

UNIVERZITET SINGIDUNUM

- MASTER STUDIJSKI PROGRAM -
SAVREMENE INFORMACIONO TEHNOLOGIJE

Musa Murati

Detekcija napada IDS – Sistemi

- Master Rad -

Mentor:
Prof. dr Mladen Veinović

Student:
Musa Murati
Br. Indeksa: 410179/2011

Univerzitet Singidunum
Beograd, 2012

Sažetak

Sistemi za otkrivanje napada su relativno nova tehnologija u celokupnoj istoriji zaštite računarskih sistema. Mrežni napadi koriste razne sigurnosne propuste u operativnim sistemima i korisničkim programima. Nakon izvršenog napada preuzimaju potpunu kontrolu nad računarom. Za napade ovog tipa često se koriste gotove skripte i programi koji mogu uspešno izvršiti pretragu računara za propustom te u slučaju pronalaska uspešno ga i iskoristiti. Mrežna sigurnost jeste najznačajniji deo informacione sigurnosti. Odnosi se na sve funkcije, operacijske procedure, sigurnosne mere, kontrole te politike, potrebne za postizanje prihvatljivog nivoa zaštite za hardversku i programsku podršku te informacije u mreži. Područje mrežne sigurnosti je vrlo široko i složeno, zbog čega ne postoji idealan multifunkcionalan alat, koji bi se odupreo svim pretnjama. Međutim, kombinacija određenih alata daje i više nego dobre rezultate.

Abstract

Intrusion detection systems (IDS) is a relatively new technology in the history of protection in computer systems. Network attacks are using a variety of security vulnerabilities in operating systems and user programs. After the attacks they take complete control of computer. For attacks of this type, there is often used ready-made scripts and programs that can successfully search the computer for a failure and in case of detecting the failure it can be used successfully. Network security is the most important part of information security. It relates to all functions, operating procedures, safety measures, controls and policies which are necessary to achieve an acceptable level of protection for the hardware and software support and information network. The area of network security is very broad and complex, so there is not an ideal multi-functional tool that would withstand all the threats. However, the combination of specific tools gives more than good results.

SADRŽAJ

UVOD 8
1. Sistemi za detekciju napada IDS 10
1.1. Definicija i funkcionalnost 10
1.1.2. Detekcija bazirana na potpisima 13
1.1.3. Detekcija za otkrivanje anomalija 13
1.1.4. Inteligentna analiza protokola 14
1.2. Vrste IDS-a 14
1.2.1. Implementacija pasivne reakcije 16
1.2.2. Implementacija aktivne reakcije 17
1.3. Arhitektura IDS-a 18
1.3.1. Dekoder paketa 20
1.3.2. Pretprocesor 20
1.3.3. Sistem otkrivanja 22
1.3.4. Sistem za vođenje dnevnika i upozoravanje 24
1.3.5. Izlazni moduli 24
2. Položaj u različitim mrežnim topologijama 25
2.1. Smeštaj IDS-a 25
2.2. Rešavanje problema s preklopnicima 27
2.3. Mrežna priključnica 29
3. Problemi i nedostaci IDS-ova 31
3.1. Zaštita IDS-a 32
3.2. Standardizacija 33
3. Potrebe za ids alatima 34
4. Implementacija IDS alata 36
5. Besplatni IDS alati 36
5.1. OSSIM 36
5.2. Razvoj 37
5.3. Alati 38
5.3.1. Arpwatch 39
5.3.2. p0f 40
5.3.3. PADS 41
5.3.4. OpenVAS 41

5.3.5. Snort 42
5.3.6. SPADE 43
5.3.7. Tcptrack 44
5.3.8. Ntop 44
5.3.9. Nagios 45
5.3.10. Osiris 46
5.3.11. Snare 46
6. Komercijalani IDS alati 46
6.1. Prelude 47
6.1.1. Arhitektura 47
6.2. Cisco Intrusion Detection System Appliance IDS-4200 49
6.3. ArcSight 49
7. SNORT 50
7.1. Istorija Snort-a 50
7.2. Snort danas 52
7.3. Komponente Snort - a 52
7.3.1. Paket dekoder 53
7.3.2. Predprocesori 53
7.3.3. Sistem detekcije 54
7.3.4. Sistem za logovanje i uzbune 54
7.3.5. Izlazni moduli 55
7.4. Tipovi Izlaznih modula za uzbune 56
7.4.1. Izlazni moduli za logovanje 57
7.5. Način rada i mogućnosti 59
7.6. Šta je senzor Snort-a i gde postaviti senzor 59
7.7. Lokacija za snort 60
7.8. Optimizacija Snort-a 60
7.8.1. Smanjivanje pogrešnih upozorenja 60
7.8.2. Stvaranje novih pravila 61
7.9. Izbor operativnog sistema za Snort 63
7.10. Primena Snort-a u organizacijama 65
7.11. Poređenje sa drugim IDS / IPS alatima 66
8. Praktičan primer 67

8.1. Postavljanje snort-a u aktivan režim rada 67
8.2. Sniffer način rada..... 68
8.3. Paket log mod..... 70
8.4. Nids mod..... 71
ZAKLJUČAK..... 73
LITERATURA 74

Metodologija istraživačkog projekta

Uvodne napomene i obrazloženje rada

Projektni zadatak ima za cilj da definiše pojmove koji se koriste kada se govori o detekciji napada, bezbednosti, pretnjama i napadima vezane za mreža. Ovaj rad opisuje sigurnosne sisteme za otkrivanje napada, način njihovog rada, strategije implementacije te njihove prednosti i nedostatke.Takođe, treba reći da je težište rada uglavnom okrenuto na opšti prikaz IDS - sistema. Detekcija napada je proces praćenja događaja koji se zbiva u računarskom sistemu ili računarskoj mreži te njihovoj analizi za otkrivanje sigurnosnih problema.Rad može da koristi manje stručnim korisnicima računara, ali i naprednijim korisnicima kao prikaz sistema za detekciju napada.

Predmeti istraživanja

Predmet istraživanja je analiza i potpuno sagledavanje sistema za detekciju napda. Sistemi za detekciju napada su razvijeni prvenstveno da funkcionišu kao alat koji služi za detekciju i sprečavanje napada na računar i/ili računarsku mrežu. Ipak, kod određivanja pravila koje oni kontrolišu i sigurnosne politike firme koju treba da realizuju, moguće je konfigurisati sisteme za detekciju napada da budu i veoma efikasan digitalni forenzički alat, koji će registrovati sve događaje potrebne za naknadnu analizu podataka u slučaju sigurnosnog incidenta.

Ciljevi i zadaci istraživanja

Ovaj rad namenjen je pre svega profesionalcima u oblasti zaštite računarskih mreža, kao i IT stručnjacima koji treba da urade studiju izvodljivosti uvođenja sistema za detekciju napada u neku kompaniju. Ovaj rad ima za cilj da pomogne organizacijama u razumevanju sistema za detekciju napada i tehnologije u projektovanju, implementaciji, konfigurisanju, obezbeđivanju, praćenju, održavanju i detekciju i sprečavanju napada.

Istraživačke hipoteze

Osnovna hipoteza rada

Uz standardne tehnike zaštite računarskih mreža često se koriste i sistemi čija je osnovna namena prepoznavanje sumnjivih aktivnosti i napada na mreži. Cilj ovoga rada je prikazati funkcije i način rada takvih sistema, opisati arhitekturu tipičnog sistema za otkrivanje mrežnih napada, te prikazati najčešće probleme kod njihovog korišćenja kao i kod mogućih rešenja.

Metode i tehnike istraživanja

Metod predstavlja način, put, kojim smo istraživanjem došli do određenih saznanja, otkrili i izložili određenu naučnu istinu, koja se primenjuje u nekoj nauci, u ovom slučaju digitalno-forenzičkoj. Jedna od podela metoda je:

- Posebni naučni metodi i naučni postupci (analitičko-sintetički metod, metodi apstrakcije i konkretizacije, metodi generalizacije i specijalizacije, metodi klasifikacije, metodi indukcije, dedukcije kao metoda saznanja i dijalektičko jedinstvo indukcije i dedukcije)
- Opšti naučni metodi (statistička metoda, metoda modelovanja i analitičko-deduktivna metoda)
- Osnovni opšti metodi društvenih nauka (pozitivistički metod, metod razumevanja, komparativni metod, metod idealnih tipova, metodološki funkcionalizam i strukturalizam i dijalektički metod).
- Metodi korišćeni u radu na ovom istraživanju su: analitičko-sintetički i logički.

Eksperimentalni metodi su oni koji se mogu upotrebiti u istraživanju predmeta koji su podobni za istraživanje eksperimenta. U njih svakako spadaju metodi kojima se može ostvariti neposredan fizički kontakt sa predmetom - realnom stvari. To su analiza i sinteza, apstrakcija i konkretizacija i indukcija.

UVOD

Zbog sve veće prisutnosti računara u raznim područjima poslovanja te zbog sve većeg broja upada i narušavanja sigurnosti istih, došlo je do potrebe za izradom sistema koji će omogućiti nadzor računarskog sistema koji se koristi u poslovanju. Cilj je bio napraviti sistem koji će olakšati nadzor komponenti sistema te davati informacije o sigurnosnim incidentima. Na taj način bi se olakšao i ubrzao postupak otklanjanja pretnji budući da se vreme reagovanja znatno smanjuje. Takođe, uz pomoć sistema za detekciju upada (IDS) i prethodno otkrivenih napada, moguće je implementirati sistem koji sam reaguje na pretnje (IPS) te na taj način povećati sigurnost štićenog sistema.

Sistem za detekciju napada (IDS) je softver koji automatizuje process detektovanja upada. Sistem za prevenciju napda (IPS) je softver koji ima sve mogućnosti za detekciju napada u sistem i može da pokuša da zaustavi mogući incidenti. Ovaj rad daje pregled IDS - sistema. Prvo objašnjava kako se IDS- sistemi mogu upotrebiti. Dalje, on opisuje ključne funkcije koje IDS - sistemi obavljaju i otkrivanje metodologije koje oni koriste. Konačno, on daje pregled glavnih klasa IDS - sistema. Ovaj rad opisuje karakteristike IDS - sistema i daje preporuke za projektovanje, implementaciju, konfigurisanje, obezbeđivanje, praćenje i održavanje. Ovaj rad razmatra sledeće četiri vrste IDS tehnologije.

- **Mrežne (engl. Network-Based)**

  Prate mrežni promet za pojedine segmente mreže ili uređaja te analiziraju mrežu i aplikaciski protokol aktivnosti kako bi mogle identifikovati sumnjive aktivnosti. Mogu identifikovati različite vrste događaja koje smatraju da će biti od interesa.

- **Bežične (engl. Wireless)**

  Prate bežični mrežni promet i analiziraju bežične mrežne protokole kako bi prepoznali sumnjive aktivnosti koje uključuju same protokole.

- **Analiza ponašanja mreža (engl. Network Behavior Analysis - NBA)**

  Ispituju mrežni promet kako bi identifikovali pretnje koje stvaraju neobične prometne tokove, kao što su DDoS napadi, skeniranja te određeni oblici malicioznih virusa.

- **Čvorni (engl. Host-Based)**

Prikupljaju podatke iz internih izvora, obično na nivou operativnog sistema (različite log datoteke) prate aktivnosti korisnika prate izvršavanje sistemskih programa.

Danas je na tržištu dostupan velik broj raznovrsnih alata koji pomažu administratorima u očuvanju sigurnosti računarskih sistema. Neki od dostupnih alata besplatni su i otvorenog koda, dok su neki komercijalni. Neki alati su pak robusni i prilično složeni, dok su drugi jednostavni i lagani za korišćenje. Niz alata koristi se za analizu rada samog sistema, deo njih služi za prevenciju napada i otkrivanje upada u sisteme, a ostali se koriste za korektivne radnje nakon sigurnosnih incidenata. Upravo zbog raznovrsnosti ponuđenih alata administratori često imaju složeni zadatak odabira skupa alata koji će na najbolji način štititi njihov sistem. Raznovrsnost i nepovezanost dostupnih sigurnosnih alata očituje se i kod obrade niza informacija koje oni u svom radu pružaju. Naime, svaki od alata na ovaj ili onaj način generiše specifične izveštaje, zavisno od svoje funkcionalnosti. Velik broj takvih izveštaja vezan je uz otkrivanje sumnjivih aktivnosti, procesa i događaja u sistemu na kojem su pokrenuti. Kako bi se olakšao posao administratora računarskih sistema, osmišljen je OSSIM (eng. *Open Source Security Information Management*). Reč je o besplatnoj distribuciji alata otvorenog koda namenjenih administraciji mrežnih i računarskih sistema, njihovoj zaštiti te uočavanju i prevenciji zlonamernih napada. Ideja OSSIM-a je integracija različitih alata u jedinstven sistem koji pruža širok spektar funkcionalnosti vezanih uz administraciju i zaštitu računarskih sistema. Samim administratorima želi se pružiti centralizovano, učinkovito i pregledno okruženje koje će im omogućiti temeljito upravljanje svim aspektima administracije računarskih sistema. Ona uključuje detaljne analize i praćenje njihova ponašanja, rangiranje i prioritiziranje eventualnih pretnji te brzo poduzimanje učinkovitih korektivnih radnji.

