičkog pregleda mrežnih paketa koji su usmjereni na određeni host, HIDS može otkriti i vidjeti koji program pristupa kojem resurs. HIDS može otkriti neregularnosti ili nelogičnosti unutar tih programa, kao na primjer da program koji je zadužen za slike počne mijenjati baze podataka sa sistemskim lozinkama. Također, sustav za detekciju upada baziran na hostu može i promatrati stanje sustava. Također promatra podatke koji su spremljeni na tom sustavu koji su spremljeni u RAM, u sustavu datoteka ili u log datotekama. Sustav može pregledavati i provjeravati ako se sadržaj prikazuje onako kako bi trebalo po parametrima to jest, da nikakav uljez nije utjecao na sadržaj. Sustav za detekciju upada na bazi hosta se koristi bazom podataka objekata koje bi trebao nadzirati. HIDS također provjerava da odgovarajući dijelovi memorije nisu mijenjani kao na primjer strukture vtable u Windowsima. [4] Vtable je virtualna tablica funkcija koja se koristi za pozivanje funkcija na dinamičan način. Vtable se koristi za virtualne funkcije te je statična. Kada napadač mijenja virtualnu tablicu podataka on dodjeljuje memoriju unutar tablice svojim podacima te tako dobije sposobnost kontroliranja toka programa/podataka.[41]

Za svaki objekt HIDS će naći attribute kao što su: dozvola, veličina i datum. Nakon toga će stvoriti neku kontrolnu sumu za sadržaj. Kontrolne sume su obično hash, MD5, SHA1. Suma

se pohranjuje u bazu podataka koja bi trebala biti sigurna te se kasnije uspoređuje kako bi se potvrdilo da podaci nisu mijenjani. Ako su podaci mijenjani onda sustav obavještava moderatora o neregularnosti i mogućem malicioznom napadu na sustav. HIDS je prva vrsta sustava za otkrivanje upada koja je dizajnirana s ciljem promatranja glavnog računala bez puno vanjskih interakcija. [4]

3.2.1.2. Sustav za detekciju upada bazirani na mreži(NIDS)

Sustavi za detekciju upada bazirani na mreži djeluju na drugačiji način od onih baziranih na hostu. Ovaj sustav osmišljen je kako bi pomogao organizacijama da nadgledaju svoja cloud, lokalna i hibridna okruženja na sumnjive događaje. Sustav skenira mrežne podatke na razini usmjerivača ili na razini poslužitelja. Sustav revidira podatke o paketima i sve što se čini sumnjivo prijavljuje u sebi posebnu datoteku log-a u kojoj se daju proširene informacije. Koristeći podatke o paketima tj. podatke o sumnjivim paketima, sustav skenira vlastitu bazu podataka potpisa mrežnih napada te dodjeljuje razinu ozbiljnosti za svaki zasebni paket. Ako je razina ozbiljnosti dovoljno visoka, moderatorima ili sigurnosnom timu se šalje upozorenje u obliku e-maila ili na mobitel kako bi oni mogli istražiti razlog neregularnosti.

S rastom i razvojem Interneta te povećanjem prometa preko interneta, rasla je i popularnost sustava za detekciju upada baziranih na mrežama. Sustav može skenirati velike količine mrežnih aktivnosti i označavati sumnjive prijenose. Sustav također ima mogućnost usporedbe potpisa za slične pakete u svrhu povezivanja ili kako bi maknuli štetne pakete s potpisom kojeg imamo u našoj bazi podataka. Najbolji način je skenirati sav dolazni i odlazni promet, ali to bi moglo dovesti do problema uskog grla koje bi smanjilo ukupnu brzinu mreže.

Sustav za detekciju upada baziran na mrežama se prema interaktivnosti sustava mogu podijeliti u dvije podvrste:

- Mrežni ili online
- Izvanmrežni ili tap način rada

Mrežni ili On-line sustav za detekciju upada se bavi mrežom i podacima u stvarnom vremenu. Sustav analizira podatke i primjenjuje pravila i potpise kako bi se donijela odluka je li nešto napad ili ne.

Izvanmrežni ili Offline sustav za detekciju upada se bavi pohranjenim podacima i prolazi kroz podatke obavljajući procese te analizira ako je nešto napad ili ne. Ovaj sustav zapravo nadzire kopiju mrežnog prometa što znači da stvarni mrežni promet ne prolazi kroz sustav.

NIDS se također mogu spajati s ostalim tehnologijama kako bi poboljšali otkrivanje i predviđanje. Zbog svojevrstne nesigurnosti TCP/IP protokola u novije vrijeme, potrebno je razviti razne skenere i alate za reviziju i otkrivanje mreže. U svrhu sprječavanja malicioznih mrežnih aktivnosti kao što su spoofanje IP-a, DOS napadi, cache poisoning, korupcija DNS-a. [5]

3.2.1.3. Podjela sustava prema načinu otkrivanja napada

Sustavi za detekciju upada se prema načinu otkrivanja napada dijele na ove vrste:

- Detekcija potpisom
- Detekcija na temelju anomalije

Kod detekcije potpisom se uspoređuju događaji koje promatramo s potpisima. Ova vrsta detekcije je najjednostavnija i najkorištenija vrsta otkrivanja jer otkriva napade traženjem uzoraka kao sljedova bajtova u mrežnom prometu ili poznate sljedove malicioznih uputa koje koristi maliciozni softver. Odnosno, ova vrsta detekcije prepoznaje loše obrasce koje koriste maliciozni programi ili malware. Ovaj sustav se koristi kod otkrivanja već poznatih napada. Budući da se koristi obrascima i već poznatim podacima, jako teško otkriva nove napade. Pogotovo one nove napade za koje još ne postoji obrazac.

Kod detekcije anomalijom se uspoređuje mrežni promet s uobičajenim modelom dobrog mrežnog prometa. Ovi sustavi su prvotno uvedeni za otkrivanje nepoznatih napada što detekcija potpisom ne može. Maliciozni softver se u novije vrijeme brže razvijao kao i nove vrste napada što je dovelo do potrebe detektiranja tih napada. Detekcija anomalijom je zapravo vrsta strojnog učenja gdje se stvara model pouzdane aktivnost ili dobrog mrežnog prometa. Zatim se svako novo ponašanje uspoređuje s tim modelom. Ova metoda ima bolja svojstva otkrivanja u usporedbi sa sustavima za detekciju upada koje detektiraju putem potpisa. Ovaj sustav uvelike omogućuje otkrivanje prethodno nepoznatih napada. No, navedeni sustav ima

problem, a to je da može patiti od lažno pozitivnih rezultata. Prethodno nepoznate aktivnosti koje su klasificirane kao legitimne se također mogu klasificirati malicioznima

Postoji još jedan pristup detekcije gdje se koriste općeprihvaćene definicije za svako stanje protokola. Promet ili protokol se analiziraju pomoću tih definicija, traže se odstupanja od postavljenih vrijednosti. Takav način se naziva detekcija temeljena na reputaciji.

3.2.2. Sustav za prevenciju upada

Sustavi za prevenciju upada (IPS) su varijacija sustava za detekciju upada (IDS) i uz sposobnost detekcije upada imaju i sposobnost odgovora na otkrivene upade ili napade. Takvi sustavi se još nazivaju i sustavi za otkrivanje i sprječavanje upada (IDPS). Sustavi za otkrivanje i sprječavanje upada najprije su usmjereni na otkrivanje mogućih incidenata, a zatim za zapisivanje tih podataka u log te prijava malicioznih radnji. Uz te osnovne radnje sustav za prevenciju upada se još koristi za identificiranje problema u sigurnosti unutar sustava i za dokumentiranje postojećih ugroza. Sustav zapisuje informacije koje su povezane s promatranim događajima te obavještava administratore o važnim događajima ili ugrozama i pravi izvješća.

Uz te radnje IDPS ima i sposobnost sprječavanja prijetnji. Za to koriste par tehnika odgovora kao što su :

- Zaustavljanje napada
- Promjena sigurnosnog okruženja
- Promjena sadržaja napada[35]

Sustavi za sprječavanje upada su proširenje sustava za otkrivanje upada. Oboje nadziru mrežu ili aktivnost sustava za maliciozne aktivnosti. Za razliku od sustava za detekciju upada, sustavi za sprječavanje upada su inline i imaju sposobnost aktivno sprječavati ili blokirati napade. Sustav za sprječavanje upada može slati upozorenja administratorima, može ispustiti pakete koji su se pokazali malicioznim, mogu resetirati vezu ili mogu potpuno blokirati mrežni promet s neke IP adrese koja se pokazala malicioznom.

3.2.2.1. Podjela sustava za prevenciju upada

Sustavi za prevenciju upada se dijele na 4 različite vrste a to su:

- Mrežni sustavi za prevenciju upada (engl. *Network-based intrusion protection system*)
- Bežični sustavi za prevenciju upada (engl. *Wireless intrusion protection system*)
- Sustavi za prevenciju upada bazirani na ponašanju mreže (engl. *Network behavior analysis*)
- Sustavi za prevenciju upada bazirani na hostu (engl. *Host-based intrusion protection system*) [2][32]

Mrežni sustav za prevenciju upada se koristi za nadgledanje mreže, zaštitu integriteta te zaštitu dostupnosti mreže. Glavne funkcije mrežnog sustava su zaštita mreže od prijetnji kao što su DOS napadi. Mrežni sustav analizira mrežu za maliciozne aktivnosti ili sumnjivi promet. Kada se mrežni sustav instalira, on stvara zonu sigurnosti te zatvara sve maliciozne stvari poput trojanaca, crvi i virusa u izolaciju gdje ne mogu više biti u kontaktu sa sustavom.

Bežični sustav za prevenciju upada se koristi za sprečavanje neovlaštenog upada lokalnim mrežama te drugim povezanim bežičnim uređajima. Obično se ovi sustavi stavljaju preko infrastrukture bežičnog LAN-a. Sustav nadzire bežičnu mrežu, nadzire aktivnosti unutar te bežične mreže. Sustav analizira aktivnosti pa nakon provedene analize aktivnosti detektira one aktivnosti koje su sumnjive te ih sprječava.