U ovom master radu mi ćemo se pozabaviti jednim od najpopularnijih alata za detekciju upada – Snort-om. Snort je besplatni alat otvorenog koda koji predstavlja mrežni sistem za otkrivanje upada NIDS (eng. *Network Intrusion Detection System*). Jedna od velikih prednosti je ta da se Snort svakim danom nadograđuje, odnosno svakim otkrivanjem novih načina upada u sistem i/ili sigurnosnih propusta on povećava broj (ili koriguje postojeća) pravila. Dodatna prednost je mogućnost prilagođavanje alata specifičnim potrebama korisnika.

1. Sistemi za detekciju napada IDS

1.1. Definicija i funkcionalnost

Sistemi za otkrivanje napada (eng. *Intrusion Detection Systems*) su deo skupa tehnologija za podizanje ukupne razine sigurnosti informaciskih sistema. Njihov rad se zasniva na prikupljanju informacija s čitavog niza mrežnih i računarskih izvora, te analiziranje tih informacija s ciljem otkrivanja eventualnih nedozvoljenih aktivnosti i zloupotrebe sistema na kojem se nalaze. Sistemi za otkrivanje napada (u daljnjem tekstu koristi se u literature uobičajena skraćenica *IDS*) prate i analiziraju mrežni promet i različite delove operativnog sistema, međutim ne poduzimaju nikakve akcije koje bi sprečile štetni ili neželjeni promet. Ukoliko se takav promet uoči, najčešće se samo zabeleži u dnevnik (eng. *log*), te se obaveštavaju nadležne osobe putem različitih mehanizama obaveštavanja. U literature se često razlikuju sistemi koji otkrivaju napade od onih koji otkrivaju pogrešno korišćenje sistema. Uobičajeno se napadom smatra štetna aktivnost koja na sistem deluje sa lokacije izvan sistema, a iste takve aktivnosti koje deluju unutar samog sistema se smatraju pogrešnim korišćenjem.

Napadi se od pogrešnog korišćenja dodatno razlikuju i po svojem intenzitetu odnosno brzini pojavljivanja, važnosti napadnutog objekta itd. Na primer, zaposlenik prijavljen za rad na jednom od računara u lokalnoj mreži koji više puta za redom unese pogrešno korisničko ime kod prijavljivanja na neku mrežnu uslugu u korporaciskom intranetu pogrešno koristi sistem. Isti postupak, ali s računara izvan kompanije lokalne mreže te koji se pojavljuje stotinak puta unutar jedne minute, smatra se napadom. U većini implementacija ovakvih programa najčešće se ne pravi razlika između napada i pogrešnog korišćenja sistema već se oboje tretiraju jednako, kao napad.

Rad IDS-a zasniva se na nadziranju pojedinih delova operativnog sistema, te analiziranju zaglavlja i sadržaja paketa na raznim slojevima mrežnog steka. Obe vrste podataka sadrže informacije o aktivnostima korisnika, pa je na osnovu analize njihovog sadržaja moguće prepoznati neuobičajene aktivnosti i napade. Pri analiziranju mrežnog prometa IDS ni na koji način ne sme usporavati mrežni promet njegov rad mora biti u potpunosti nevidljiv korisnicima mreže i svih sistema na mreži. Iz razloga što analiza celokupnog prometa na visoko propusnim mrežama može biti prilično zahtevan posao, primenjuju se posebne metode, npr. mrežni priključci (eng. *network taps*), da bi se izbegle

takve vrste problema. Analizu mrežnog prometa moguće je obavljati na računaru posebno namenjenom samo svrsi prepoznavanja i praćenja napada na određeni server, aplikaciju ili grupu računara u mreži. Iako je moguće smestiti IDS na isti računar na kojem je instaliran i server koji se želi nadzirati, najčešće se ipak nalazi na posebnom računaru namenjenom posebno prepoznavanju napada. U literaturi se ovakav računar namenjen isključivo radu IDS-a ili nekih od njegovih komponenti naziva *senzor*[1].

Da biste u potpunosti shvatili ovu tehnologiju i uspješno pratili ostatak razmatranja u ovom radu, potrebno je da poznajete nekoliko ključnih pojmova:

Aktivnost je element (karakteristika) izvora podataka koji ima značaja za operatora. Aktivnost izvora podataka obuhvata i one radnje koje mogu biti sumnjive. Primer takve aktivnosti može biti uporno ponavljanje zahteva za TCP vezom koji dolazi sa jedne IP adrese.

Administrator je osoba koja je odgovorna za uspostavljanje sistema zaštite u nekoj organizaciji. Nadležan je za donošenje odluka o postavljanju i konfiguraciji IDS sistema. On mora donositi pravilne odluke o nivoima aktiviranja alarma, obimu pamćenja dešavanja na mreži (*history logging*) i o obimu praćenja sesija. Pored toga, odgovoran je i za definisanje odgovarajućih reakcija na napade, kao i za njihovu realizaciju.

Upozorenje (eng. *Alert*) je poruka koja potiče od analizatora i ukazuje da se desio događaj koji zaslužuje pažnju. On sadrži podatke o aktivnosti i o okolnostima pod kojima se događaj desio. Upozorenje se, primera radi, može generisati u slučaju pojačanog ICMP (*Internet Control Message Protocol*) prometa ili u slučaju više uzastopnih neispravnih pokušaja prijave na sistem (*log on*). Svaka mreža ima određeni nivo prometa koji je potpuno normalan za njeno funkcionisanje. Upozorenja se generišu kada aktivnosti određenog tipa pređu unapred definisane granične vrednosti (pragove). Upozorenje se, primera radi, neće generisati uvek kada neki spoljašni korisnik ispituje server programom *Ping*. Ukoliko, međutim, intenzitet komandi ping poraste ili pređe unapred definisani prag, upozorenje će verovatno biti dobro došlo.

Analizator je komponenta ili proces koji analizira podatke prikupljene pomoću senzora. Pokušava da pronađe sumnjive aktivnosti među prikupljenim podacima. On funkcioniše tako što prati događaje i utvrđuje da li na mreži postoje neuobičajene aktivnosti ili koristi procese zasnovane na pravilima koja se definišu prilikom konfiguracije IDS sistema.

[1] Wikipedia, http://en.wikipedia.org/wiki/Intrusion_detection_system

Izvor podataka (*data source*) je "sirova" informacija koju IDS koristi da bi otkrio sumnjive aktivnosti. Kao izvor podataka mogu se koristiti nadzorne (*audit*) datoteke, sistemske log datoteke i konkretan mrežni promet.

Događaj (eng. *Event*) predstavlja bilo kakvo dešavanje u okviru izvora podataka koje ukazuje na sumnjive aktivnosti. Događaji se beleže u datoteke-dnevnike radi naknadnog proučavanja. Pored toga, oni obično aktiviraju i proces izveštavanja o nesvakidašnjim dešavanjima na mreži. IDS može početi beleženje događaja ukoliko dođe do iznenadnog porasta dolaznih e-mail veza, takav događaj može biti indikator da neko ispituje vašu mrežu. Događaj može aktivirati i upozorenje ukoliko se pojavi odstupanje od uobičajenog mrežnog prometa ili kada dođe do prekoračenja granične vrednosti (praga) neke aktivnosti.

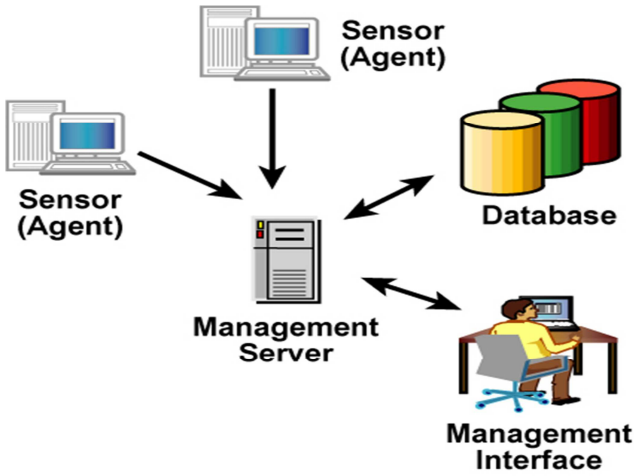
Menadžer je komponenta ili proces pomoću koje korisnik upravlja IDS sistemom. IDS konzola je primer menadžera. Sve izmene IDS konfiguracije obavljaju se korišćenjem IDS menadžera.

Obaveštavanje (eng. *Notification*) predstavlja proces ili metodu pomoću koga IDS menadžer upozorava operatora o neuobičajenim aktivnostima. Obaveštavanje se može realizovati pomoću grafičkih prikaza intenziteta prometa ili pomoću e-mail poruka koje se šalju administrativnom osoblju mreže.

Operator je osoba koja je prvenstveno odgovorna za funkcionisanje IDS sistema (korisnik, administrator, itd), pod uslovom da je najvažnija u lancu odgovornosti.

Senzor je IDS komponenta koja prikuplja podatke iz izvora podataka i prosleđuje ih do analizatora radi dalje analize. Senzor može biti neki postojeći drajver na sistemu ili "crna kutija" koja je priključena na mrežu isključivo radi prikupljanja podataka za IDS. Upamtite da je senzor osnovna tačka za prikupljanje podataka u IDS sistemu.

IDS, kao što možete videti, poseduje veliki broj različitih komponenata i procesa koji funkcionišu kao celina radi kreiranja slike o mrežnom prometu u realnom vremenu.



Slika 1. *Komponente IDS-sistema*

1.1.2. Detekcija bazirana na potpisima

Detekcija bazirana na potpisima je najčešća metoda koja se koristi za prepoznavanje i identifikovanje malicioznih virusa i zlonamernih programa. Ova metoda je ograničena činjenicom da može identifikovati samo ograničeni iznos novonastalih pretnji. Detekcija bazirana na potpisima je vrlo učinkovita u otkrivanju poznatih pretnji, ali uglavnom neučinkovita u otkrivanju ranije nepoznatih pretnji i pretnji prekrivenih pomoću tehnike za izbegavanje. Budući da se novi virusi stvaraju svaki dan, detekcija bazirana na potpisima zahteva česta ažuriranja. Softver za detekciju baziran na potpisima obično ispituje datoteke koje operativni sistem stvara, otvara, zatvara ili ih mail-a. Na taj način može otkriti virus odmah po prijemu. Administratori sistema mogu staviti antivirusni softver da skenira sve datoteke na tvrdom disku u tačno određeno vreme i postaviti određeni datum.

1.1.3. Detekcija za otkrivanje anomalija

Detekcija za otkrivanje anomalija je sistem za otkrivanje upada u računar i zloupotrebe od strane sistema za praćenje aktivnosti. To je proces gde se upoređuju aktivnosti koje se smatraju normalnima te one aktivnosti gde možemo primetiti značajna odstupanja. IDS tehnologije koje koriste detekciju za otkrivanje anomalija imaju profile koje predstavljaju normalno ponašanje stvari kao što su korisnici, domaćini, mrežne veze i aplikacije. Profili su razvijeni tako što su se pratile tipične aktivnosti tokom dužeg vremenskog perioda. Glavna korist od detekcija za otkrivanje anomalija je ta što može biti veoma učinkovita za otkrivanje dosada nepoznate pretnje. Na primer, pretpostavimo da

računar postane zaražen novom vrstom malicioznog programa, te da taj program može konzumirati računarske resurse za obradu, slati veliki broj e-mail-ova, pokrenuti veliki broj mrežnih veza i obavljati druga ponašanja koja se značajno razlikuju od već ranije utvrđenog profila za računar. Maliciozni program bi bio otkriven upravo zbog značajnog odstupanja u odnosu na ranije aktivnosti. Početni profil je generisan tokom nekog vremenskog razdoblja (obično dana, ponekad i nedeljama) a može biti statički ili dinamički. Jednom generisan, statički profil ne može biti promenjen, osim ako IDS tehnologije nisu specifično usmerene da generišu novi profil. Dinamički profil se prilagođuje kako se promatraju novi događaji, znači prilagođuje se stalno i konzistentno. Budući da se sistemi i mreže menjaju tokom vremena, odgovarajuće mere normalnog ponašanja se takođe menjaju skupa s njima. Statički profil će s vremenom postati netačan, tako da treba biti periodički generisan. Dinamički profili nemaju taj problem.

1.1.4. Inteligentna analiza protokola

Ova metoda prepoznaje odstupanje od protokola tako da upoređuje promatrane događaje sa unapred određenim profilima opšte prihvaćenih definicija neškodljivih odnosno benignih aktivnosti. Za razliku od detekcije za otkrivanje anomalija, koja koristi mrežu specifičnih profila, inteligentna analiza protokola zavisi o isporučiteljskim razvijenim univerzalnim profilima koji određuju kako treba i ne treba koristiti određene protokole. Inteligentna analiza protokola može identifikovati neočekivane sekvence naredbi, kao što su recimo izdavanje iste naredbe više puta i slično. Ipak, problemi s njom su to da je često vrlo teško ili nemoguće razviti potpuno tačne modele protokola, te to da inteligentna analiza protokola ne može otkriti napad koji ne krši karakteristike prihvatljivog protokola ponašanja.