Sustavi za prevenciju upada bazirani na ponašanju mreže analiziraju podatke o ponašanju mreže te nam govore kako funkcioniraju naši sigurnosni protokoli i sustavi. Konstantnim promatranjem mreže se osigurava sigurnost jer nas sustav informira kakvo je ponašanje na mreži normalno. Kada ima sliku toga kakvo je normalno ponašanje na mreži, sustav može upozoriti administratora na pokušaje upada budući da se svaka anomalija tj. neregularnost u mreži nadzire i prijavljuje.

Sustav za prevenciju upada baziran na hostu je obično instalirani paket koji nadgleda promet i sustav na jednom zasebnom računalu ili serveru. Sustav štiti od poznatih i nepoznatih malicioznih napada tako što redovito provjerava karakteristike poslužitelja i događaje koji se događaju unutar toga poslužitelja. Uspoređuje stanje prije događaja sa stanjem nakon događaja te ako detektira anomalije u stanju, obavještava administratora ili blokira aktivnost. Ova vrsta sustava se može postaviti na razne vrste strojeva kao što su serveri, radna računala, kućna računala. Ovaj sustav jednako dobro se nosi sa šifriranim i nešifriranim prometom, ali ne može nadgledati događaje na mreži.

4. Pregled besplatnih i komercijalnih sustava

U ovome dijelu rada su opisani neki od najviše korištenih sustava za detekciju/prevenciju upada. SNORT će biti objašnjen kasnije jer ćemo ga koristiti za praktični dio.

4.1. Besplatni sustavi za detekciju/prevenciju upada

U ovome dijelu rada će biti opisani neki od najkorištenijih besplatnih sustava danas. Prvo je opisan program Suricata, a zatim program OSSEC.

4.1.1. SURICATA

Suricata je besplatan i slobodno dostupan (engl. *Open source*) sustav za detektiranje mrežnih prijetnji i napada. Suricata je i sustav za detektiranje upada u stvarnom vremenu i sustav za prevenciju upada. Možemo ga klasificirati kao sustav za detektiranje i prevenciju upada. Uz to što se bavi upadima, Suricata još nadgleda sigurnost mreže te offline pregledava i analizira PCAP vrstu datoteka. PCAP datoteka se sastoji od paketnih podataka neke mreže te se koristi kod analiziranja karakteristika neke mreže. Također može promet spremi u obliku pcap zapisa. Ima potporu za dekodiranje paketa od raznih protokola od kojih su neki: IPv4, IPv6, TCP, UDP, SCTP, ICMPv4, ICMPv6, GRE.

Suricata koristi ključne riječi za protokole i koristi se pravilima. Također, Suricata ima sposobnost uspoređivanja i spajanja datoteka prema veličini, imenu i ekstenziji. Suricata ima sposobnost osvježavanja pravila bez da se mora ponovno pokrenuti program. Koristi kontrolne sume koje imaju svoj format (MD5, SHA1/SHA256). Suricata ima jaku potporu za LUA skriptiranje koje se koriste za osobno detektiranje složenijih prijetnji. Suricata koristi standardni input format YAML-a, a za outpute koristi JSON (engl. *JavaScript Object Notation*) integraciju kako bi se lako povezala s bazama podataka. [9]

Jedna od glavnih razlika Suricata od ostalih sustava za detekciju/prevenciju je njena višedretvenost (engl. *Multithreading*) što je novitet na području sigurnosti. Prijašnji sustavi za detekciju/prevenciju upada nisu imali sposobnost višedretvenosti te su uvijek koristili jednu jezgru. Nije bilo bitno koliko jezgri su imali jer je sustav uvijek koristio samo jednu jezgru. [36]

Omogućuje brzinu i stabilnost rada programa pogotovo s time što ima ugrađeno ubrzavanje hardwarea pomoću npr. grafičke kartice.

Suricata projekt i kod je pod vlasništvom OISF (engl. *Open information security foundation*) koja je neprofitna organizacija odlučna da osigura razvoj i uspjeh ovog slobodno dostupnog projekta. [9]

Suricata ima potporu za većinu operacijskih sustava današnjice:

- Linux
- Mac OS/mac OS X
- Windows
- Operacijski sustavi bazirani na UNIX-u

4.1.2. OSSEC

OSSEC je besplatan i slobodno dostupan (engl. *Open source*) sustav za detektiranje upada. OSSEC je sustav za detektiranje upada baziran na hostu. Ime OSSEC je skraćeno od (engl. *Open source security*)

Glavne funkcije OSSEC-a su:

- Otkrivanje upada temeljem zapisa (engl. log) - Sustav aktivno nadgleda i analizira podatke iz više zapisa u stvarnom vremenu te uspoređuje podatke koje trenutno imamo s podacima u zapisima. Nadgledanjem zapisa na računalu program zna kada je pod sustav pod napadom, kada je nešto instalirano na sustav ili kad netko promijeni pravilo u našem vatrozidu (engl. *Firewall*).
- Provjera integriteta podataka – Svaki napad na našu mrežu ili računalo uvijek ostavi trag tako da napad mijenja naš sustav na neki svoj način. Pomoću provjere integriteta podataka OSSEC uspoređuje normalne parametre podataka te parametre sada kako bi detektirao promjene. Kada se promjene detektiraju, one se javljaju administratoru na uvid jer neke od tih promjena mogu biti greške zaposlenika, mogu biti i napad.
- Detekcija rootkita - Rootkit obično dođe na sustav preko root ili administratorskog računa te kada dođe na sustav mijenja neke od osnovnih datoteka s izmijenjenim

verzijama i time uspostavlja backdoor ulaz. Uspostavljanjem backdoora omogućava pristup čak i ako se promijeni root ili admin lozinka. Nakon toga se brišu svi podaci o izmjeni sustava kao što su izmjena imena datoteke te promjena privremene predmemorije. OSSEC ima sposobnost detekcija rootkita. Detektira na način sličan pravilima koje javljaju sustavu ako je došlo do nekog slučaja promjene prava koji je dio rootkit-a jer trebaju dobiti prava kako bi uspostavili backdoor ulaz. Kada se jedno od upozorenja pomoću pravila detektira, program javlja administratoru o mogućem rootkitu.

- Aktivni odgovor - Ova funkcija OSSEC-a dozvoljava sustavu da kada se pokrenu određena upozorenja, obično opasnije prirode, da sustav sam reagira i blokira širenje i korištenje na sustavu. Ovaj proces je automatiziran te spašava sustav od napada prije nego administrator napravi neku radnju.

OSSEC dozvoljava i nadgledanje s agentom i nadgledanje bez agenta što znači da radi i s vatrozidovima i routerima. Ova mogućnost dozvoljava sustavima koji imaju zabranu instaliranja programa da se svejedno osiguraju sa OSSEC. [10][11]

OSSEC ima potporu za većinu operacijskih sustava današnjice:

- Linux
- Mac OS/mac OS X
- Windows
- Operacijski sustavi bazirani na UNIX-u

4.2. Komercijalni sustavi za detekciju/prevenciju upada

U ovome dijelu su navedena dva komercijalna sustava za detekciju/prevenciju upada kao i njihove funkcije

4.2.1. Cisco Stealthwatch

Cisco Stealthwatch je komercijalni sustav za detekciju/prevenciju upada. Najnovija verzija ovoga sustava je verzija 7. 3 koja je izašla 10. 08. 2020. Ovaj sustav u vlasništvu je Cisco Systems korporacije.

Glavne odlike programa Cisco Stealthwatch su:

- Neprestano nadgledanje svih uređaja, aplikacija i korisnika kroz cijelu mrežu te sprema podatke o mreži, aplikacijama, uređajima i korisnicima na dnevnoj, tjednoj i mjesečnoj bazi. Svi se podaci spremaju pod istim tipom te se tako lakše interpretiraju.
- Sustav ima sposobnost detekcije napada preko cijele mreže sustava te ima sposobnost detekcije na kojoj lokaciji se napad dogodio i kada. Odnosno, ima mogućnost reći koji korisnik na mreži je napadnut na kojem uređaju, u koje vrijeme te na kojoj aplikaciji. Cisco Stealthwatch ima sposobnost analiziranja enkriptiranih podataka kako bi se utvrdilo jesu li oni prijetnja ili ne bez dekrpcije tih podataka.
- Sustav može odrediti kada se sustav ili korisnik ponašaju drugačije nego normalno npr. šalju previše podataka ili e-mailova. Može detektirati koji resurs na našoj mreži je pod napadom te u istu ruku može identificirati napadača. Sustav upozorava ako je naš sustav u komunikaciji s nepoželjnim osobama.
- Koristi telemetrijske podatke s mrežne infrastrukture i time daje naprednu detekciju prijetnji tako što ima sustav ozbiljnosti i prema tome prioritizira upozorenja od najkritičnijih do lažno pozitivnih.
- Daje ubranu reakciju na prijetnje jer sprema telemetrijske podatke na duže vremena te se tako mogu lakše koristiti za analizu prijetnji ili forenzičku analizu nakon napada.
- Ovaj sustav je poprilično skup i cijena ovisi o vrsti i količini usluge koje koristimo.
- Ima mogućnost testiranja prije kupnje (engl. Free trial) u trajanju od 14 dana.[13]

4.2.2. SolarWinds Security Event Manager

SolarWinds Security Event Manager (SEM) je prvotno je upravitelj zapisnika to znači da je sustav za detekciju upada baziran na hostu, no isto tako može upravljati podacima sakupljenima s programom Snort što ga u isto vrijeme čini i sustavom za prevenciju baziranim na mreži.