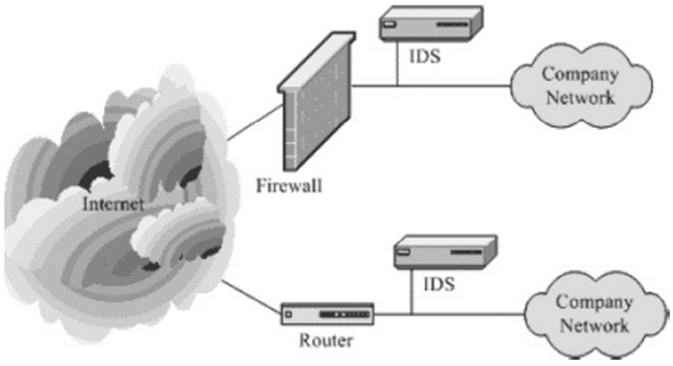
1.2. Vrste IDS-a

Za razliku od klasifikacije sistema za prepoznavanje napada po načinu na koji prepoznaju napade, u literaturi se IDS- sistemi najčešće klasifiikuju prema svojoj nameni. Po toj podeli možemo ih razvrstati u dve osnovne celine:

- Čvorni IDS (*host-based IDS*)
- Mrežni IDS (*network-based IDS*)

Čvorni IDS sistem (H-IDS, *host-based IDS*) dizajniran je u obliku softvera koji se izvršava na računarskim sistemima - hostovima. Takav IDS se obično pokreće u obliku servisa ili pozadinskog procesa, H-IDS prati sadržaj log datoteka na računare, sistemske događaje i interakcije između aplikacija, on obično ne nadgleda dolazni mrežni promet prema hostu. H-IDS sistemi su popularni na serverima koji koriste kriptozaštićene kanale ili kanale prema drugim serverima.

H-IDS sistemi poseduju dva osnovna problema koja nije lako rešiti. Prvi problem je "vezan" za kompromitaciju sistema. Ukoliko dođe do nje, može doći i do narušavanja sadržaja log datoteka u koje IDS upisuje izveštaje. Pogrešan sadržaj tih datoteka može otežati uočavanje grešaka, pa čak i dovesti do nestabilnosti sistema. Drugi problem sa H-IDS sistemima je sadržan u činjenici da se on mora instalirati na svaki računar koji treba zaštititi, što može predstavljati ogromno opterećenje za administratora i osoblje podrške.

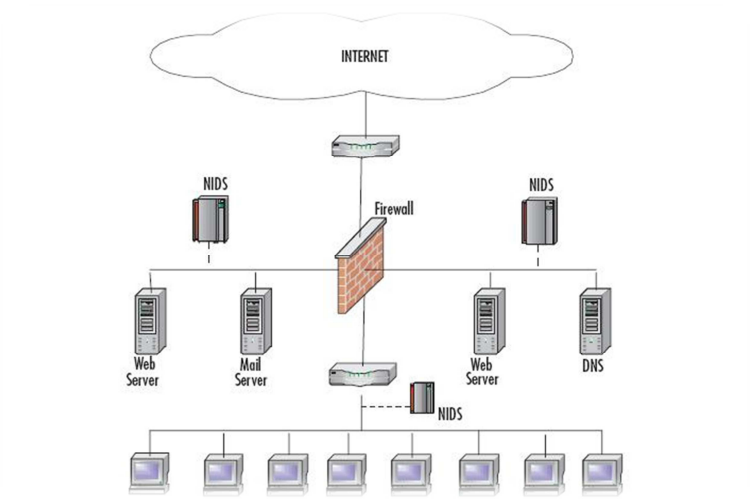


Slika 2. *H-IDS*

Osnovna prednost H-IDS sistema detekcije je mogućnost formiranja kontrolnih suma (*checksum*) za datoteke, radi informisanja administratora o eventualnim izmenama sadržaja datoteka tokom napada. Oporavak od posledica napada je u tom slučaju olakšan, zato što je lako utvrditi na kojim datotekama je došlo do izmena. IDS instaliran na hostovima obično reaguje pasivno na incidente, mada je teoretski moguće predviđene aktivne reakcije slične onima koje se primenjuju u mrežnim IDS sistemima.

Mrežni IDS sistemi (N-IDS, *network-based IDS*) je priključen na takvo mesto u mreži odakle može nadgledati celokupan mrežni promet i izveštavati o stanju. Postavljanjem N-IDS sistema ispred mrežne barijere osigurava se nadzor nad celokupnim dolaznim prometom. Time se osigurava obrada velike količine podataka, uz mogućnost kontrole celokupnog prometa koji dolazi na mrežu. Postavljanjem N-IDS iza mrežne barijere

omogućava se praćenje samo onog prometa koji prolazi barijeru. Mada se time smanjuje količina podataka za obradu, takav pristup ne osigurava uvid u sve napade koji su možda u fazi razvoja.



Slika 3. *N-IDS*

N-IDS se može priključiti na komutator ili na čvorište ili se priključuje na prenosni medij (*tap*). Većina čvorišta i komutatora poseduje specijalni nadzorni port, koji je namenjen za dijagnostiku i otklanjanje kvarova. Takav port može funkcionisati na sličan način kao i običan priključak na kablu. Običan priključak, međutim, ima prednosti, zato što je IDS jedini uređaj koji će ga koristiti. Na nivou mreže je moguće definisati dva osnovna načina reakcije: pasivni i aktivni. Oba su ukratko objašnjena u narednom delu.

1.2.1. Implementacija pasivne reakcije

Pasivna reakcija je najčešći način reagovanja na veliki broj upada u sistem. U principu, ona je najjednostavnija za razvoj i implementaciju. Njena strategija obuhvata:

Evidentiranje (logging) Evidentiranje podrazumeva beleženje pojave određenih događaja i okolnosti pod kojima su oni nastali. Funkcija evidentiranja mora osigurati dovoljno informacija o prirodi napada da bi administrator mogao utvrditi ono što se desilo i proceniti pretnju. Te informacije se, zatim, mogu iskoristiti i za formulisanje odgovora na pretnju.

Obaveštavanje (*notification*) Funkcija obaveštavanja prenosi sve podatke o napadu koji se desio do odgovarajućeg osoblja. To podrazumeva i prenos onih podataka o događaju koji

mogu pomoći u proceni situacije. Ukoliko je IDS sistem neprekidno "pokriven" osobljem, poruke se mogu prikazati i na menadžerskoj konzoli da bi upozorile na pojavu određenih događaja.

Ignorisanje (*shurming*) Ignorisanje napada je jedan od uobičajenih načina reagovanja. Ukoliko IDS detektuje Internet Information Server (*IIS*) napad na sistem na kome se izvršava drugi web-hosting servis, kao što je Apache, napad se može slobodno ignorisati. On neće biti efikasan, zato što Apache ne reaguje na isti način kao i IIS. Na opterećenoj mreži može postojati više različitih napada u jednom trenutku. Ukoliko neki od tih napada ne izaziva previše brige, zašto biste gubili energiju i vreme za njegovo ispitivanje i obaveštavanje drugih korisnika? IDS može jednostavno notirati takav napad u log datotekama i preći na važnije poslove[2].

1.2.2. Implementacija aktivne reakcije

Aktivna reakcija obuhvata poduzimanje određenih mera, u zavisnosti od vrste napada ili pretnje. Cilj takvog reagovanja jeste poduzimanje najbrže moguće akcije da bi bili umanjeni mogući efekti napada. Aktivno reagovanje zahteva postojanje odgovarajućih planova za konkretne događaje, jasnu politiku i određeni oblik inteligencije IDS sistema da bi bila osigurana njegova efikasnost. Aktivna reakcija uključuje jednu od sledećih mera:

Zaustavljanje procesa ili sesije. Ukoliko se detektuje napad radi "gušenja" servisa (*flood atack*), IDS može resetartovati sve tekuće sesije preko određenog podsistema, kao što je TCP. Na taj način se oslobađaju resursi i stvaraju uslovi za normalan rad TCP protokola. Naravno, time se zaustavljaju i sve legalne sesije, koje se moraju ponovo uspostaviti - ponovna uspostava će ipak biti moguća i neće previše opteretiti krajnje korisnike. IDS procenjuje novonastale događaje i određuje najbolji način za njihovo neutralizovanje.

Izmena mrežne konfiguracije ukoliko se detektuje neka IP adresa koja predstavlja izvor stalnih napada na mrežu, IDS može naložiti graničnom ruteru ili mrežnoj barijeri da ubuduće odbije bilo kakav zahtev ili promet koji dolazi sa te adrese. Takva izmena konfiguracije može biti trajna ili se vezuje za određeni vremenski period.

Ukoliko IDS "zaključi" da je napad usmeren na određeni port ili utičnicu, on može naložiti mrežnoj barijeri da blokira taj port u određenom vremenskom intervalu. Time se efikasno

[2] Security Study Guide Second Edition

eliminiše napad, mada se istovremeno može nenamerno prouzrokovati i DoS situacija ("gušenje" servisa) usled eliminisanja legitimnog prometa. To se naročito odnosi na promet pomoću porta 80 (HTTP ili web).

Obmanjivanje (*deception*) Aktivna reakcija u vidu obmanjivanja kod napadača stvara lažnu sliku o uspešnosti napada, osiguravajući, istovremeno, pun nadzor nad događajima, uz eventualno preusmeravanje napadača na specijalno pripremljen sistem. Na taj način operator (ili administrator) može prikupiti podatke o načinu izvršenja napada i upotrijebljenim tehnikama. Čitav proces se često naziva i preusmeravanje na lažni mamac (engl. *honey pot*).

Osnovna prednost ovakvog načina reagovanja jeste mogućnost praćenja i beleženja svih aktivnosti radi naknadne analize nakon završetka napada. Definisanje čitavog scenarija je prilično otežano, dok je puštanje hakera na mrežu relativno riskantan potez, čak i kada potpuno nadgledate sve događaje.

1.3. Arhitektura IDS-a

IDS je logički podeljen na nekoliko zasebnih komponenti. Logička podela može odgovarati i stvarnoj fizičkoj realizaciji sistema, međutim češći je slučaj gde su pojedine komponente udružene u kompleksnije celine ili su dodatno razdvojene na još nekoliko komponenti. Pojedine komponente sistema međusobno komuniciraju s zajedničkim ciljem učinkovitog otkrivanja pojedinih vrsta napada te stvaranja sigurnosnih upozorenja i njihovog snimanja u dnevnik. Komunikacija između pojedinih komponenti ostvarena je razmenom poruka. Primarna namena razdvajanja celokupnog sistema na više međusobno zavisnih komponenti je jednostavnost razvoja, bolja podela zadataka unutar sistema te mogućnost korišćenja modula iz različitih IDS sistema u jednoj raznolikoj (heterogenoj) celini. Pojedine komponente mogu se smeštati i na zasebnim računarima čime se ostvaruje mogućnost izrade sistema koji će maksimalno iskoristiti paralelni rad (npr. *clustering rešenja*) i dati najbolja radna svojstva.

Nažalost, arhitektura pojedinačnih IDS rešenja u stvarnosti dosta varira od proizvođača do proizvođača, bilo zbog nedovoljne primene standarda ili namere proizvođača da onemogući korišćenje pojedinih komponenti i od konkurentskih proizvoda (npr. da se uz pretprocesor jednog proizvođača može koristiti sistem otkrivanja drugog proizvođača). Glavna prepreka komunikaciji komponenti raznih proizvođača je uglavnom nedostatak primene standardnog formata poruka i primene standardizovanih protokola za razmenu tih

poruka. Umesto otvorenih i slobodnih češće se koriste zatvoreni i vlasnički formati te poruke, što predstavlja dodatni problem kod međusobne saradnje različitih komponenti, ali i kod procesa standardizacije. U ovome smislu programi otvorenog koda više pridonose standardizaciji protokola i formata poruka jer je njihova specifikacija kao i implementacija javno dostupna, pa je moguće i zajedničko korišćenje različitih komponenti. Primeri, moguće je kombinovati dva vrlo popularna IDS sistema otvorenog koda, Snort i Prelude, tako da Snort obavlja funkcije otkrivanja napada, a otkrivene napade predaje na daljnju obradu programu Prelude[3].

Tipične komponente od kojih se sastoji sistem za otkrivanje napada s mreže su sledeće:

- Dekoder paketa
- Pretprocesor
- Sistem otkrivanja
- Sistem za vođenje dnevnika i upozoravanje
- Izlazni moduli

Kao što je već navedeno, moguća su i grupisanja pojedinih komponenti sa sličnim funkcijama u jedinstveni modul. Primeri dekoder paketa i pretprocesor mogu biti deo jedinstvenog modula za pripremanje mrežnih paketa, a izlazni modul može biti ugrađen u sistem za vođenje dnevnika i upozoravanje, npr. u slučaju da će se koristiti isključivo zapisivanje upozorenja u datoteku dnevnika.

Moguće je da u nekim implementacijama IDS-ova neke od navedenih komponenti nisu zasebno realizovane ili se uopšte niti ne koriste. Komponente koje se u pravilu nalaze u svakom IDS-u su dekoder paketa, pretprocesor i sistem otkrivanja. Za sistem otkrivanja ponegde se koristi i naziv *engine.*

Sve navedene komponente čine IDS, odnosno IDS senzor. Celokupni sistem za nadzor mreže može se sastojati od jednog ili više senzora. Ponegde se kao posebna vrsta IDS senzora izdvaja i konzola za nadziranje rada sistema. Konzola za nadziranje sistema služi za praćenje rada (*eng. monitoring*) i upravljanje pojedinim delovima sistema, primarno IDS senzorima i njihovim funkcijama. U većini implementacija IDS-ova, konzola za nadziranje sistema nije ništa drugo nego posebno izdvojeni senzor ili još češće svaki senzor ima mogućnost upravljanja ostalim senzorima, tj. ne pravi se razlika između nadzornog (aktivnog) i ostalih senzora. Funkcionalnost nadzornog senzora najčešće se može koristiti

[3] Prelude IDS-a, http://www.prelude-ids.org/spip.php?rubrique6

putem posebnog programa s grafičkim korisničkim interfejsom ili putem web aplikacije. Primer nadzornog programa je Site Protector Console u slučaju ISS IDS-a.