Glavne odlike ovog sustava su:

- Analiziranje zapisnika
- Procesiranje podataka u stvarnom vremenu te reakcija na prijetnje u stvarnom vremenu

Uz mogućnosti detekcije sustav ima i sposobnost prevencije upada jer ima ugrađene automatizirane akcije koje se obavljaju kada se detektira upada. Sustav ima preko oko 1000 pravila što mu omogućava da prepozna sumnjive aktivnosti te da odmah implementira radnje kako bi se odgovorilo na taj upad. Takav način reakcije naziva se aktivni odgovor. [16]

Postoji dosta vrsta aktivnih odgovora a neki od njih su:

- Blokiranje IP adresa napadača
- Blokiranje korisnika
- Izbacivanje korisnika iz sustava
- Izoliranje USB-a kako bi se zaštitio sustav i spriječilo širenje
- Ubijanje procesa
- Gašenje sustava ili ponovno pokretanje
- Javljanje upozorenja administratorima preko E-maila ili preko poruka na ekranu
- Gašenje servisa [12]

SEM se trenutno jedino može instalirati na serveru Windows, ali on može zapisati podatke koji su generirani i na računalima koja imaju operacijske sustave Linux, Unix, Mac.

SEM je komercijalni sustav za detekciju/prevenciju upada te je početna cijena oko 2000 eura zavisno o paketu, no postoji mogućnost besplatnog testiranja u trajanju od 30 dana na njihovoj stranici.

4.3. Usporedba slobodno dostupnih sustava

U ovome dijelu rada uspoređena su dva najkorištenija besplatna sustava za detekciju upada trenutno Suricata i Snort.

4.3.1. Pravila

Sustav za detekciju upada je dobar onoliko koliko su dobra njegova pravila za praćenje prometa ili mreža[18]. Snort je program koji ima najveću riznicu pravila i ima potporu internetske zajednice kroz godine te se samim time ta riznica pravila povećavala. Pravila napisana od strane zajednice se sva nalaze u jednoj datoteci koja je nazvana `community.rules` te se besplatno skinu sa službene stranice od Snorta[38]. Snort još ima i pravila za download na stranici za registrirane korisnike. Postoje pravila razvijena od strane komercijalnih tvrtki koja se mogu kupiti na mjesečnoj ili godišnjoj. Snort također daje korisnicima mogućnost pisanja osobnih pravila koja se spremaju u `local.rules`.

Suricata isto može koristiti većinu komercijalnih pravila napisanih za Snort, no ne može koristiti pravila napisana za korisnike Snorta radi drugačije sintakse. Suricata ima i svoja pravila. Suricata pravila se prvo daju pretplatnicima te se tek nakon toga daju ostalim korisnicima u roku od 60 dana što naš sustav može ostaviti ranjivim u tom vremenskom roku. Jedna od najbitnijih razlika u Suricatinim i Snortovim pravilima je to što Suricata pravila imaju automatsku detekciju protokola. To radi na način da detekciju vrši bez potrebe za predefiniranim portom. Dok Snort detekciju vrši putem pretprocesora koji, kako bi se primijenili na promet, moraju ići preko postavljenog porta.[19]

4.3.2. Izvođenje procesa

Snort je poprilično star program dok je Suricata noviji te samim time Suricata ima nešto funkcija koje su potrebne za izvođenje na kompleksnijoj strukturi mreža i sustava današnjice koje nisu bile potrebne u vrijeme kad je Snort izašao. Jedna od najbitnijih razlika od onda do danas je povećanje količine podataka u mrežnom prometu te promjena na ipv6. Što znači da se promet povećao tako se povećala i potreba za procesiranjem tih podataka te brže procesiranje

podataka. Snort 2.x ne podržava višedretvenost te samim time se procesiranje obavlja sporije jer se radi na jednoj jezgri, dok Suricata kao noviji program ima od početka ugrađenu višedretvenost te se s time može pohvaliti i bržim procesiranjem podataka jer se procesiranje podataka odvija na više jezgri istovremeno te se jedan proces također može procesirati na više jezgri istovremeno.

4.3.3.Primjena

Uzimajući u obzir Snortovu limitiranost dretvi i procesiranja podataka, Snort je bolje korišten u manjim mrežama gdje procesiranje ogromne količine podataka nije potrebno. Također Snort je manje intenzivan na CPU (engl. *Central processing unit*) i na RAM te je samim time bolje postavljati ga na manju mrežu koja još ima stariju infrastrukturu te starija računala.

Suricata je u većoj mjeri korištena u većim mrežama zbog svoje višedretvenosti, no Suricata je u isto vrijeme intenzivnija na CPU i RAME te samim time zahtjeva novija računala i infrastrukturu. Suricata sposobnost vađenja podataka i spremanje tih podataka u neki izolirani spremnik kako bi se podaci mogli dalje analizirati joj pomaže u većim mrežama.

4.3.4.Snort 3.0

Treba napomenuti da Snort trenutno ima i beta verziju Snort 3.0 koja dodaje nove funkcionalnosti na program Snort i uklanja neke zastarjele mehanizme. Jedna od tih novih funkcionalnosti je gore spomenuta višedretvenost. Jednostavnije konfiguriranje programa je također jedna od novosti Snort 3.0 je automatsko detektiranje servisa bez korištenja portova. Ova verzija Snorta ima i svoju vrstu pravila koja se s instalacijom za Snort 3.0 mogu skinuti na službenoj stranici od Snorta.

5. Praktični dio rada

U ovom dijelu rada prvo je opisan program koji će biti korišten, zatim je prikazana konfiguracija programa na Windowsima i na Linux- te na kraju testiranje programa nekim najčešćim vrstama napada s Kali Linuxa.

5.1. Snort

Snort je jedan od najraširenijih slobodno dostupnih sustava za detekciju upada. Također, Snort spada u skupinu sustava za detekciju upada baziranih na mreži. Snort je kreirao Martin Roesch 1998. godine. Martin Roesch je ustanovio i bio glavni tehnološki direktor tvrtke Sourcefire. Sourcefire je kupljen od strane Cisco Systemsa 2013. godine koji su sada zaslužni za daljnji razvoj ovog projekta otvorenog koda. [33]

Godinu nakon kreacije u 1999-toj godini je izašla verzija 1.0 i od tada počinje pravi život Snorta. Od 1999. do 2020. su izašle razne verzije. Svaka nova verzija donosi svoja poboljšanja. Kako su se vremena i zahtjevi u mrežnom prometu mijenjali, tako se mijenjao i Snort. Najnovija stabilna verzija u vrijeme pisanja ovog rada je Snort 2.9.16.1 i ta se verzija koristi u ovom radu.

Kada preuzimamo Snort moramo preuzeti i pravila koja su dostupna na istom mjestu gdje i program. Pravila su podijeljena u tri dijela:

- Pravila zajednice (engl. *Community rules*) - sva se pravila nalaze u jednoj datoteci koja se naziva *community.rules*. U njoj su sva osnova pravila unutar jedne datoteke, no ta pravila nisu toliko održavana kao ostala dva tipa.
- Pravila za prijavljene korisnike (engl. *Registered rules*) su podijeljena po protokolima i po vrstama pravila u zasebne datoteke svaka s nastavkom. *rules*. Kako bi mogli skinuti ova pravila potrebno je samo registrirati se na stranicu Snort. U ova pravila dolaze i pravila iz pretplate nakon 30 dana.
- Pretplata (engl. *Subscription*) - ovdje dolaze najnovija razvijena pravila za najnovije prijetnje, nakon 30 dana ova pravila idu u pravila za prijavljene korisnike. Pretplata je godišnje prirode te se može kupiti pravila za jednog osobnog korisnika kao i pravila za jedno poduzeće. Cijena pravila za jednog osobnog korisnika je 30\$ godišnje dok je cijena za jedno poduzeće 399\$ po senzoru.

Snort može analizirati protokole, pretraživati sadržaj te upariti napade. Snort obavlja analizu mrežnog prometa u stvarnom vremenu.

Snort ima tri osnovna načina rada :

- Packet sniffer mode
- Packet logger mode
- Network intrusion detection

5.1.1. Packet sniffer mode

Ovo je najjednostavnija funkcionalnost Snorta, njuši pakete podataka koji putuju kroz mrežu. Administratori koriste ovaj alat kako bi pratili promet na mreži na razini paketa te tako osigurali dobro ponašanje sustava i zdravlje.

Ovaj mod se najjednostavnije pokreće s naredbom Snort i onda dodamo slovo parametra zavisno koju funkcionalnost njušenja želimo:

- Snort -v :samo ispisuje IP/TCP/UDP/ICMP zaglavlja
- Snort -vd uz prethodna zaglavlja još ispisuje podatke aplikacija u prijelazu
- Snort -vde ispisuje uz prethodna dva moda još i zaglavlja paketa

Vrijedno spomena je da se prijašnje naredbe mogu i napisati odvojeno, tako Snort -vde se može napisati i kao Snort -d -e -v.

Primjer pokretanja Snorta u Sniffer modu je prikazan na slici 1. gdje vidimo kako Snort ispisuje zaglavlje protokola te podatke aplikacija i zaglavlja paketa koje nanjuši.

```

09/06-15:38:11.658341 18:5F:0F:DA:7D:4D -> 64:6F:FA:26:F1:75 type:0x800 len:0x233
192.168.1.4:53057 -> 172.217.19.99:443 UDP TTL:128 TOS:0x0 ID:7932 IpLen:20 DgmLen:549 DF
Len: 521
4F 7E B5 0C 64 C3 19 42 B9 28 69 35 8E 9B C1 D5 0~.d..B.(i5...
2D EC 9B 71 76 6F 32 CC 92 2D D9 C4 F1 8E 33 29 -.qvo2...3)
7E A2 C6 FD 8E A9 59 00 E2 70 CC 62 C7 6C 5F 26 ~....Y..p.b.l&
FE 47 FA 56 29 D4 0F 18 B2 98 15 E3 8C 6F 6A 06 .G.V).....oj.
7A FC 22 0F DD 2F FR 3F 09 1A 06 3F 9F 8C R6 5F 7 " ? > _
BB EF 67 9A 9E B9 6C B1 05 E0 06 B3 47 53 57 D4 ;.g...l....GSW.
C2 AE 94 33 5C 60 AC 70 1A 0D CA 4B 13 C9 BA 88 ...3\`p...K....
5A 57 D7 B4 90 BD 10 C5 BB 92 24 2B CD BA CA 2B ZW.....$!...t
9C 87 01 6E 12 B8 24 55 75 E2 3D 80 20 0A 2C 6E ...n..$Uu,=. ,n
8B 3D 0D 86 48 23 1D E6 75 F9 89 AF 5C 8E FD 07 .=.H#..u...\.
5C 11 D9 F1 2D BF FF 95 AE 9F 5F 21 62 B9 39 31 l...-.....lb.91
50 2F 12 93 80 62 55 4C F9 FD C4 64 93 A2 9A 44 ~...hUJ...d...D
C5 3E 4A 90 E1 30 40 82 E1 24 8B 24 8D 16 6D 9E .>J..0@..$.$.m.
12 9B 88 64 68 55 1D 1A 2C D3 83 CE 9C 95 32 1E ...dhU...2.
0A 89 5E 99 5B 7A C9 24 F9 35 9C 61 AC 4D 92 F9 ...^[z,$.5.a.M..
D1 E8 81 82 68 B9 DA 47 73 77 E8 C5 04 E4 F0 FA ...h..Gsw.....
8C C5 D4 FR DF BD DR 2D 06 86 FF 18 74 29 AB FR .....t)....
43 3D 19 7A 29 1A 47 95 51 C1 26 C4 02 FA AC 41 C=.z)..G.Q.&...A
84 50 17 FR 3D 05 87 C9 0C GA D1 5F 30 56 BF 49 4P...=...j.^0V.T
C4 FA 8C D7 03 C6 E4 14 CE 56 38 26 3A 19 FB 43 .....V8&:...C
A9 DE 3D 5C 42 B2 14 85 C1 28 EB 88 0D 08 51 48 ..=\B....(....QH
84 ED 1B 9D 61 27 ED 9D 6D 3F A7 20 BD 7F 57 E5 ...a'..m?...W.
E4 BD A0 BD 8B 81 69 EE 1E AD 81 C9 D9 E0 2E 5D .....i.....]
8B FE D8 87 59 46 2C 0A D0 03 25 10 85 F9 0B D7 ...YF...%.....
26 3B 15 A6 23 2C E6 EA 2B F8 AF F6 CC 82 73 2A &;..#,...+....S*
88 9D 7B 70 04 6B 88 10 B5 6F 76 7D 84 FB 02 D3 X.{ ..k...n&}....
85 BF BD 13 50 97 6E 40 06 A2 BE A3 A7 24 08 6F ....P.n@.....$.o

```

Slika 1:Packet sniffer mode

5.1.2.Packet logger mode

Packet logger je u principu dosta sličan načinu rada Packet sniffera, najveća razlika kod logger moda je ta što se paketi koje Snort uhvati uhvate ne prikazuju na ekranu nego se zapisuju unutar zapisnika.

Kao i packet sniffer mode i on ima svoje parametre tako logger mode ima sljedeće naredbe:

- -l pokreće logger mode te se kraj -l mora naći i mjesto gdje se zapisnik nalazi najosnovniji način upisivanja ovoga bi izgledao ovako :. /Snort -l. /log što govori da pokrećemo logger mode i spremamo podatke u log.
- -h pomoću ovog parametra određujemo koja mreža je kućna mreža pa bi uz parametar -h obično išla neka IP adresa kao npr :. /Snort -l. /log -h 192. 168. 1. 1 ili se IP adresa može pisati u tipu 192. 168. 1. 0/24 što bi značilo da se pregledava cijela mreža. Zapisuju se samo podaci koji kao odredište imaju adresu ili mrežu zapisanu nakon parametra -h.
- -b parametar b u logger modu govori Snortu da zapisuje podatke u log u binarnom obliku. Ovaj parametar se obično koristi kada imamo puno podataka ili kada želimo kasnije grupirati i analizirati te podatke. Primjer naredbe bi bio isti kao i u prošlom primjeru samo bi se na kraj još dodao parametar -b.

5.1.3. Intrusion detection mode

Mod detekcije upada je najsloženiji mod i glavni razlog zašto se Snort uopće koristi pa će zato biti i osnova ovoga praktičnoga dijela rada. U ovom modu se paketi u prolazu dohvaćaju te se pomoću već spomenutih pravila pregledavaju. Nakon pregledavanja paketa Snort radi onu akciju koja mu je predodređena bazirano na spoznaji o vrsti podataka. Kako bi pokrenuli Snort u ovom modu uz slične parametre kao i u prva dva moda još moramo dodati i lokaciju konfiguracijske datoteke snort.conf u kojoj se nalaze svi podaci za konfiguraciji, a o kojoj će više biti riječi kod instalacije i implementacije sustava. Još se mora dodati i ime adaptera koji će prikazivati a dobije se pomoću ipconfig naredbe u command prompt-u za Windows korisnike. Za Linux Debian korisnike ta naredba je ip a ili pomoću ifconfig koja je zastarjela te. Na slici ispod možemo vidjeti kako se vidi ime adaptera s naredbom ip a u Debianu. Kao što se može vidjeti ime adaptera je eth0.

```
kali2@kali2:~$ sudo ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
efault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:50:97:a4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefix
route eth0
        valid_lft 547sec preferred_lft 547sec
    inet6 fe80::a00:27ff:fe50:97a4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
kali2@kali2:~$
```

Slika 2: ip a naredba u Linuxu

Kada znamo ime adaptera i dodajemo još parametar -A i uz njega dodajemo još opis parametra:

- **Fast** je osnovni prikaz
- **None** je bez obavijesti
- **Full** je potpuna obavijest uz zaglavlje paketa
- **Console** se prikazuje u konzoli

Kada znamo način na koji želimo prikazati podatke te znamo ime adaptera možemo pokrenuti način rada detekcije upada. S naredbom u naredbenom retku ili u terminalu kao na slici 3.

```
kali@kali:~$ sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf
Running in IDS mode

--= Initializing Snort =--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741
1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145
7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181
8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 3444
3:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 12
20 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:70
01 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 81
23 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 99
99 11371 34443:34444 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]
Detection:
```

Slika 3: Pokretanje Snorta u IDS modu

Kod pokretanja Snorta se također učitavaju pravila koja su definirana u rules datoteci, a putanje do njih se upisuju u konfiguracijskoj datoteci koju pokrećemo. U ovom slučaju uključena su većina pravila iz registriranih pravila te se može vidjeti da je učitano 12665 pravila. Također se može vidjeti kojeg tipa su pravila to jest koji protokol nadgledaju što se može vidjeti na slici 4.

```

+++++
Initializing rule chains ...
12665 Snort rules read
    12665 detection rules
    0 decoder rules
    0 preprocessor rules
12665 Option Chains linked into 460 Chain Headers
+++++

+-----[Rule Port Counts]-----+
--
|
|   src    tcp    udp    icmp    ip
|   dst    4167   26     0      0
|   any    7956   97     0      0
|   nc     410    7      6      0
|   s+d    5      0      2      0
|
+-----+
--

+-----[detection-filter-config]-----+
|
| memory-cap : 1048576 bytes
+-----[detection-filter-rules]-----+

```

Slika 4: Broj učitanih pravila

5.2. Pravila

Glavna osnovica sustava Snort su njegova pravila. Već smo spomenuli kako postoje tri vrste pravila koje možemo skinuti besplatne, za prijavljene korisnike i za pretplatnike. Pravila omogućuju analizu mrežnog prometa te analizu mrežnih paketa. Ona omogućavaju i pretragu IP adrese.

Pravilo broj jedan Snort pravila je to da pravila moraju unutar datoteke biti unesena u jednom redu. Znači svako pravilo mora biti u zasebnom redu. Svako pravilo se sastoji od 2 logička dijela, zaglavlje pravila i opcije pravila[25].

Zaglavlje pravila sadrži podatke o osnovnim podacima o paketu kao tko šalje paket, tko prima paket i na koji način se šalju te na koji port. Prvi podatak koji se nalazi u pravilu je akcija pravila. Ovaj dio pravila govori sustavu što da napravi s paketom koji je detektiran. Postoji pet osnovnih akcija a to su :

- **Alert**-Snort generira upozorenje kad se dogodi određena radnja i zatim bilježi paket.
- **Log**- samo bilježi paket

- **Pass**-ignorira paket
- **Activate**-Pravi upozorenje te onda aktivira dinamičko pravilo povezano s tim upozorenjem
- **Dynamic**-Povezani s activate naredbom

Iduća opcija po redu unutar pravila je definiranje mrežnog protokola. Tu imamo tri osnovne opcije protokola:

- UDP
- ICMP
- TCP

Nakon protokola definiramo IP adrese te smjer prometa. Prvo ide IP adresa te port nakon toga ide smjer koji može biti u jednom smjeru i označen je strjelicom ili može biti obostran i označen je s <>. Osnovna sintaksa jednog pravila bi izgledala kao na slici dolje gdje ćemo prikazati dva napisana pravila koja ćemo kasnije iskoristiti.

```
alert icmp any any → 192.168.56.2 any (msg:"ICMP test"; sid:10000001; rev:001;)
alert icmp any any → 192.168.56.2 any (msg: "NMAP ping sweep Scan";sid:10000002; rev: 001;)
```

Slika 5:Lokalna pravila

Kao što vidimo na slici 5. postoji i dio nakon zaglavlja te se on sastoji od poruke koje će sustav ispisati kada se to pravilo bude dogodilo te identifikator pravila koji mora biti najmanje broj 1000001 jer je prvih 1000000 brojeva pravila tj identifikatora pravila rezervirano za pravila zajednice i Sourcefire pravila. Pravilo na kraju ima i verziju pravila ili rev.

Uz msg koji možemo vidjeti u primjeru postoji još opcija koje mogu biti umjesto msg:

- **Flow** definira smjer prometa te pomaže da se pravila održavaju samo u jednom smjeru.
- **Reference** ili ref referencira izvore koji opisuju napad koji se događa.
- **Classtype** ili class određuje klasu pravila po već unaprijed određenim postavkama, postoji otprilike 40 već postavljenih vrsta pravila koji bliže određuju funkciju napada te njegovu učinkovitost.