1.3.1. Dekoder paketa

Svi paketi koji dolaze s mreže prvo ulaze u dekoder paketa. Zadatak ove komponente IDS-a je pripremanje paketa s različitih mrežnih interfejsa pre daljnjeg prosleđivanja pretprocesoru. Dekoder paketa omogućuje da se ostatak sistema koristi na jednak način bez obzira na vrstu mrežnog interfejsa koja se koristi na ulazu u IDS, na primer Ethernet, SLIP ili PPP. Pošto u IDS dolaze sirovi paketi (*eng. raw packets*) koji će biti različiti zavisno o vrsti interfejsa s kojeg dolaze, nužno je ostalim delovima sistema osigurati jedinstveni format dekodiranog paketa. Bez ovakvog pristupa u projektovanju IDS-a, sistem bi bio vrlo ograničen jer bi se sve komponente kojima dekoder dalje prosleđuje pakete morali dizajnirati za neku specifičnu vrstu paketa. Primeri, morali bi zasebno biti implementirani pretprocesori za SLIP odnosno PPP interfejsa. Umesto toga, nakon dekodiranja sadržaja dalje se prosleđuje i koristi vlastiti format paketa.

1.3.2. Pretprocesor

Pretprocesor je vrlo važna komponenta IDS-a koja može preurediti i menjati tekstualne ili binarne nizove znakova u paketu pre nego što ih sistem otkrivanja počne detaljnije analizirati. Neke vrste IDS-a su tako dizajnirane da čak i pretprocesor može stvarati sigurnosna upozorenja nakon što otkrije greške u pojedinim delovima paketa (primer, Snort). Pogreške koje na paketima može otkriti pretprocesor su vrlo raznolike i u osnovi se svode na elementarnu proveru zaglavlja u primljenim paketima. IDS može imati i više od jednog pretprocesora. U takvom modularnom slučaju svaki od pretprocesora obavlja uže specijalizovani deo posla koji bi inače obavljao celi pretprocesor. U različitim kombinacijama raspodeljenih i centralizovanih IDS arhitektura obično svaki od senzora ima svoj vlastiti pretprocesor. Prednost IDS-ova koji sadrže više pretprocesora je puno lakši razvoj i nadogradnja sistema, te smanjenje zahteva nad pojedinim senzorima u slučaju da je arhitektura IDS-a takva da se pojedini pretprocesori izvode na različitim računarima.

Uloga pretprocesora može se najbolje opisati na jednostavnom primeru otkrivanja napada na hipotetsku ranjivu mrežnu aplikaciju ili npr. na web serveru. Možemo pretpostaviti

da će svaki bolji sistem za otkrivanje napada sadržavati pravilo koje će stvarati sigurnosno upozorenje ako se tokom nadgledanja HTTP veze (*eng. session*) u jednom od paketa otkrije pokušaj pristupa važnoj sistemskoj datoteci poput administratorske IIS datoteke scripts/iisadmin. Ukoliko sistem otkrivanja doslovce upoređuje staze do datoteka s onima navedenima u bazi potpisa, napadač može zaobići otkrivanje napada na nekoliko vrlo jednostavnih načina. Najjednostavniji takvi načini su korišćenje zamenskih ili escape znakova, korišćenje apsolutnih staza, te korišćenje oznaka "." za pristup tekućem direktorijumu i ".." za pristup roditeljskom direktorijumu. Tako na primer svaka od sledećih staza pokazuje na istu datoteku.

scripts/./iisadmin

scripts/examples/../iisadmin

scripts\iisadmin

scripts/.\iisadmin

Problem se može i dodatno zakomplikovati korišćenjem Unicode (UTF-8 ili UTF-16) heksadecimalnih kodova u URI (*eng. uniform resource identifier*). Web server će uvek moći otkriti da se radi o stazi do datoteke *scripts/iisadmin*, međutim ukoliko IDS tumači staze doslovno onda neće moći otkriti ovakve vrste napada. Uloga pretprocesora je rešavanje upravo ovakvih i sličnih vrsta problema. Pretprocesor se takođe koristi za defragmentaciju paketa. Kada se mrežom prenosi preveliki paket, isti se uobičajeno razdvaja na više delova – fragmenata. Na primer, najveći neprekinuti niz podataka koji se može prenositi u Ethernet mrežama je 1500 okteta. Iznos ove veličine se nadzire pomoću MTU (*eng. maximum transfer unit*) vrednosti mrežnog interfejsa. Ukoliko se mrežom želi prenositi paket čija je veličina veća od vrednosti MTU, paket prvo mora biti razdvojen na delove veličine manje ili jednake vrednosti MTU. Nakon primanja svih delova paketa, primaoc podataka može ponovno sastaviti izvorni paket i očitati izvornu poruku. Fragmentacija paketa je jedan od uobičajenih načina kojim napadači pokušavaju zaobići otkrivanje napada i to ne samo kod IDS-a. Napadač u ovu svrhu može koristiti određena polja u zaglavljima paketa, poput zastavice DF (*eng. don't fragment*). Pre nego što IDS počne proveravati da li primljeni paket ispunjava neki od uslova iz baze potpisa, prvo ga mora ponovno sastaviti iz svih njegovih primljenih delova. U slučaju prethodnog primera s pristupom važnim sistemskim datotekama, jedan deo URI-a se može nalaziti u jednom, a ostatak u drugom delu paketa. Kada bi se baza potpisa počela pretraživati za svaki deo paketa posebno, sistem otkrivanja ne bi mogao otkriti napad. Pretprocesor može rešiti sve prethodno navedene probleme. U većini danas dostupnih IDS-

ova pretprocesor je veoma važna komponenta i obavlja funkcije defragmentacije paketa, dekodiranja URI-a te praćenja i ponovnog sastavljanja (*eng. reassembly*) TCP veza bez kojih IDS ne bi mogao korektno raditi. Potrebno je naglasiti da je pojam pretprocesora prilično široko definisan, te da u praktičnim implementacijama sistema za otkrivanje napada s mreže modeli pretprocesora znaju znatno odstupati od modela opisanog u ovom poglavlju.

1.3.3. Sistem otkrivanja

Sistem otkrivanja je najvažniji deo IDS-a. Njegova uloga je proveravanje da li primljeni paketi ispunjavaju neki od uslova iz baze potpisa i pravila tj. da li je sadržaj paketa u dozvoljenim granicama rada sistma u slučaju da je reč o sistemu koji prepoznaje napade prepoznavanjem neuobičajenog ponašanja. Baza potpisa i pravila organizovana je u interne strukture podataka u obliku povezanih lista (*eng. linked lists*). Pravila su u listama grupisana tako da ako se pronađe određeni uzorak u ulaznim podacima onda se aktivira čitavi niz ispunjenih pravila. Time se izbegava nepotrebno ponovno ispitivanje za isti uslov, ali se i omogućava izrada složenih nizova pravila. Npr, ako postoji pedeset pravila za otkrivanje niza *"/etc/shadow"* unutar paketa, mogu se grupisati unutar jedne liste. Umesto da se paket pretražuje za svako od tih pedeset pravila, pretraživati će se samo jednom za prvi element liste, te ukoliko zadovoljava njegove uslove automatski će se aktivirati svako drugo pravilo iz te liste. Ako paket ispunjava bilo koje od pravila u bazi potpisa, preduzet će se odgovarajuća akcija koja može biti stvaranje sigurnosnog upozorenja, stvaranje zapisa u dnevniku ili oboje. U slučaju da paket nije zadovoljio niti jedno od pročitanih pravila, sistem će ga smatrati ispravnim korišćenjem sistema te ga neće dalje obrađivati.

Sistem otkrivanja je vremenski najkritičnija i najosetljivija komponenta IDS-a. Zavisno o tome koliko je brz računar na kojem se nalazi, te o broju pravila i potpisa, vremenski odziv na različite vrste paketa može jako varirati. Ovo može biti naročito važno ukoliko se od sistema zahteva da radi u stvarnome vremenu (*eng. real time*) gde vreme prepoznavanja nikad ne sme prekoračiti unapred zadanu vremensku granicu. Problem posebno dolazi do izražaja kada sistem radi na mreži s iznimno velikom količinom prometa ili u slučajevima gde je baza potpisa velika. Npr, ako je baza potpisa vrlo velika moguće je da će doći do zagušenja sistema čak i ako IDS radi na mreži s relativno malim prometom, npr. na 10 Mbit vezi. U takvim uslovima sistem neće moći raditi u stvarnome vremenu, a moguće su i situacije gde će neki paketi biti propušteni, a da uopšte nisu bili pretraživani za tragove

napada. U najgorem slučaju može se dogoditi i da se nestručno postavljen IDS u potpunosti zaguši i da prekine komunikaciju na pojedinim segmentima mreže. Kao rešenje ovog problema moguće je primeniti samo zapisivanje paketa, a tek naknadno napraviti pretraživanje njihovog sadržaja na drugom serveru. Npr, ovakav pristup može se ostvariti kombinacijom programa Snort i Barnyard, gde Snort stvara zapise, a Barnyard ih naknadno analizira. Otkrivanje napada može se obavljati na raznim delovima paketa:

- zaglavlja na mrežnom sloju (IP, IPX, ICMP, itd.)
- zaglavlja na prenosnom sloju (TCP, UDP, itd.)
- zaglavlja na aplikaciskom sloju (DNS, FTP, SNMP, SMTP, NNTP itd.)
- sadržaj paketa (*eng. payload*) na aplikaciskom sloju

Sistem otkrivanja može raditi na nekoliko načina. U prvom se načinu za paket koji zadovoljava jedan od uslova odmah stvara sigurnosno upozorenje i zapis u dnevniku, a paket se prestaje dalje obrađivati. Posledica ovakvog načina rada je činjenica da će se stvarati samo jedno sigurnosno upozorenje za paket iako možda zadovoljava uslove više pravila. Ovakva tehnika će doprineti rasterećenju sistema otkrivanja zato jer se neće morati pretraživati ostatak potpisa, međutim predstavlja ozbiljnom ograničenje celokupnog sistema ukoliko bi paket zadovoljavao više pravila, a prvo pronađeno je najmanje važnosti.

Delomično rešenje ovog problema je i prethodno navedeno grupisanje pravila u povezane liste, međutim to samo umanjuje problem jer će još uvek postojati grupe međusobno isključivih pravila koja mogu biti aktivirana istim paketom. Drugo rešenje uzima u obzir i ovakve probleme te svakome pravilu pridružuje i odgovarajuću težinu u zavisnosti o opasnosti koju paket može uzrokovati. Nakon što se pronađe prvo odgovarajuće pravilo za paket, otkrivanje se nastavlja dalje sve dok nisu pretražena sva pravila u bazi potpisa.

Nakon što se pretraži cela baza i sve liste pravila, može se dogoditi da paket zadovoljava i više pravila. Zavisno o tome kako je sistem postavljen, kao konačni rezultat pretrage se mogu prikazati sva zadovoljena pravila ili samo ona najvišeg prioriteta. Kao dodatna optimizacija moguće je da su pravila već unapred poredana po vrednosti prioriteta, pa nema potrebe za pretraživanjem cele liste ukoliko se dođe do zadovoljavajuće razine prioriteta. U tom slučaju prekida se s daljnjim pretraživanjem ostatka liste pravila.

1.3.4. Sistem za vođenje dnevnika i upozoravanje

Zavisno o tome šta sistem otkrivanja pronađe unutar paketa, sistem za vođenje dnevnika i upozoravanje može stvoriti sigurnosno upozorenje te odgovarajući zapis u dnevniku. Samo fizičko spremanje zapisa u npr. datoteku ili bazu podataka obavljaju izlazni moduli. Dnevnik sadrži podatke primljene od sistema otkrivanja i obično se sastoji od sledećih podataka o aktiviranom pravilu te paketu koji ga je aktivirao:

- vreme kada je pravilo aktivirano
- izvorna i odredišne IP adresa paketa
- identifikaciski broj pravila
- opis pravila i referenca na detaljnije informacije
- sadržaj i zaglavlje paketa
- oznaka komponente koja je otkrila napad
- prioritet otkrivenog napada

Većina IDS sistema pruža dodatne opcije za nadziranje ovog dela sistema čime se omogućuje filtriranje određenih vrsta upozorenja ili ograničavanje brzine njihovog zapisivanja. Mogućnost filtriranja i ograničavanja broja zapisa kod stvaranja upozorenja je važna jer će u većini slučajeva IDS stvarati veliki broj lažnih upozorenja iste vrste, te bez razloga puniti dnevnik i stvarati dodatni bespotrebni mrežni promet. Filtriranjem i ograničavanjem može se jednostavno ograničiti broj zapisivanja određene vrste pravila u jedinici vremena ili ga u potpunosti zabraniti.

1.3.5. Izlazni moduli

Izlazni moduli su zaduženi za formatiranje podataka primljenih od sistema za vođenje dnevnika i upozoravanje. Ovaj deo IDS-a određuje format zapisa, npr, u datoteci ili bazi podataka, te definiše metodu dojave sigurnosnog upozorenja. Izlazni moduli su vrlo važna komponenta IDS-a jer omogućavaju rad IDS-a na raznorodnim okolinama. Na primer, sistem otkrivanja i ostale kritične komponente se mogu nalaziti na Linux operativnom sistemu, dok se zahvaljujući izlaznom modulu koji omogućava stvaranje zapisa pomoću SMB poruka dnevnici mogu čuvati na računarima koja koriste drukčiji operattivni sistem poput Microsoft Windowsa.

Dodatni razlog za različite vrste izlaznih formata dnevničkih zapisa je činjenica da se u većini slučajeva datoteke zapisa ne čitaju direktno nego se dalje analiziraju s različitim dodatnim alatima. Korišćenjem dodatnih alata za analizu moguće je dobiti detaljniji uvid u najčešće napade i njihovu strukturu[4]. Neki od uobičajenih i najčešćih korišćenih formata datoteka dnevnika te načina dojavljivanja su sledeći:

- tekstualne datoteke
- XML (eng. Extensible Markup Language) datoteke
- postavljanje SNMP varijabli
- korišćenje usluga operativnog sistema (npr. syslog)
- korišćenje sistema za upravljanje relacionim bazama podataka (RDBMS)
- slanje SMB (eng. Service Message Block) poruka
- slanje upozorenja elektronskom poštom

2. Položaj u različitim mrežnim topologijama

2.1. Smeštaj IDS-a

Zavisno o mrežnoj topologiji na koju se postavlja, IDS senzor može biti smešten na jedno ili više mesta u mreži koja se nadzire. Smeštaj takođe ovisi o prirodi napada koju je potrebno otkrivati: napade s lokalne mreže (*unutrašnje napade*), napade izvan lokalne mreže (*spoljne napade*) ili oboje. Na primer, ako želimo otkrivati samo spoljne napade, a mreža se sastoji od samo jednog usmerivača (*eng. router*) preko kojega lokalna mreža s nekoliko segmenata ima pristup Internetu, najbolje je IDS postaviti odmah iza usmerivača jer će tako sav promet koji dolazi s Interneta u lokalnu mrežu morati proći kroz usmerivać pa samim time i kroz IDS senzor. Ako mreža ima više izlaza na Internet, tada se IDS mora nalaziti na svakoj izlaznoj tački iz mreže. Ako želimo otkrivati i unutrašnje napade, tada je IDS senzore potrebno postaviti na svaki zasebni mrežni segment kako bi senzori imali pristup celokupnom prometu koji prolazi tom mrežom. U većini slučajeva nema potrebe za nadziranjem svakog pojedinačnog segmenta nego se nadziru samo pojedini poverljivi i kritični delovi mreže, npr, segment na kojem se nalazi sigurnosna stena ili neki važniji aplikaciski server, baza podataka itd. U pravilu se IDS senzor u lokalnoj mreži postavlja iza svakog usmerivača i sigurnosne

[4] Sans Institute, Intrusion Detection FAQ, URL: http://www.sans.org/ resources /idfaq/

zaštitne stene. Mesta koja se obično izbegavaju su redundantni delovi mreže poput paralelnih mrežnih veza, nakupina računara (*eng. clusters*), višestrukih preklopnika, sistem s failover rešenjima protiv ispada pojedinih mrežnih kartica itd.

Posebnu pažnju kod postavljanja IDS senzora i dimenzioniranja mreže treba obratiti na način na koji će se obrađivati prikupljeni podaci. Ovi problemi posebno dolaze do izražaja u složenijim konfiguracijama IDS-a gde je potrebno postaviti više od jednog IDS senzora, te kada isti moraju međusobno komunicirati. U takvim slučajevima najčešće je potrebno odabrati rešenje koje će biti kompromis između potrebne procesorske snage te količine prometa koja će se izmenjivati između pojedinih IDS senzora. Podaci iz pojedinih senzora mogu se obrađivati raspodeljeno ili centralizovano. Kod raspodeljenog pristupa podaci se obrađuju u onim senzorima u kojima su i prikupljeni, dok se kod centralizovanog pristupa podaci sakupljeni u senzorima šalju u centralni senzor ili analizator gde će biti podrobnije ispitani. Prednost centralizovanog pristupa projektovanju IDS-a je lakše upravljanje celokupnim sistemom, međutim takvo rešenje znatno povećava količinu mrežnog prometa koji se razmenjuje između pojedinih senzora i centralnog senzora tj. konzole, te zahteva znatno veću procesorsku moć centralnog senzora koji će morati obrađivati događaje iz celog sistema.

Raspodeljeni je pristup znatno teže realizovati i održavati, ali se zato smanjuje količina mrežnog prometa koji se izmenjuje između pojedinih senzora u sistemu, a za razliku od centralizovanog sistema potrebna je puno manja procesorska snaga za obradu prikupljenih podataka. Dodatna činjenica koja ide u prilog raspodeljenom pristupu u oblikovanju IDS je bolja otpornost na ispade u sistemu. Pošto su pojedine komponente u raspodeljenom sistemu potpuno samostalne i ne zavise o niti jednom drugom elementu sistema, ukoliko jedan od senzora u sistemu prestane s radom, ostatak sistema može i dalje ispravno raditi. Centralizovani sistem nije tako fleksibilan i otporan je samo na ispade senzorskih računara, a ukoliko se pokvari centralni senzor ili nadzorna konzola, tada će prestati ispravno funkcionisati i ostatak sistema. U praksi se najčešće javlja kombinacija oba prethodno navedena pristupa pri čemu raspodeljeni senzori prikupljaju podatke i na njima obavljaju neke osnovne akcije (*npr. Pretprocesiranje mrežnih paketa*), te ih zatim prosleđuju jednom ili više centralnih senzora koji će ih podrobnije analizirati te poduzeti potrebne akcije zavisno o njihovom sadržaju. Svim komponentama u takvim sistemima može se upravljati s jednog mesta putem posebne nadzorne konzole. Nadzorna konzola je specijalizovani senzor koji uz mogućnosti otkrivanja napada ima i mogućnost upravljanja drugim senzorima, npr, menjanje

njihovih postavki za vreme rada sistema ili pokretanje i gašenje senzora. Moguće je da nadzorna konzola ima isključivo upravljačke mogućnosti te da ne može otkrivati napade.

Kao što je već navedeno, pravilan smeštaj IDS-a je od ključne važnosti za njegovo pravilno delovanje. Odabirom ispravne pozicije možemo delovati na niz aspekata celog sistema, počevši od količine prometa koji će biti potrebno analizirati, pa sve do procesorske snage koja će biti potrebna za analizu dobivenih podataka. U slučaju da se zahteva da mreža mora sadržavati i demilitarizirano područje, takođe je potrebno smestiti jedan od IDS senzora i u taj segment mreže, iako bi u pravilu postavke za stvaranje sigurnosnih upozorenja u tom području trebale biti puno blaže od onih u privatnim delovima mreže, što zbog same prirode i namene tog područja u kojem se obićno ne nalaze ključni sistemi za poslovanje nego sistemi čiji ispad neće uzrokovati pad cele mreže, te zbog očekivanoga većeg broja prepoznatih napada nego u unutrašnjoj mreži. U protivnome je moguće zagušenje IDS senzora u demilitariziranom području zbog prevelikoga broja otkrivenih potencijalnih napada.

2.2. Rešavanje problema s preklopnicima

Poseban problem u smeštaju IDS-a i pojedinih senzora predstavlja mrežna oprema koja na bilo koji način blokira neku vrstu prometa. Primer takve mrežne opreme je preklopnik (*eng, switch*). Preklopnik će pakete koje dobije na jedan od izlaza proslediti na tačno onaj izlaz koji je priključen na onaj segment mreže na kojemu se nalazi ciljna adresa (*preklopnik radi sa adresama na podatkovnom sloju, tj. s MAC adresama*). Naravno, izuzetak od ovoga je početna faza rada uređaja u kojemu još nisu popunjene sve tablice s podacima o usmeravanju prometa. Ovo predstavlja ozbiljno ograničenje u radu IDS-a jer je u tom slučaju potrebno postavljati senzore na svaki od izlaza preklopnika, tj. na svako zasebno područje vidljivosti što bitno uvećava troškove i vreme potrebno za postavljanje, ali i održavanje takvog rešenja.

Većina preklopnika (npr. kompanije Cisco) su tako dizajnirani i mogu raditi u takvome načinu rada da se TX/RX (*eng. transmit i receive*) linije s jednog ili više izlaza mogu reflektovati na drugi izlaz preklopnika. Takav izlaz na koji se reflektuje se obično naziva SPAN (*eng. Switch Port Analyser*) ili reflektovani izlaz (eng. *mirrored port*). Ukoliko se sensor priključi na takav izlaz, a preklopnik je postavljen tako da reflektovani izlaz koji je priključen prema usmerniku na SPAN izlaz, sav promet koji prolazi od usmernika ka serveru kroz preklopnik dolaziti će i do IDS senzora, ne ometajući pri tome normalnu komunikaciju

prema serverima. Na taj način se izbegava potreba za postavljanjem dodatnih senzora u svaki pojedini segment mreže na koji je spojen preklopnik. Korišćenje SPAN izlaza je vrlo jednostavno i ne zahteva nikakvo dodatno menjanje postojeće mrežne topologije, uz uslov da ga uređaj podržava. Priključivanjem IDS senzora na SPAN izlaz preklopnika nije potrebno menjati konfiguracije ostalih mrežnih komponenti poput tablica prosleđivanja u usmernicima ili pravila u sigurnosnim stenama. Međutim, uz sve dobre osobine, korišćenje SPAN izlaza ima i niz nedostataka. Na jednom preklopniku se može nalaziti samo jedan SPAN izlaz. Ukoliko je potrebno nadzirati više od jednog izlaza preklopnika tada se na SPAN izlaz umesto samo jednog mora reflektovati promet sa čitavog raspona izlaza. Ovakvo reflektovanje prometa sa više izlaza nije poželjno jer može vrlo brzo zakrčiti i preopteretiti SPAN izlaz i znatno smanjiti radna svojstva (*performanse*) rada preklopnika.

Problem je pogotovo izražen ako se nadzire promet u dvosmernoj vezi (*eng. full duplex link*). Nadalje, bez dodatnih promena u konfiguraciji IDS-a, ovakav način rada preklopnika omogućava napadaču napad direktno na IDS senzor spojen na SPAN izlaz, osim ako je zaštićen radom u prikrivenom (*eng. stealth*) načinu rada kako je opisano u radu o zaštiti IDS-a. Još jedan nedostatak ovakvog rešenja je i činjenica da će preklopnik odbacivati pakete za koje izračunato zaštitno CRC polje ne odgovara onome u primljenom paketu. IDS inače analizira i ovakve pakete jer napadači koriste razne tehnike u kojima namerno stvaraju pakete sa krivim CRC poljima kako bi zaobišli mehanizme zaštite i otkrivanja ili otkrili njihovu prisutnost. Ipak, ovakve pakete je moguće videti na samom preklopniku, s podacima o adresi napadača, vremenu napada i broju takvih paketa.

Drugo rešenje problema s preklopnicima je korišćenje dodatne specijalizovane mrežne opreme. Jedno od najdostupnijih i najjeftinijih rešenja je upotreba koncentratora (*eng. hub*). Koncentrator se koristi u paričnim mrežama i radi na takav način da će sve pakete koje dobije na jedan od svojih izlaza proslediti na sve ostale izlaze, tj služi kao obnavljač signala sa više izlaza (*eng. multiport repeater*). Ovakav način rada koncentratora može se iskoristiti za slanje preslike prometa prema IDS-u pri čemu je potrebno napraviti takvu topologiju mreže u kojoj će se koncentrator nalaziti između usmerivaća i preklopnika, a IDS se priključi na dodatni izlaz koncentratora. U ovom slučaju sav promet koji prolazi od usmerivaća prema preklopniku, zahvaljujući koncentratoru biti će prosleđen i preklopniku i IDS senzoru. Potrebno je i napomenuti da je ovakva topologija mreže pogodna samo za otkrivanje spoljnih napada, jer IDS može videti samo promet koji dolazi s Interneta.

Zbog načina rada preklopnika IDS neće videti promet koji međusobno razmenjuju serveri koji se nalaze iza preklopnika. U slučaju da je potrebno nadzirati promet i na tom segmentu mreže, potrebno je dodati još jedan IDS senzor na već pre navedeni SPAN izlaz preklopnika. Korišćenje koncentratora za rešavanje problema sa preklopnicima ima niz prednosti. Koncentratori s četiri izlaza koji se koriste u ovakvim konfiguracijama su relativno jeftini uređaji i vrlo se jednostavno postavljaju u već postojeću mrežnu topologiju. Postavljanjem koncentratora takođe nije potrebno menjati postavke usmerivaća i sigurnosnih stena. Međutim i korišćenje koncentratora ima svoje nedostatke. Ako je veza između usmerivaća i preklopnika dvosmerna, zbog učestalih kolizija paketa propusnost veze može postati bitno smanjena što će znatno smanjiti radna svojstva cele mreže. Kolizije će naročito doći do izražaja ako se uz usmerivać, preklopnik i IDS senzor na četvrti izlaz koncentratora spoji i nadzorna konzola IDS-a što će bitno uticati na tok podataka između usmernika i koncentratora. U zadnje vreme koncentratori se sve manje koriste u ovu namenu, te se češće koriste specijalizovani mrežni uređaji konstruisani upravo za ovakve namene. Primer takvog uređaja je mrežna priključnica (*eng. network tap*).