5.3. Instalacija i konfiguracija Snorta

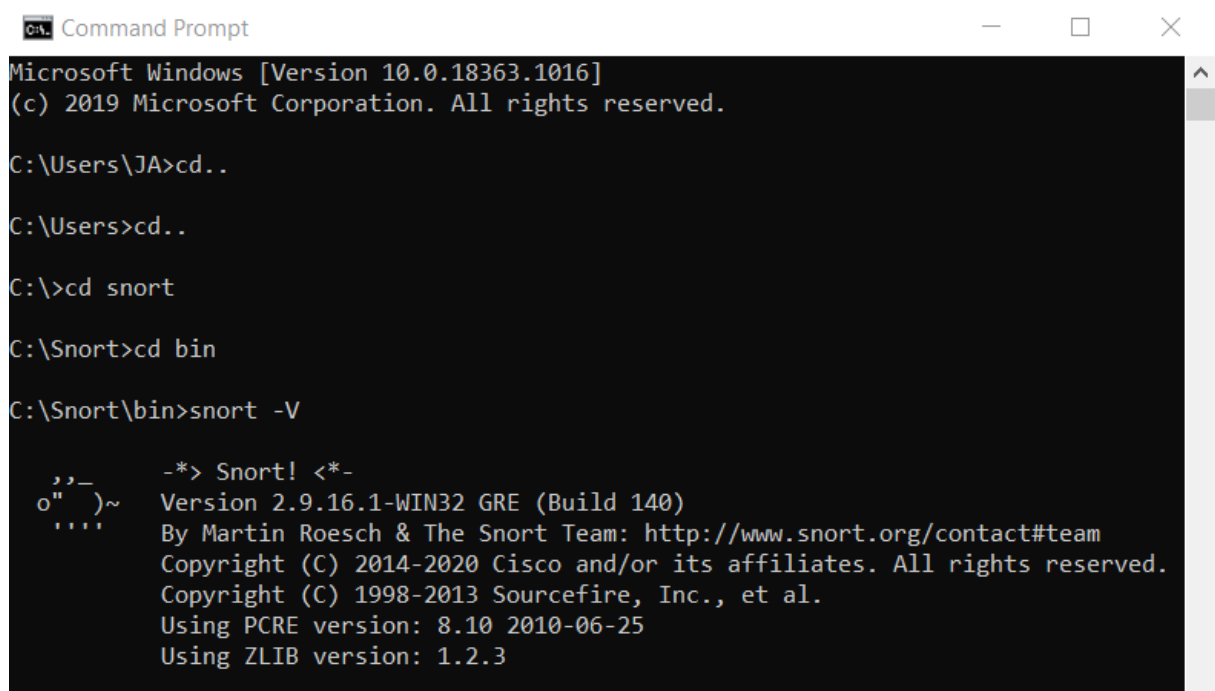
U ovome dijelu prikazano je kako se instalira i konfigurira Snort na Windowsima i na Linuxu

5.3.1. Instalacija i konfiguracija na Windowsu

Iako je instalacija na Windowsu dosta jednostavna, postoji par stvari koje se moraju napraviti kako bi program radio kako treba. Prvo treba skinuti najnoviju verziju Winpcap-a koja daje sustavu pristup mreži te mogućnost ulaska u slojeve mreže niže razine. Uz winpcap još trebamo i skinuti već spomenuta pravila. Pravila se nalaze na službenoj stranici od Snort-a pod downloads te ih raspakiravamo unutar rules foldera. Nakon instalacije Snorta postoje i preproc.rules koja raspakiravamo u folder preproc rules. Oba foldera moramo sami napraviti, ne naprave se tijekom instalacije.

Nakon što smo instalirali Snort i winpcap te raspakirali pravila u svoje foldere možemo ući u command prompt te navigiramo do foldera bin na disku gdje smo instalirali Snort, u ovom projektu je Snort bio instaliran na C: tako da je put do foldera bin bio C:\Snort\bin.

Kada smo u folderu bin pokrenemo provjeru da vidimo ako program reagira, to radimo s naredbom Snort -V. Kada unesemo tu naredbu command prompt treba vratiti osnovke podatke o Snort programu verziju i autora kao što je prikazano na slici 6.



```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\JA>cd..

C:\Users>cd..

C:\>cd snort

C:\Snort>cd bin

C:\Snort\bin>snort -V

  ,,_-
o"  )~
  ' ' '

-*> Snort! <*-
Version 2.9.16.1-WIN32 GRE (Build 140)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2020 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: 1.2.3
```

Slika 6: Testiranje reagiranja programa

Nakon što testiramo reagira li program ili ne reagira, potrebno je otići u folder etc unutar Snorta te naći datoteku nazvanu Snort.conf. Otvaramo tu datoteku s alatom za obradu teksta. U ovom radu korišten je notepad++, ali se može koristiti i obični WordPad. Ako želimo neki dio unutar datoteke ugasiti onda samo ispred linije koda u kojem se nalazi stavimo # tj stavimo liniju koda u komentar.

U datoteci Snort.conf moramo promijeniti većinu stvari kako bi Snort radio kako treba. Prvi korak koji moramo napraviti je konfigurirati IP adresu/e koje sustav nadgleda. To napravimo tako da upišemo IP adresu na način prikazan na slici 7.

```
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.56.0/24
46
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET any
```

Slika 7 Deklariranje IP adresa domaće i vanjske mreže

Postavili smo IP adrese pod ipvar HOME_NET na mrežu 192. 168. 56. 1-254 te smo deklarirali da vanjski IP može biti bilo koji. U vanjski ip se obično može staviti i !\$HOME_NET što znači da je vanjski IP svaki IP koji nije HOME_NET. Zatim trebamo prilagoditi putanje do pravila te ih konfigurirati, a radimo to na način prikazan na slici 8.

```
# Path to your rules files (this can be a relative path)
# Note for Windows users: You are advised to make this an absolute path,
# such as: c:\snort\rules
var RULE_PATH c:\snort\rules
#var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH c:\snort\preproc_rules
```

Slika 8:Konfiguriranje putanje do pravila

Treba napomenuti kako se putanje pišu na drugačiji način kod Linuxa, a na drugačiji način u Windowsima. Preporučljivo je da putanje u Windowsima budu apsolutne što znači da prikažemo cijelu putanju i također kod Windowsa se folderi odvajaju s \ dok je kod Linuxa to s /.

Navigiramo do koraka 4. i tamo mijenjamo putanju do pretprocesora jer su postavljeni za Linux. Pretprocesor je program koji obrađuje svoje ulazne podatke i proizvodi izlazne podatke koje neki drugi program ili dio sustava onda koristi kao ulazne podatke. Snortovi pretprocesori se dijele u dvije kategorije. Prva kategorija se koristi kako bi se pregledali podaci na sumnjive radnje. Druga kategorija mijenja pakete podataka kako bi sustav za detekciju mogao točno

detektirati sumnjive radnje.[39] Dinamičke knjižnice se pokreću prije pokretanja Snorta. Na slici 9 je prikazan izgled konfiguracije ako je Snort instaliran na C:.

```
241 #####
242 # Step #4: Configure dynamic loaded libraries.
243 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
244 #####
245
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\sfe_engine.dll
251
252 # path to dynamic rules libraries
253 #dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
255 #####
```

Slika 9:Konfiguracija dinamičkih knjižnica

Unutar konfiguracije navigiramo do koraka 7 te odabiremo koja pravila ćemo koristiti zavisno o pravilima koja smo instalirali. Ako smo instalirali besplatna pravila bez registriranja onda moramo dodati jednu liniju koda koja vodi do datoteke community.rules i također trebamo komentirati sva pravila ispod local.rules jer su to pravila za registrirane korisnike. Ako smo skinuli pravila za registrirane korisnike onda biramo koja pravila će se priključiti programu prilikom pokretanja. Pravila koja želimo koristiti maknemo # a ona koja ne želimo koristiti stavljamo #.

Community pravila Snorta su pravila koja su napisana od strane članova Snort zajednice. Ona se mogu skinuti na stranici bez registriranja ili plaćanja pretplate. Sva pravila koja se nalaze u Community pravilima se nalaze i u pravilima za registrirane korisnike. Ova pravila se održavaju i ažuriraju no nisu testirana kao pravila za registrirane korisnike ili pretplatnike. Community pravila imaju samo tip alert što znači da samo generiraju upozorenja. Preporučljivo je registrirati se na stranici te skinuti pravila za registrirane korisnike. Na slici 10. može se vidjeti primjer kada su instalirana pravila bez registriranja tj. community.rules.

```

538 #####
539 # Step #7: Customize your rule set
540 # For more information, see Snort Manual, Writing Snort Rules
541 #
542 # NOTE: All categories are enabled in this conf file
543 #####
544
545 # site specific rules
546 include $RULE_PATH/local.rules
547 include #RULE_PATH/community.rules
548 #include $RULE_PATH/app-detect.rules
549 #include $RULE_PATH/attack-responses.rules
550 #include $RULE_PATH/backdoor.rules
551 #include $RULE_PATH/bad-traffic.rules
552 #include $RULE_PATH/blacklist.rules
553 #include $RULE_PATH/botnet-cnc.rules
554 #include $RULE_PATH/browser-chrome.rules
555 #include $RULE_PATH/browser-firefox.rules
556 #include $RULE_PATH/browser-ie.rules
557 #include $RULE_PATH/browser-other.rules
558 #include $RULE_PATH/browser-plugins.rules
559 #include $RULE_PATH/browser-webkit.rules

```

Slika 10:Konfiguriranje pravila Snorta

I završni korak koji moramo konfigurirati je korak 8. Tražimo pravila za preprocesore koje smo uključili te konfiguriramo putanju za njih. Potrebno je konfigurirati putanje kao na slici 11.

```

653 #####
654 # Step #8: Customize your preprocessor and decoder alerts
655 # For more information, see README.decoder_preproc_rules
656 #####
657
658 # decoder and preprocessor event rules
659 include $PREPROC_RULE_PATH/preprocessor.rules
660 include $PREPROC_RULE_PATH/decoder.rules
661 #include $PREPROC_RULE_PATH/sensitive-data.rules
662

```

Slika 11:Konfiguriranje pretprocesorskih pravila

Ovim korakom konfiguracija je gotova te se program može pokrenuti putem naredbenom retka kojeg pokrećemo kao administrator.

5.3.2.Instalacija i konfiguracija na Linuxu

Instalacija na Linuxu je slična kao i instalacija na Windowsu. Instalacija se izvodi putem terminala. Prije nego instaliramo Snort na Ubuntu potrebno je instalirati par knjižnica te Daq source code koji nam je potreban kako bi program radio bez problema. Najnovija verzija Daq source code-a je 2.0.7. Preko terminala skidamo i raspakiravamo Snort i instaliramo te ga

konfiguriramo. Konfiguracija za Linux je različita od Windows instalacije za par malih stvari. Prva razlika je da putanje ne moraju biti apsolutne, a druga da se putanja između mapa označava s /.

6. Testiranje Snorta

Kako bi testirali sposobnosti detekcije Snorta u ovom dijelu smo unutar virtualnog okruženja testirali njegove sposobnosti. Pomoću virtualnog računala Kali smo napali ostala 2 računala, Ubuntu i računalo koje ima Metasploitable2 operacijski sustav. To smo sve detektirali s Snortom koji je instaliran na virtualnom računalu Ubuntu. Sva tri računala su postavljena u virtualnu mrežu.

6.1. Postavljanje virtualnog okruženja

Kako bi ispravno testirali sposobnosti detekcije Snorta potrebno je bilo napraviti pravo okruženje za testiranje. Instaliran je program Oracle VM VirtualBox manager te su na njega instalirana 3 virtualna računala. Na jedno računalo je instaliran 64 bitni Ubuntu na kojem je instaliran Snort.

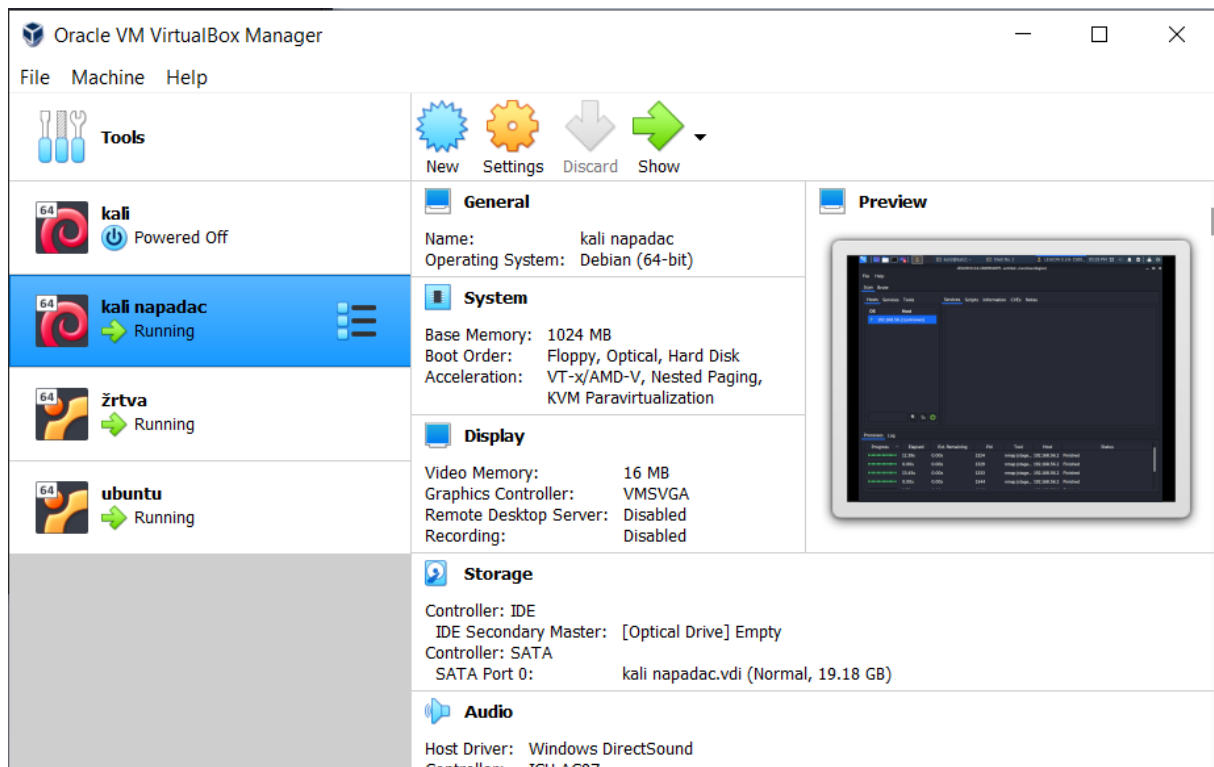
Na drugo računalo je instaliran Metasploitable2. Metasploitable2 je namjerno ranjivo računalo na Ubuntu-u koje se koristi kako bi se testirala slabost ili jačina sustava.

Na treće virtualno računalo je instaliran Kali koji se koristi za napadanje druga dva računala. Kali je Linux baziran na Debianu koji dolazi s unaprijed instaliranim alatima za testiranje informacijske sigurnosti sustava.

Skinuli smo sve ISO datoteke za svako posebno virtualno računalo te smo putem Oracle VirtualBoxa napravili virtualna računala. Virtualna računala se dodaju u oracle box pritiskom na tipku new, zatim nas izbornici vode kroz instalaciju.

Nakon što smo instalirali Snort na Ubuntu prebacili smo sva računala na host only adapter kojeg smo kreirali unutar izbornika Host network manager. Potrebno je prebaciti sva računala u host only kako ne bi imala pristup internetu te kako bi test bio izoliran.

Na slici ispod možemo vidjeti kako su sva tri virtualna računala pokrenuta te kako su spremna za testiranje.



Slika 12: Virtualno okruženje testa

Sva tri računala se nalaze u privatnoj mreži na host only adapteru kako bi mogla međusobno komunicirati i kako bi bila izolirana za svrhu pokusa. Svakom od računala je određena IP adresa:

- Ubuntu – 192.168.56.2
- Kali-192.168.56.102
- Metasploitable2 – 192.168.56.104

6.2. Testiranje detekcije Snorta

Testiranje smo izvršili putem Kali napada na računala unutar mreže. Za početak smo počeli s osnovnim pingom kako bi utvrdili da Snort pravilno radi. Pomoću naredbe ping smo s računala Kali pingali računalo Ubuntu koje je bilo pokrenuto u modu za detektiranje upada. Na slici ispod vidimo pinganje Ubuntu računala te upozorenja koja Snort ispisuje za napad.

```

tion: Misc activity] [Priority: 3] {ICMP} 192.168.56.102 -> 192.168.56.2
09/07-00:32:23.915391  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP}
192.168.56.102 -> 192.168.56.2
09/07-00:32:23.915391  [**] [1:29456:3] PROTOCOL-ICMP Unusual PING detected [**]
[Classification: Information Leak] [Priority: 2] {ICMP} 192.168.56.102 -> 192
.168.56.2
09/07-00:32:23.915391  [**] [1:384:8] PROTOCOL-ICMP PING [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 192.168.56.102 -> 192.168.56.2
09/07-00:32:23.915413  [**] [1:10000001:1] ICMP test [**] [Priority: 0] {ICMP}
192.168.56.2 -> 192.168.56.102
09/07-00:32:23.915413  [**] [1:408:8] PROTOCOL-ICMP Echo Reply [**] [Classifica
+ 1 host(s) tested
kali2@kali2:~$ 192.168.56.2
bash: 192.168.56.2: command not found
kali2@kali2:~$ ping 192.168.56.2
PING 192.168.56.2 (192.168.56.2) 56(84) bytes of data:
64 bytes from 192.168.56.2: icmp_seq=1 ttl=64 time=0.474 ms
64 bytes from 192.168.56.2: icmp_seq=2 ttl=64 time=0.640 ms
^S64 bytes from 192.168.56.2: icmp_seq=3 ttl=64 time=0.545 ms
^C
--- 192.168.56.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.474/0.553/0.640/0.068 ms

```

Slika 13: Ping računala na mreži

Kao što vidimo Snort je izbacio upozorenja za ping. Snort je izbacio i jedno upozorenje koje se zove ICMP test. To je upozorenje koje smo sami napravili i dodali ga u datoteku local.rules. Također, Snort je javio dosta neobičnih pingova te smo tako potvrdili da je Snort dobro konfiguriran i da ga možemo dalje testirati s nekim od alata koji su već instalirani na našem Kali računalu.

6.2.1. Testiranje pomoću alata NMAP

Prva vrsta alata koja je obrađena u ovom radu je Nmap (engl. *Network mapper*). To je alat otvorenog koda za otkrivanje mreža te kako bi se testirala sigurnost. Nmap koristi IP pakete kako bi odredio koja računala su na mreži, koje servise ta računala imaju, koji OS koriste te kakvu vrstu paketa filtera/vatrozida imaju. Nmap se može pokrenuti na skoro svim OS-ovima. Dizajniran je s namjerom da se brzo pregledaju velike mreže ali može pregledati i samo jednog hosta. [27]

Kako bi testirali Snort Nmapom prvo moramo pokrenuti Snort u IDS načinu rada. Zatim na našem Kali računalu pokrećemo Nmap preko terminala s naredbom Nmap i unosimo IP adresu računala kojeg želimo skenirati. U našem primjeru je to bilo računalo Ubuntu, no treba dodati kako Nmap može biti napravljen i za cijelu mrežu. To napravimo tako da umjesto unošenja IP adrese računala na kraju dodamo /24 te bi onda Nmap naredba izgledala ovako nmap 192.168.56.0/24.