2.3. Mrežna priključnica

Za rešavanje svih prethodno navedenih problema kod rada IDS-a s preklopnicima danas se najčešće koriste posebni mrežni uređaji dizajnirani upravo tako da zaobiđu sve nedostatke prethodno navedenih pristupa. Mrežna priključnica (eng. *network tap*) je uređaj namenjen preslikavanju prometa u Ethernet, 802.11, FDDI i ATM mrežama. Funkcionalnost uređaja odvija se na fizičkom i podatkovnom sloju, a dodatna je pogodnost što može raditi i na žičanim (*bakarnim*) i na optičkim medijima. Na tržištu su dostupne dve vrste mrežnih priključnica: aktivne i pasivne. Pasivne mrežne priključnice (*eng. passive ethernet tap*) ne koriste nikakvo dodatno napajanje pa mogu dovesti do male degradacije signala na mrežnom kablu na kojem se koriste. Aktivne mrežne priključnice (*eng. active Ethernet tap*) koriste dodatni izvor napajanja.

Priključnica se obično koristi za nadziranje i ispitivanje prometa na najkritičnijim i najzahtevnijim segmentima mreže iz razloga što kvarovi i gubitak napajanja na uređaju ni na koji način neće uticati na povezanost i radna svojstva ostatka mreže. Ponegde se u literaturi izraz tap iz engleskog izvornika tumači kao skr. od engleskog izraza Test Access Port.

Mrežne priključnice su uređaji s tri ili više izlaza. Na primer, mrežna priključnica s četiri izlaza uobičajeno izlaze ima označene sa *A, B, A tap i B tap*. Izlazi *A i B* su podatkovni izlazi i priključuju se izravno na segment mreže koji je potrebno nadzirati, a tap izlazi se priključuju na IDS-ove. Tap izlazi služe za reflektovanje podatkovnih izlaza, tako da tap izlazi A i B reflektuju promet koji prolazi podatkovnim izlazima. Svaki tap izlaz daje identičan promet. Mrežna priključnica se u mrežu postavlja na isto mesto na koje se postavlja i koncentrator. Prednosti korišćenja mrežnih priključnica nad koncentratorom i SPAN izlazom preklopnika su mnogobrojne. Priključnica je tolerantna na ispade u napajanju jer su podatkovni izlazi unutar priključnice spojeni žicom bez ikakvih drugih pasivnih ili aktivnih električkih elemenata pa stoga i ne zavise o spoljnom izvoru napajanja. Moguća je jedino mala degradacija signala u slučaju korišćenja pasivnih izvedbi ovog uređaja. Priključnica ne utiče na protok podataka između usmerivaća i preklopnika, ne zahteva promene u konfiguracijama ostalih mrežnih uređaja poput sigurnosnih stena i usmerivaća, te ne zahteva promenu arhitekture mreže u celini. Nadalje, uređaj ne odbacuje pakete s lošim CRC poljem pa će sav promet dolaziti nepromenjen do IDS senzora koji će moći provesti kompletnu analizu dobivenoga prometa. Konačno, priključnica ne dozvoljava izravno povezivanje napadača na IDS jer je njezino delovanje na mreži potpuno nevidljivo pa pridonosi i celokupnoj sigurnosti sistema. Svi prethodno navedeni argumenti su glavni razlog zašto su danas mrežne priključnice najčešći korišćeni i najprihvatljiviji način postavljanja (*eng. deploying*) IDS senzora u kompleksnija mrežna okruženja. Dva danas najčešće korišćena modela mrežnih priključnica su proizvodi kompanije NetOptics[5] i Finisar[6]. Nedostatak korišćenja priključnica je već spomenuti problem s gubljenjem signala kod pasivnih izvedbi, te još uvije prilično visoka cena aktivnih izvedbi. Dodatna je činjenica i da se bez dodatnih modifikacija većina ovakvih uređaja ne može koristiti za nadzor prometa u oba smera. U slučaju nadzora prometa u oba smera potrebno je primeniti udruživanje više kanala u jedan (*eng. channel bonding*) na IDS senzorima koji se priključuju na takve uređaje. Iako se poslednjih godina proizvode ovakvi uređaju i za primenu u dvosmernim (*eng. full duplex*) segmentima, zbog cene se još uvek vrlo retko koriste. Nadalje, IDS koji je priključen na priključnicu mora biti postavljen za rad u prikrivenom načinu rada, kao što je opisano u poglavlju o zaštiti sistema.

[5] Net Optics, Network Taps and Aggregation Solutions for Passive Network Access, URL: http://www.netoptics.com

[6] Finisar, High Speed Data Communication for Networking and Storage, http://www.finisar.com

3. Problemi i nedostaci IDS-ova

Uz brojne prednosti, postoje i određeni nedostaci korišćenja IDS-ova. Pregledavanjem sadržaja mrežnih paketa često se dozvoljeni promet krivo karakterizuju kao nedozvoljeni, odnosno nedozvoljeni kao dozvoljeni. Do ove pojave dolazi usled previše opštih definicija potpisa pojedinih vrsta napada u slučaju da IDS radi prepoznavanje potpisa. Do grešaka može doći kod prepoznavanja neuobičajenog ponašanja, na primer u dinamičkim okolinama gde se vrata prometa menja vrlo često. IDS-ovi uglavnom ne mogu zaustaviti ili usporiti aktivne mrežne napade, nego se za tu namenu koriste sistemi za sprečavanje mrežnih napada. Zbog mogućih dojava o lažnim napadima na sistem, nije niti je poželjno da IDS-ovi prekidaju uspostavljene veze jer bi to rezultiralo velikim brojem neopravdanih prekida u radu. Posledica činjenice da IDS-ovi ne mogu prekinuti napade je stvaranje onoliko sigurnosnih upozorenja koliko ima dnevničkih zapisa o samim napadima. Pošto se ponekad i ispravni promet neopravdano prepoznaje kao napad, količina upozorenja za sistemske administratore se time dodatno povećava.

Sledeći problem je vremenski raskorak između napada i otkrivanja napada. Iz razloga što IDS mora analizirati veliki broj zapisa te moguće potrebe za naknadnom proverom dojave o napadu od strane sistemskog administratora, IDS-ovi pružaju spor odgovor na napade na računarske i mrežne resurse. Još jedan od nedostataka IDS-ova je činjenica da ne mogu otkriti nove vrste napada, nego mogu ustanoviti samo već postojeće. Kako ovakvi sistemi uglavnom upoređuju sadržaje dnevničkih zapisa i mrežnih paketa s definisanim potpisima za napade, novi napadi nisu sadržani u bazi potpisa pa ih IDS niti ne prepoznaje.

Otkrivanje nepoznatih vrsta napada moguća je samo ukoliko je IDS zasnovan ili kombinovan sa metodom prepoznavanja neuobičajenog ponašanja, međutim i takav pristup ima svojih nedostataka.

Na kraju, u zadnje vreme je prisutna sve veća tendencija enkripcije prometa koji prolazi kroz mrežu. Sve više mrežnih usluga razmenjuje poruke preko sigurnih kanala i virtualnih privatnih mreža (*eng. virtual private network, VPN*). IDS-ovi imaju velikih problema u nadziranju veza zaštićenih IPSec rešenjima ili ostvarenih putem npr. SSH tuneliranja. Nemogućnost prepoznavanja napada u kriptiranim vezama vidljiva je i na primeru *HTTPS-a*, odnosno *HTTP* veza koje se odvijaju u kriptovanom obliku putem *SSL* protokola. Problem nadziranja *HTTPS-a* je naročito važan kada se uzme u obzir sve veći broj različitih napada na web aplikacije. Zbog spomenutih nedostataka jasno je da je nužno razinu sigurnosti

pomaknuti s otkrivanja zlonamernih aktivnosti na njihovo sprečavanje, pa je tako došlo do razvoja sistema za sprečavanje napada sa mreže (*eng. Intrusion Prevention Systems – IPS*). IPS-ovi omogućavaju preventivno delovanje u pogledu računarskih napada i izbegavanje nastajanja štete koja bi bila prouzročena tim napadima.

3.1. Zaštita IDS-a

IDS je poput svake ostale komponente na mreži podložan raznim vrstama napada. Uobičajene su dve mere zaštite IDS-a, rad na prikrivenom mrežnom interfejsom i rad na mrežnom interfejsom bez adrese. Interfejs na kojem radi IDS obično su postavljeni i za primanje i za slanje podataka, te rade u takvome načinu rada da primaju i pakete koji nisu namenjeni direktno njima (*eng. promiscuous mode*). Prikriven interfejs (eng. *stealth interface*) je mrežni interfejs na kojem IDS senzor može samo primati podatke, ali ih ne može slati nazad. Takav interfejs se može ostvariti na dva načina. Prvi način je korišćenje mrežne priključnice koja sama po sebi ne dozvoljava vraćanje prometa, a drugi je korišćenje posebne vrste UTP kabela (*eng. Sniffing cable*). Ova vrsta kabla se može napraviti od običnog UTP kabela na nekoliko načina. Najjednostavnije je na jednom kraju kabela kratko spojiti linije 1 i 2, a linije 3 i 6 se spoje s istim tim linijama na drugom kraju kabla. Umesto da se linije 1 i 2 kratko spoje, između njih se može umetnuti kondenzator koji će služiti kao visoko propusni filter.

Iznos kondenzatora se može izračunati iz izraza $f = 1 / 2 \pi RC$ *i iznosi 150 pF za 10 megabitni te 15 pF za 100 megabitni Ethernet*. Postoje i naprednije izvedbe prikrivenog interfejsa poput takozvanoga *UTP Y-kabla*. Druga metoda zaštite IDS-a je rad na interfejsu bez dodeljene IP adrese. Ako IDS sensor nema dodeljenu IP adresu tada se na njega ne može pristupiti. Postavljanje ovakvog interfejsa je vrlo jednostavno i svodi se na podizanje interfejsa bez adrese i onemogućavanje protokola za automatsko dodeljivanje adrese poput DHCP-a. Senzor obično ima dva interfejsa od kojih je jedan, onaj prema Internetu, bez IP adrese ili ostvareno putem prikrivenog interfejsa, dok je drugo prema lokalnoj mreži ili nadzornoj konzoli, sa dodeljenom IP adresom. Još jedna popularna metoda zaštite su i transparentni IDS premosnici (*eng. Transparent bridges*). Takvi sistemi sadrže dva mrežna interfejsa i mogu se postaviti na bilo koje mesto u lokalnoj mreži što ne zahteva dodatne promene u mrežnoj topologiji ili već postojećim konfiguracijama usmerivača. Transparentni IDS premosnici kombinuju korišćenje Ethernet premosnika s IDS-om i sigurnosnom stenom

što ima niz prednosti nad npr. implementacijom sigurnosne stene na usmerivaću. Pošto se u takvim konfiguracijama sigurnosna stena i IDS nalaze na premosniku, a ne na usmerivaću, to ih čini nevidljivima na mrežnom sloju i stoga ih štiti od napada. Svaki paket prolazi kroz premosnik sve dok IDS ne pronađe potpis napada.

3.2. Standardizacija

Jedan od prvih pokušaja da se standardizuju protokoli i aplikacijsko programski interfejs (*eng. Application Programming Interface – API*) koja se koriste u IDS sistemima rezultovao je stvaranjem zajedničkog IDS okvira poznatog kao CIDF (*eng. Common Intrusion Detection Framework*). Glavni cilj CIDF-a je pojednostaviti deljenje informacija i resursa između različitih IDS sistema te omogućiti da se njihove pojedine komponente mogu ponovno iskoristiti u drugim sistemima. Sam CIDF definisan je u četiri Internet nacrta (*eng. draft*):

- "The Common Intrusion Detection Framework Architecture"
- "Communication in the Common Intrusion Detection Framework"
- "A Common Intrusion Specification Language (CISL)"
- "CIDF APIs: Their Care and Feeding"

Prvi nacrt je najvažniji i opisuje osnovne pojmove u CIDF terminologiji te daje detaljni opis CIDF arhitekture. Ostali dokumenti definišu protokole, format poruka i programski interfejs putem kojih pojedine komponente iz CIDF arhitekture IDS-a mogu međusobno komunicirati na siguran i učinkovit način te obavljati međusobne provere autentičnosti i raspoloživosti.

Komponente sistema su sledeće:

- generatori događaja (eng. event generators, E ‑ box)
- analizatori događaja (eng. event analyzers, A ‑ box)
- baze podataka o događajima (eng. event databases, D ‑ box)
- jedinice za odgovaranje na događaje (eng. response units, R ‑ box)

Sve četiri komponente međusobno komuniciraju razmenom poruka (*eng. Message passing*) standardizovanog formata i korišćenjem jezika CISL (*eng. Common Intrusion Specification Langague*). S vremenom je CISL koji je prilično kompliciran zamenjen jednostavnijim oblikom poruka za razmenu podataka poznatim pod imenom IDMEF (*eng. Intrusion*

Detection Message Exchange Format). Uloga generatora događaja je prikupljati podatke s mreže i slati poruke o događajima ostalim komponentama sistema. Analizatori događaja su komponente na koje se uobičajeno misli kada se govori o IDS-ovima. Njihova uloga je primati poruke od ostalih komponenti, analizirati ih te vratiti poruke koje predstavljaju sažetak analize ulaznih poruka. Baza podataka o događajima i jedinice za odgovaranje na događaje imaju sličnu namenu kao i baza potpisa i izlazni moduli u uobičajenim IDS arhitekturama.

Neke od ideja koje su proizašle iz CIDF projekta potakle su nastanak IETF radne grupe (*eng. Internet Engineering Task Force Working Group*) koja je dobila naziv Intrusion Detection Working Group. Glavni cilj ove radne grupe je predstaviti ideje CIDF projekta široj javnosti. Svi pokušaji standardizacije nastali u okviru IETF-ovih radnih grupa i CIDF projekta datiraju još iz 1998. godine i nisu rezultirali nekim preteranim uspehom, a pojedini delovi standarda koriste se svega u nekoliko dostupnih IDS-ova (*npr. Prelude*), iako se neke ideje iz tih nacrta i danas koriste u promenjenom obliku. Umesto primene ovakvih pravnih i službenih standarda u većini današnjih IDS-ova prisutni su neformalni (*de facto*) standardi proizašli iz nekih od najpopularnijih IDS programa.

3. Potrebe za ids alatima

Velika učestalost napada na računarske sisteme na Internetu implicira nužnost upotrebe IDS alata za provođenjem i uspostavljanjem sigurnosne politike organizacije na javno, ali i interno dostupnim računarskim sistemima. Istraživanja pokazuju da najveći broj neovlašćenih upada na računarske sisteme dolazi upravo sa internih računarskih mreža, odnosno zaposlenika organizacija.

Postoje dva načina sprečavanja neovlašćenih aktivnosti korisnika. Prvi način je izrada potpuno sigurnog računarskog sistema. Moguće je napraviti takav računarski sistem koji će zahtevati od svih korisnika potpunu identifikaciju i autentifikaciju, te koji će zaštititi podatke različitim kriptografskim metodama i vrlo striktnim mehanizmima za kontrolu pristupa. Međutim, ovo rešenje nije moguće izvesti iz nekoliko razloga:

U praksi, nije moguće izraditi potpuno sigurni računarski sistem. Miller[7] daje iscrpan izveštaj o sigurnosnim rupama popularnih programa i operativnih sistema koji ukazuje da (a) ne postoji operativni sistem ili program bez sigurnosnih rupa i (b) da se proizvođači niti ne trude napraviti takav operativni sistem ili program. Dizajniranje i implementacija potpuno sigurnog računarskog sistema je izrazito komplikovan zadatak.

Osim toga, kriptografske metode imaju druge probleme – lozinke je moguće otkriti, korisnici mogu izgubiti ili zaboraviti svoje lozinke i celi kriptografski sistemi mogu biti provaljeni. Osim navedenih sigurnosnih propusta potrebno je napomenuti da i na potpuno siguran sistem može biti provaljeno ukoliko korisnici iznutra zloupotrebljavaju svoje korisničke privilegije.

Drugi način sprečavanja neovlašćenih aktivnosti korisnika je upravo detekcija tih aktivnosti. Ukoliko postoji napad na neki računarski sistem, potrebno je taj napad detektovati što je pre moguće (poželjno u realnom vremenu) da bi se mogla poduzeti neka akcija. Ovo esencijalno predstavlja predmet rada alata za detekciju neovlašćenih upada (IDS-a). IDS alati obično ne poduzimaju preventivne mere kada su detektovane neovlašćene aktivnosti, već obično samo obaveštavaju za to nadležne osobe.

Najčešći način detektovanja neovlašćenih aktivnosti implementiran u IDS alatima je pregledavanje logova generisanih od strane operativnog sistema. Logovi predstavljaju zapise aktivnosti korisnika na računaraskim sistemima koji su zapisani u pojedine datoteke hronološkim redosledom. Budući da se u njih zapisuju sve aktivnosti korisnika neovlašćene aktivnosti je moguće detektovati i ručnim pregledavanjem datoteka sa logovima od strane administratora sistema. Međutim, ove datoteke su izuzetno velike što ručno pregledavanje čini nemogućim. IDS alati automatizuju pregledavanje datoteka sa logovima.

U mnogo slučajeva, čak i nakon napada, važno je analizirati podatke o aktivnostima iz log datoteka kako bi se mogla ustanoviti pričinjena šteta ili otkrivanje neovlašćenih korisnika, kao i koraci koje je potrebno poduzeti da bi se takvi napadi onemogućili u budućnosti. IDS alat se takođe može koristiti za analiziranje tih podataka.

Spaffordov[8] izveštaj koji govori o nužnosti implementacije IDS alata pokazuje da je krađa informacija porasla za 250% u zadnjih 5 godina, da je 99% velikih organizacija prijavilo barem jedan sigurnosni incident i da šteta pričinjena neovlašćenim upadima u

[7] Barton P Miller, David Koski, Cjin Pheow Lee, Vivekananda Maganty, Ravi Murthy, Ajitkumar Natarajan, Jeff Steidl. Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services. Computer Sciences Department, University of Wisconsin, 1995

[8] Eugene H Spafford. Security Seminar, Department of Computer Sciences, Purdue University, Jan 1996

računarske sisteme raste na 10 milijardi US dolara godišnje. Zbog ovde navedenih činjenica vidimo potrebu implementacije dobrog IDS alata na računarske sisteme.

4. Implementacija IDS alata

Prilikom postavljanja IDS alata administratori mrežnih sistema moraju odgovoriti na dva ključna pitanja: Gde je potrebno postaviti IDS alat i kako će funkcionisati taj IDS alat? Što se tiče pitanja gde, IDS može biti postavljen na računaru na kojem rade korisnici, na mreži gde skuplja podatke ili u kombinaciji računara i mreže. Drugo pitanje je komplikovanije. Većina današnjih komercijalnih IDS paketa koristi jedan od dva pristupa za detektovanje neovlašćenih aktivnosti: pregledavanje eksplicitnih potpisa poznatih napada ili pregledavanje za statističke anomalije preko poznatih uzoraka. Ova dva pristupa za detektovanje neovlašćenih aktivnosti u suprotnosti jedan sa drugim. Prvi pristup pregledava eksplicitno podatke za koje se zna da su neovlašćeni dok drugi smatra bilo koju anomaliju implicitno neovlašćenom. Naravno, postoje dobre i loše strane svakog od ovih pristupa.

5. Besplatni IDS alati

5.1. OSSIM

Kao što je već spomenuto u uvodu, OSSIM[9] je besplatna distribucija integrisanih alata otvorenog koda razvijena s ciljem da pruži učinkovito i centralizovano okruženje za administraciju i zaštitu računarskih sistema. Njegovim razvojem želi se pružiti jedinstveno okruženje koje će centralizovati, organizovati te unaprediti otkrivanje i prikaz sigurnosnih podataka na pojedinim računarskim sistemima. Samim time, olakšat će se posao administratora računarskih sistema i omogućiti učinkovitija zaštita pojedinih računara.

OSSIM se sastoji od sledećih elemenata:

- upravljačka ploča namenjena prikazu podataka na visokoj razini apstrakcije,
- komponente za praćenje rizika od pojedinih sigurnosnih incidenata i sumnjivih aktivnosti koje omogućuju nadgledanje sistema na srednjoj razini apstrakcije te

[9] Wikipedia: OSSIM, http://en.wikipedia.org/wiki/OSSIM

- forenzička konzola i komponente za praćenje mrežne aktivnosti namenjene nadgledanju podataka na vrlo niskoj razini apstrakcije.

Cilj integracije postojećih, proverenih sigurnosnih alata u kompaktnu celinu kao što je OSSIM, ujedno je i povećanje osetljivosti te smanjenje nepouzdanosti sigurnosnih izveštaja o radu sistema. OSSIM to postiže kroz sledeće funkcionalnosti:

- međusobna saradnja specijalizovanih alata,
- prioritiziranje u postupku otkrivanja sumnjivih aktivnosti te prediktivnih i korektivnih radnji poduzetih od samog sistema te
- procena rizika od sigurnosnih incidenata koji nastaju kao posledica sumnjivih aktivnosti.

OSSIM distribucija sastoji se od poznatih i proverenih sigurnosnih alata koji unose funkcionalnosti kao što su:

- otkrivanje upada u sistem (eng. *Intrusion Detection System*, IDS) na temelju
- prepoznavanja uzoraka,
- otkrivanje anomalija u radu sistema,
- vatrozidi te
- nadgledanje rada i performansi sistema.

5.2. Razvoj

Kompanija Alien Vault[10] razvila je, i 10. jula 2009. godine objavila prvu službenu verziju OSSIM distribucije. AlienVault je poznata firma sa sedištem u Siliciskoj dolini (eng. *Sillicon Vallei*) koja se bavi razvojem besplatnih i komercijalnih SIEM (eng. *Security Information and Event Management*) rešenja.

Slika 4. *Logo kompanije Alien Vault*

OSSIM je razvijen zbog potrebe da se postojeći korisni sigurnosni alati, od kojih svaki za sebe pruža delić korisnih informacija o stanju nadgledanog sistema, povežu u integrisani

[10] AlienVault: Official Web Site, http://alienvault.com/community

sistem koji će biti olakšan posao, kako u pogledu administriranja sistema, tako i po pitanju njegove zaštite i otkrivanja neovlašćenih upada. Osim toga, cilj je bio napraviti alat koji će na jednom, centralizovanom mestu u računarskoj mreži, upravljati sigurnošću svih njenih udaljenih komponenata[11].

Povezujući dve navedene pretpostavke želeo se izbeći klasičan scenario koji se do tada javljao u poslovnim i ostalim računarskim mrežama. Naime, vrlo je čest slučaj u kojem je, po pojavi određenog sigurnosnog incidenta, koji zbog otežanog načina administriranja sistema nije bio uočen na vreme, došlo do neovlašćenog upada u sistem. Sve skupa je često rezultiralo ozbiljnim posledicama. Takav upad bi se lakše mogao sprečiti upotrebom alata kao što je OSSIM.

Danas je aktuelna verzija 4.0, koja se može preuzeti sa razvojnih stranica alata (http://sourceforge.net/projects/os-sim/). Distribucija je namenjena radu na operativnim sistemima Unik. OSSIM je objavljen pod GPL licencom (eng. *GNU General Public License*). to znači da je sav njegov kod otvoren i dostupan razvojnoj zajednici radi dodatnih modifikacija ili rada na nadogradnji alata. Upravo zato je vrlo velika uloga programerske zajednice u razvoju samog alata. Jedan od ciljeva njegovog razvoja bio je iskoristiti postojeća dostupna rešenja otvorenog koda i povezati ih u svima dostupnu integrisanu celinu. Sav kod je, između ostalog, dostupan na već navedenoj internet stranici. Rad na nadogradnji alata sastoji se od aktivnosti na unapređene alate i modula koji ulaze u osnovnu OSSIM distribuciju te na razvoju raznovrsnih dodatnih funkcionalnosti preko specifičnih programskih dodataka (eng. *plugin*) . Što se tiče matične kompanije , AlienVault je nastavio sa podrškom u razvoju ovog projekta, no sa slabijim intenzitetom nego što je to bilo nedugo po objavi alata . Kompanija se danas okrenula razvoju moćnijih, komercijalnih verzija ovog alata.

5.3. Alati

Osnovna OSSIM distribucija sastoji se od sledećih poznatijih alata:

- Arpwatch,
- p0f,
- PADS,
- OpenVAS,

[11] Karg, D. et al.: OSSIM – General System Description,

- Snort,
- SPADE,
- Tcptrack,
- ntop,
- Nagios,
- Osiris,
- Snare,

Navedeni alati integrisani u jedinstven sistem, pružaju širok spektar funkcionalnosti. Neke od značajnijih su:

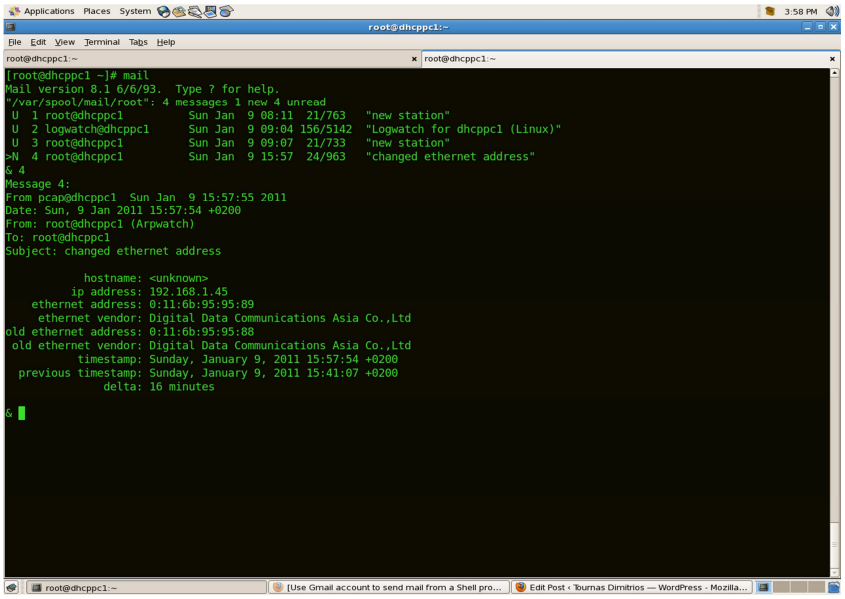
- Otkrivanje anomalija MAC (eng. Media Access Control) adresa,
- Otkrivanje operativnog sistema na pojedinom računaru,
- Otkrivanje anomalija u radu usluga,
- Procena ranjivosti sistema,
- Otkrivanje neovlašćenih upada u sistem,
- Statističko otkrivanje nepravilnosti mrežnih paketa,
- Pregled podataka o uspostavljenim sednicama,
- Izgradnja baze s podacima o uobičajenim aktivnostima sistema i njegovih korisnika te otkrivanje sumnjivog ponašanja,
- Pregled informacija o dostupnosti pojedinih usluga,
- Prikupljanje zapisa o ponašanju korisnika na operativnim sistemima Windows (premda valja naglasiti kako je OSSIM namenjen administraciji i zaštiti svih popularnih operativnih sistema),
- Međusobna poređenje izveštaja i rezultata te
- Pregledan i centralizovan prikaz pojedinih informacija o stanju sistema.