Na slici 14. možemo vidjeti Nmap testiranje te upozorenja koja Snort izbacuje kada se skenira računalo. Nmap koristi dva protokola koji koriste portove. TCP protokol i UDP protokol. U našem testu Snort je detektirao TCP zahtjev pokušaj curenja informacija. Možemo vidjeti IP adrese napadača te portove s kojih se napad izvršava i na kojima se napad događa.

```

Commencing packet processing (pid=2189)
09/06-23:48:23.111006  [**] [1:1418:19] PROTOCOL-SNMP request tcp [**] [Classif
ication: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:42168 -
-> 192.168.56.2:161
09/06-23:48:23.117747  [**] [1:1421:19] PROTOCOL-SNMP AgentX/tcp request [**] [
Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:
37248 -> 192.168.56.2:705
Nmap done: 1 IP address (1 host up) scanned in 13.15 seconds
kali2@kali2:~$ nmap 192.168.56.2
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-06 16:48 CDT
Nmap scan report for 192.168.56.2
Host is up (0.00061s latency).
All 1000 scanned ports on 192.168.56.2 are closed
Nmap done: 1 IP address (1 host up) scanned in 13.14 seconds
kali2@kali2:~$

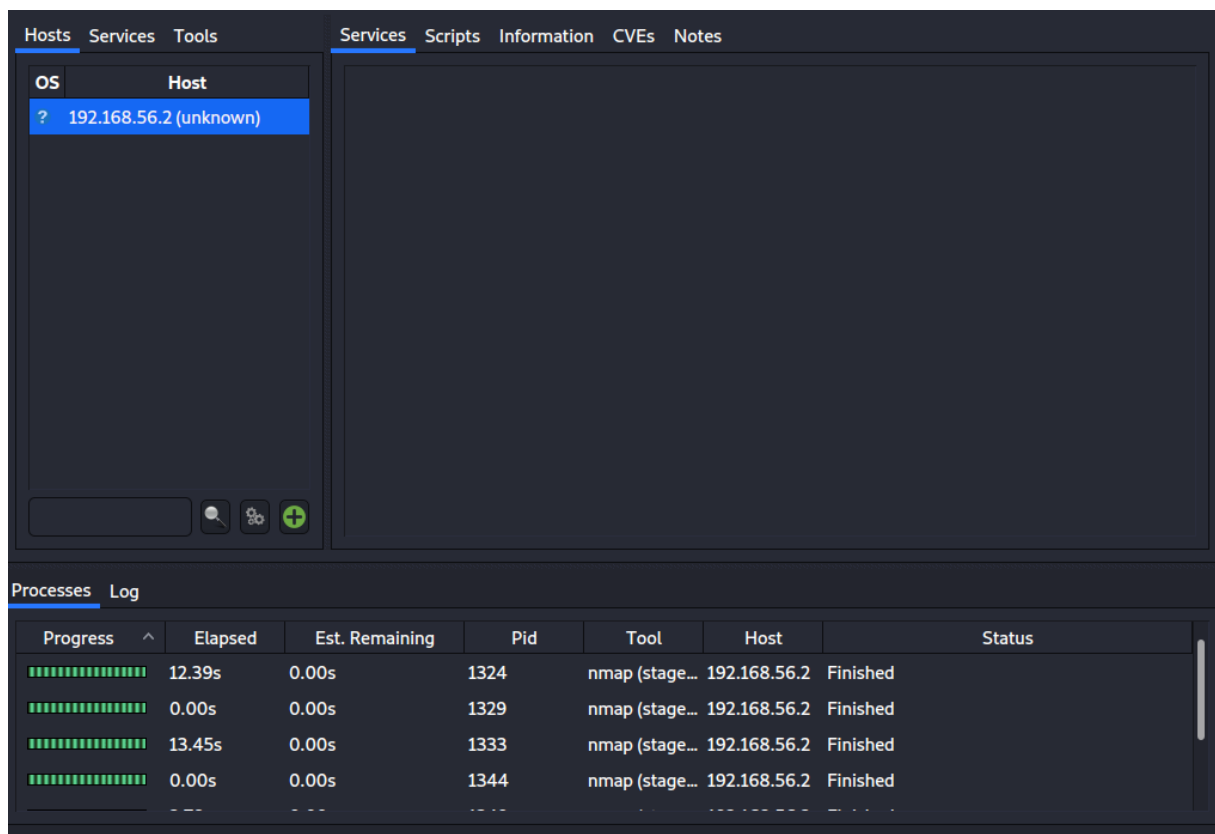
```

Slika 14: Nmap testiranje

6.2.2. Testiranje pomoću alata Legion

Sljedeći alat korišten u ovom radu je Legion. Legion je alat otvorenog koda koji se koristi za testiranje sigurnosti računala ili mreža. Legion pregledava i otkriva slabosti sustava. Sustav je poluautomatski te unutar sebe ima postavljene module od ostalih alata te ih koristi u fazama otkrivanja. Neki od alata koje ima integrirano u sebi su: Hydra, Vulners, SMBenum i gore obrađeni Nmap. [31]

Kako bi testirali sustav s Legion alatom, opet moramo pokrenuti Snort u IDS načinu rada te na Kali računalu pokrećemo Legion s naredbom legion. Nakon što smo unijeli naredbu, otvori se grafičko sučelje u kojem možemo odabrati vrste napada te način napada, kao i odabrati IP adresu/e koje želimo testirati. U ovom radu testirali smo računalo Ubuntu na kojem je i instaliran Snort. Nakon što odaberemo opcije sustav vrši analizu kao što je vidljivo na slici ispod.



Slika 15: Legion testiranje

Legion testiranje obavlja razne radnje poput Nmap i ostalih funkcija koje odaberemo te ga Snort detektira kroz neka od postavljenih pravila kao što je prikazano na slici 16.


```
09/07-00:04:12.813870  [**] [1:1421:19] PROTOCOL-SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:40820 -> 192.168.56.2:705
09/07-00:04:12.856191  [**] [1:1418:19] PROTOCOL-SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:46916 -> 192.168.56.2:161
09/07-00:04:19.683068  [**] [1:1421:19] PROTOCOL-SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:58964 -> 192.168.56.2:705
09/07-00:04:21.310233  [**] [1:1420:19] PROTOCOL-SNMP trap tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:56914 -> 192.168.56.2:162
09/07-00:04:21.444683  [**] [1:1418:19] PROTOCOL-SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:45074 -> 192.168.56.2:161
```

Slika 16: Detekcija legion testiranja

Možemo vidjeti da se zbog različitosti napada unutar Legion alata pojavljuju različita upozorenja jer je pogođeno više pravila. Legion koristi kombinaciju različitih alata kako bi testirao mrežu ili računalo. Kao što se vidi na slici 16. detektirano je više pokušaja curenja informacija. Možemo vidjeti IP napadača te port s kojeg napada kao i IP žrtve te napadnuti port.

6.2.3. Testiranje pomoću alata Nikto

Nikto je alat otvorenog koda koji se koristi za skeniranje web poslužitelja. Testira sustav na opasne datoteke/programme te provjerava zastarjele verzije poslužitelja i probleme vezane za njih. Uz to još provjerava i konfiguraciju poslužitelja i njegove mogućnosti. Nikto pregledava i skenira poslužitelje u kratkom vremenu, ali zbog brzine je lako detektiran od strane sustava za detekciju/prevenciju, ali treba napomenuti kako postoji mogućnost proširenja alata kako bi postao manje vidljiv sustavima za detekciju upada. Nikto dolazi instaliran na sustavu Kali. [29]

Nikto ima podršku za SSL i HTTP. Može spremati svoja izvješća u XML, i HTML formatu. Nikto skenira preko 6000 opasnih datoteka. Provjerava zastarjele verzije preko 1000 servera kao i njihove probleme. Provjerava konfiguraciju servera i mogućnosti HTTP servera. Nikto također ima sposobnost identifikacije web poslužitelja i softver koji taj poslužitelj koristi. [29]. Nikto može skenirati više poslužitelja i portova. Prema početnim postavkama Nikto skenira port 80 no može se koristiti parametar -p kako bi specificirali port koji želimo skenirati. Nikto se od ostalih alata za skeniranje odvaja po velikoj količini skeniranih datoteka i servera.

Pokrećemo Snort na Ubuntu računalu u IDS načinu rada. Pomoću naredbe nikto -h 192.168.56.104 testiramo i skeniramo računalo Metasploitable2. Detektiramo skeniranje pomoću Snorta postavljenog na Ubuntu računalu.

```

09/07-00:10:41.174416 [**] [1:987:32] FILE IDENTIFY .httr access file download request [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.56.102:44268 -> 192.168.56.104:80
09/07-00:10:41.285974 [**] [1:977:25] SERVER-IIS .cnf access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.56.102:44268 -> 192.168.56.104:80
09/07-00:10:41.306473 [**] [1:1242:24] SERVER-IIS ISAPI .ida access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.56.102:44268 -> 192.168.56.104:80
09/07-00:10:41.327976 [**] [1:1129:16] SERVER-WEBAPP .htaccess access [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:44268 -> 192.168.56.104:80
09/07-00:10:41.339704 [**] [1:1131:14] SERVER-WEBAPP .wwwacl access [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:44268 -> 192.168.56.104:80
09/07-00:10:41.455961 [**] [1:1129:16] SERVER-WEBAPP .htaccess access [**] [Classification: LibreOfficeWriter tempted Information Leak] [Priority: 2] {TCP} 192.168.56.102:44270 -> 192.168.56.104:80
09/07-00:10:41.459797 [**] [1:1044:17] SERVER-IIS webhits access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.56.102:44270 -> 192.168.56.104:80
09/07-00:10:41.465591 [**] [1:971:28] SERVER-IIS ISAPI .printer access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.56.102:44270 -> 192.168.56.104:80

initially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changeLog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92467, size: 40540, mtime: Tue Dec 9 11:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/: phpMyAdmin directory found
+ OSVDB-3092: /phpMyAdmin/documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3092: /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ 8726 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time: 2020-09-06 17:11:06 (GMT-5) (26 seconds)

1 host(s) tested
kali2@kali2:~$ sudo nikto -h 192.168.56.104

```

Slika 17: Testiranje Snorta Nikto alatom

Kao što vidimo na slici iznad nikto napadom na ranjivo računalo smo pokrenuli puno pravila unutar Snorta:

- **SERVER-IIS ISAPI .ida access** nam govori da je Snort detektirao iskorištavanje ranjivosti web poslužitelja. Pomoću ove ranjivosti skener dobije putanju do root direktorija. To je pravilo rednog broja 1242 unutar community pravila.
- **SERVER-IIS webhits access** nam govori da je snort detektirao iskorištavanje ranjivosti web poslužitelja. Točnije detektirao je čitanje datoteka od strane napadača. Ovo je pravilo rednog broja 1244 unutar community pravila.
- **FILE IDENTIFY .httr access file download request** nam govori da je Snort detektirao pokušaj dobivanja izvornog koda tako što se na URL dodaje nastavak .httr. Ovo je pravilo rednog broja 987 unutar community pravila.
- **SERVER-WEBAPP .htaccess access** nam govori da je snort detektirao iskorištavanje slabosti aplikacija na serveru. Detektiran je pokušaj ulaska u datoteku htaccess sa skenera. Htaccess datoteka se koristi za konfiguraciju postavki na apache serveru[43]. Ovo je pravilo rednog broja 1129 unutar community pravila.

Na svakom pravilu koje je pokrenuto možemo vidjeti vrijeme kada je pokrenuto što ga je pokrenulo te s koje IP adrese se dogodio upad i na kojem portu se dogodio napad. Kada smo završili sa Snortom za potrebe testiranja izlazimo iz IDS načina rada pritiskom na tipke ctrl+c.

7. Zaključak

U ovom radu su prikazane vrste i povijest sustava za detekciju/prevenciju i navedeni su neki od tih sustava. Naravno tih sustava ima puno više, no ovdje su navedeni neki od najkorištenijih. Postoje komercijalni i besplatni sustavi za detekciju/prevenciju. Provođenjem testiranja unutar virtualnog okruženja Oracle VirtualBoxa pokazano je na koji način Snort sustav detektira određena testiranja ili napade. Snort se pokazao poprilično pouzdan način za detektiranje upada iako je besplatan. Uvjet dobrog rada programa Snort su baza podataka pravila koja se svako malo obnavlja i može se skinuti na službenoj stranici od Snorta te dobro konfiguriran sustav. Treba napomenuti da Snort sustav ima poprilično kompliciran način konfiguracije zbog njegovog neimanja grafičkog sučelja i da je u ovom radu поближе objašnjen način konfiguriranja i postavljanja sustava. Također su opisani načini rada programa uz najveći fokus na načinu rada detekcije upada.

Popis literature

[1]Pirc, John The Evolution of Intrusion Detection/Prevention: Then, Now and the Future, <https://www.secureworks.com/blog/the-evolution-of-intrusion-detection-prevention>

[2]wikipedia. org: Intrusion detection system,

https://en.wikipedia.org/wiki/Intrusion_detection_system

[3] Intrusion Detection Systems: A Deep Dive Into NIDS & HIDS, <https://securityboulevard.com/2020/03/intrusion-detection-systems-a-deep-dive-into-nids-hids>

[4]wikipedia. org:Host based intrusion detection system,

https://en.wikipedia.org/wiki/Host-based_intrusion_detection_system

[5]Techopedia. com: What is Network-based Intrusion Prevention System (NIPS)?,

<https://www.techopedia.com/definition/4030/network-based-intrusion-prevention-system-nips>

[6] WhatIs. com: What is WIPS (wireless intrusion prevention system)?,

<https://whatis.techtarget.com/definition/WIPS-wireless-intrusion-prevention-system>

[7]Techopedia. com: What is a Host-Based Intrusion Prevention System (HIPS)?,

<https://www.techopedia.com/definition/4290/host-based-intrusion-prevention-system-hips>

[8]Suricata. org:all features,

<https://suricata-ids.org/features/all-features/>

[9]Suricata homepage, <https://suricata-ids.org/>

[10]OSSEC homepage, <https://www.ossec.net/about/>

[11]ossec. net: Getting started with OSSEC, <https://www.ossec.net/docs/manual/non-technical-overview.html>

[12]Comparitech: 10 top network intrusion detection tools for 2018, <https://www.comparitech.com/net-admin/network-intrusion-detection-tools/>

[13]Cisco Stealthwatch homepage, <https://www.cisco.com/c/en/us/products/security/stealthwatch/index.html>

[14]Andrea Alilović, [SUSTAVI ZA OTKRIVANJE I SPRJEČAVANJE NAPADA, ZAVRŠNI RAD br. 5947](#), Fakultet elektrotehnike i računarstva, Zagreb, 2019

[15]Soitron. com: Cisco Stealthwatch - Gain visibility of your network, <https://www.soitron.com/solutions-and-services/security/cisco-stealthwatch/#introduction>

[16] Solarwinds. com:main page, <https://www.solarwinds.com/>

[17]Comparitech: 7 best intrusion prevention Systems, <https://www.comparitech.com/net-admin/ips-tools-software/>

[18] Infosec Resources, <https://resources.infosecinstitute.com/open-source-ids-Snort-suricata>

[19] Suricata. readthedocs. io: 6. 35. Differences From Snort — Suricata 6. 0. 0-dev documentation, <https://suricata.readthedocs.io/en/latest/rules/differences-from-Snort.html>

[20]Bricata: Suricata, Snort & Bro: IDS Open Source Technologies, <https://bricata.com/blog/Snort-suricata-bro-ids/>

[21]UserManual. wiki: Snort 3 User Manual, <https://usermanual.wiki/Pdf/Snortmanual.1346323497/help>

[22] [Snort manual: 1. 2 Sniffer Mode](#), <http://manual-Snort-org.s3-website-us-east-1.amazonaws.com/node4.html>

[23] Snort manual, <http://manual-Snort-org.s3-website-us-east-1.amazonaws.com>

[24]winpcap: <https://www.winpcap.org/>

[25]Securityarchitecture. com: Configuring Snort, <https://www.securityarchitecture.com/learning/intrusion-detection-Systems-learning-with-Snort/configuring-Snort/>

[26]Snort homepage, <https://www.Snort.org/>

[27] Tools Kali Linux Nmap, <https://tools.Kali.org/information-gathering/Nmap>

[28] Kali Linux Tutorials: Legion : An Open Source Network Penetration Testing Tool, <https://KaliLinuxtutorials.com/legion-penetration-testing/>

[29]Tools Kali Linux Nikto, <https://tools.Kali.org/information-gathering/nikto>

- [30]Kali homepage, <https://www.Kali.org/>
- [31] Practice Tests, Hackingloops. com: Network Penetration Testing with Legion Framework, <https://www.hackingloops.com/legion-framework/>
- [32] Vacca, J., 2010. *Managing Information Security*. Burlington, MA: Syngress.
- [33] Michael Merced, DealBook: Cisco to Buy Sourcefire, a Cybersecurity Company, for \$2.7 Billion, <https://dealbook.nytimes.com/2013/07/23/cisco-to-buy-sourcefire-a-cybersecurity-company-for-2-7-billion/>
- [34]Solarwinds. com: Security Event Manager, <https://www.solarwinds.com/security-event-manager>
- [35] K. Scarfone, P. Mell, Special Publication 800-94: Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology (NIST) (2007)
- [36]wikipedia.org:Thread(computing) [https://en.wikipedia.org/wiki/Thread_\(computing\)#Multithreading](https://en.wikipedia.org/wiki/Thread_(computing)#Multithreading)
- [37]Lastline: Rootkit Prevention – Understanding Rootkits and the Role They Play in Malware Attacks, <https://www.lastline.com/blog/rootkit-prevention/>
- [38]Snort:Community rules download, <https://www.snort.org/downloads/community/community-rules.tar.gz>
- [39] Jack Koziol,Informit.com: Dissecting Snort | Feeding Snort Packets with Libpcap, <https://www.informit.com/articles/article.aspx?p=101148>
- [40] Business Resource Center: The History Of Intrusion Detection Systems, <https://smallbusiness.yahoo.com/advisor/resource-center/great-applied-technology-typically-needs-enabling/>
- [41]Practice.geeksforgeeks.org:Explain what is Virtual table, <https://practice.geeksforgeeks.org/problems/explain-what-is-virtual-table>
- [42] Paginas.fe.up.pt:Writing Snort Rules, https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm
- [43] Hostinger Tutorials: How to Locate and Create the .htaccess File – A Step-by-Step Guide, <https://www.hostinger.com/tutorials/locate-and-create-htaccess>

Popis slika

| | |
|---|----|
| Slika 1:Packet sniffer mode | 21 |
| Slika 2:Ip a naredba u Linuxu | 22 |
| Slika 3:Pokretanje Snorta u IDS modu..... | 23 |
| Slika 4:Broj učitanih pravila | 24 |
| Slika 5:Lokalna pravila | 25 |
| Slika 6:Testiranje reagiranja programa | 26 |
| Slika 7 Deklariranje IP adresa domaće i vanjske mreže | 27 |
| Slika 8:Konfiguriranje putanje do pravila | 27 |
| Slika 9:Konfiguracija dinamičkih knjižnica | 28 |
| Slika 10:Konfiguriranje pravila Snorta | 29 |
| Slika 11:Konfiguriranje pretprocesorskih pravila | 29 |
| Slika 12:Virtualno okruženje testa..... | 32 |
| Slika 13:Ping računala na mreži..... | 33 |
| Slika 14:Nmap testiranje..... | 33 |
| Slika 15:Legion testiranje | 34 |
| Slika 16:Detekcija legion testiranja | 35 |
| Slika 17:Testiranje Snorta Nikto alatom..... | 36 |