Sledi kratak opis svakog od navedenih programskih alata.

5.3.1. Arpwatch

Arpwatch je programski alat namenjen praćenju paketa koji se razmenjuju protokolom ARP (eng. *Address Resolution Protocol*). Razvilo ga je odeljenje za mrežna istraživanja organizacije Lavrence Berkelei National Laboratori. Objavljen je pod BSD (eng. *Berkelei Software Distribution*) licencom te je otvorenog koda.

Arpwatch prati rad mrežnih interfejsa u lokalnim mrežama prema Ethernet standardu . Pritom gradi održava bazu podataka u koju skladišti sve parove MAC i IP (eng. *Internet Protocol*) adresa te vremenske oznake trenutka kada je pojedino sparivanje uočeno. Dođe li do sumnjivih promena upisanih parova, elektronskom poštom se obaveštava administrator. Mrežni administratori ovaj alat koriste prvenstveno kako bi uočili tzv. ARP spoofing napade Reč je o napadima koji stvore lažno preslikavanje IP i MAC adresa čime dolazi do preusmeravanja saobraćaja poslatog drugim sistemima, najčešće prema računaru napadača. *Slika 5.* prikazuje obaveštenje koje se elektronskom poštom šalje administratoru u slučaju pojave sumnjive promene para IP i MAC adresa na sistemu sa pokrenutim alatom Arpwatch **[1]**.



Slika 5. *Obaveštenje alata Arpwatch o sumnjivoj promeni para IP i MAC adresa*

5.3.2. p0f

OSSIM distribucija sadrži i svestran programski alat za otkrivanje operativnog sistema na nekom računaru u mreži. Njegovo ime je " p0f " . Alat je razvio Majkl Zalevski i namenjen je radu na operativnim sistemima Linux i Windows.

Alat može prepoznati operativni sistem na računaru :

- s kojim je već uspostavljena TCP (eng. *Transmission Control Protocol*) veza,
- s kojim se uspostavlja TCP veza,
- s kojim uspostava TCP veze nije uspela,

- koje pokušava uspostaviti TCP vezu s računara na kojem je alat pokrenut te
- koje komunicira s nekim drugim računarom preko TCP veze koja prolazi kroz računar na kojem je pokrenut alat.

Dodatno, p0f može proveravati i:

- udaljenost od računara na kojem se prepoznaje operativni sistem te njegovo vreme rada (eng. *uptime*),
- prisutnost vatrozida te
- priključak tog računara na mrežu i njegovog internet provajdera (eng. *Internet Service Provider, ISP*).

Po svom se pasivnom načinu rada p0f razlikuje od ostalih alata za prepoznavanje operativnih sistema. On osluškuje mrežni promet bez stvaranja novih, dodatnih paketa. Operativni sistem određuje se na temelju analize određenih polja paketa koji se prihvaćaju. Analizirane vrednosti uspoređuju se s potpisima pojedinih operativnih sistema. Zbog takvog načina rada, udaljeni sistem nije u mogućnosti otkriti delovanje spomenutog alata. Sam alat nema grafičko korisnički interfejs, već se koristi iz komandne linije [2].

5.3.3. PADS

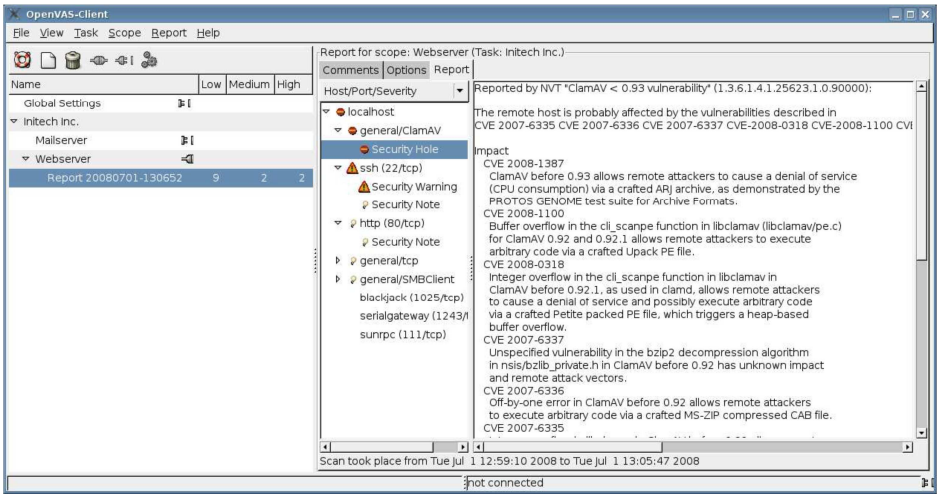
Programski alat PADS (eng. *Passive Asset Detection System*) je kombinacija programa za osluškivanje mrežnog saobraćaja i sistema za otkrivanje mrežnih upada zasnovanih na nizu unapred određenih pravila. Izradio ga je Matt Shelton. Koristi se za osluškivanje mrežnog saobraćaja te pružanje ažuriranih podataka o računarima u mreži i uslugama koje su na njima pokrenute. Praćenjem ovih parametara može se uočiti prisustvo eventualnih sumnjivih usluga. Alat takođe radi pasivno, bez stvaranja dodatnih paketa te je njegov rad gotovo nemoguće primetiti na ostalim računarima u mreži **[3]**.

5.3.4. OpenVAS

OpenVAS je jedan od najpopularnijih alata današnjice koji se koristi za skeniranje ranjivosti na udaljenim računarima. Nastao je kao naslednik programskog paketa Nessus , u trenutku kada je Nessus postao komeracijalan. OpenVAS je alat otvorenog koda izrađen pod GPL licencom. Koristi se za analizu ranjivosti udaljenog računara skeniranjem njegovih priključaka ako pronađe otvorene portove, pokreće daljnje radnje kako bi testirao otpornost

sistema na neke od standardnih napada zlonamernih korisnika. Što se osnovnih funkcionalnosti tiče, OpenVAS može sledeće :

- Odrediti ranjivosti koje udaljenim napadačima mogu omogućiti pristup osetljivim podacima,
- Pronaći pogrešne postavke pojedinih usluga,
- Otkriti nepromenjene podrazumevane lozinke,
- Iskoristiti programski alat Hidra za pokretanje napada rečnikom (napad koji pokušava pogoditi korisničku lozinku poređenjem sa unapred saćuvanim kombinacijama znakova) kako bi testirao bezbednost postavljenih lozinki te
- Pokrenuti napad uskraćivanja usluge posebno oblikovanim paketima kako bi testirao otpornost skeniranog sistema na takve vrste napada.



Slika 6. *Primer izveštaja stvorenog programskim alatom Nessus*

5.3.5. Snort

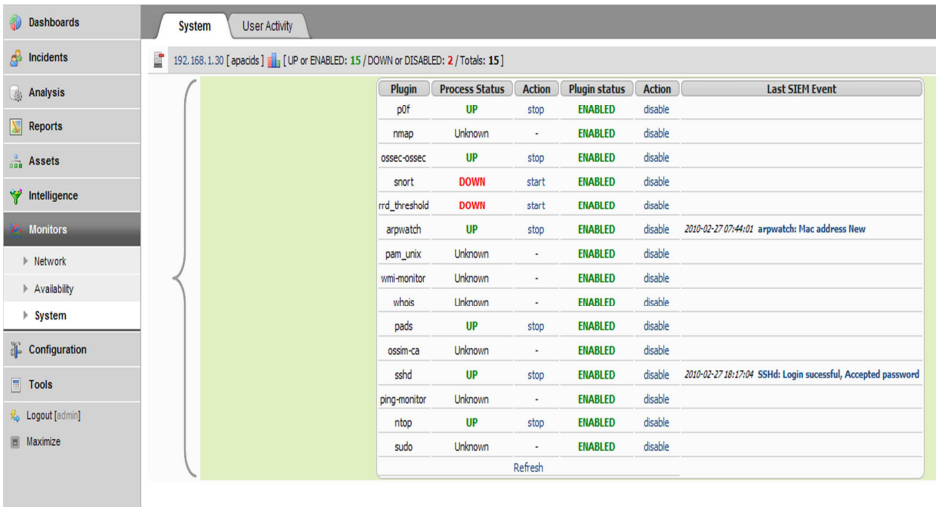
Snort je besplatan programski alat otvorenog koda namenjen otkrivanju i sprečavanju upada u mrežne sisteme. Prvu verziju ovog alata razvio je Martin Roesch 1998. godine. Danas se o razvoju ovog alata brine firma Sourcefire kojoj je Roesch osnivač.

Rad alata zasniva se na mogućnosti izvođenja analize mrežnog saobraćaja i stvaranja odgovarajućih zapisa u realnom vremenu. Snort može obavljati analizu rada pojedinih mrežnih protokola i pretragu za određenim sadržajem te sprovoditi usklađivanje odabranih sadržaja. Isto tako, može se koristiti i za otkrivanje pokušaja skeniranja sistema koji štiti, zlonamernih napada na sam sistem, pokušaja otkrivanja operativnog sistema, prepisivanja bafera itd..

Snort se može podesiti da radi u jednom od tri osnovna načina :

- " Njuškanje " mrežnog saobraćaja (eng. *sniffer*),
- Zapisivanje podataka o mrežnim paketima (eng. *packet logger*) te
- Otkrivanje upada u mrežni sistem (eng. *intrusion detection*).

Kada radi u prvom načinu (" njuškanje " mrežnog saobraćaja), alat presreće mrežne pakete, analizira ih i prikazuje odgovarajuće informacije u konzoli. Za analizu mrežnih paketa te zapisivanje tako dobijenih podataka sve potrebne informacije zapisuju se u posebne datoteke na disku. Napokon , kada alat radi kao sistem za otkrivanje upada u mrežu, pratiće mrežni saobraćaj i upoređivati ga sa nizom pravila koje je postavio administrator. Zavisno o identifikovanju aktivnosti sprovode se odgovarajuće akcije **[4]**. Sledeća slika prikazuje korisnički interfejs OSSIM alata sa delom u kojem se koristi Snort.



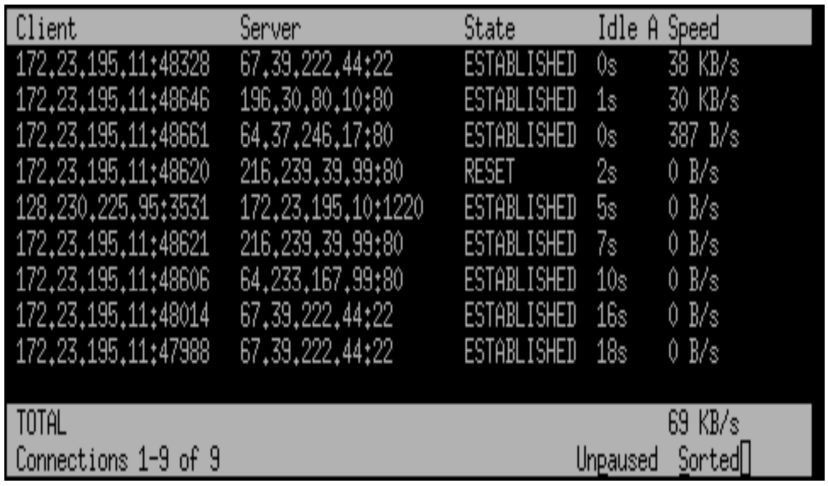
Slika 7. *Snort unutar OSSIM alata*

5.3.6. SPADE

SPADE (eng. *Statistical Packet Anomali Detection Engine*) je programski dodatak namenjen programskom alatu Snort. Snort korišćenjem SPADE dodatka dobija mogućnost slanja dodatnih, detaljnijih obaveštenja po otkrivanju anomalija u posmatranim mrežnim paketima. Ovaj dodatak koristi se za prikupljanje podataka o zlonamernim napadima čiji se potpisi još ne nalaze u poznatim bazama podataka. To mu omogućavaju posebni mehanizmi statističke analize mrežnih paketa te posebni oblici heurističkih algoritama.

5.3.7. Tcptrack

U osnovnu OSSIM distribuciju spada i programski alat " Tcptrack ". Namenjen je analizi mrežnog prometa vezanog uz TCP veze pronađene na mrežnom interfejsu računara na kojem je pokrenut. Alat na pasivan način posmatra veze na mrežnom interfejsu, prati njihova stanja i na pregledan način u konzoli prikazuje osnovne podatke o njima . *Slika 8.* daje primer podataka koje Tcptrack ispisuje.

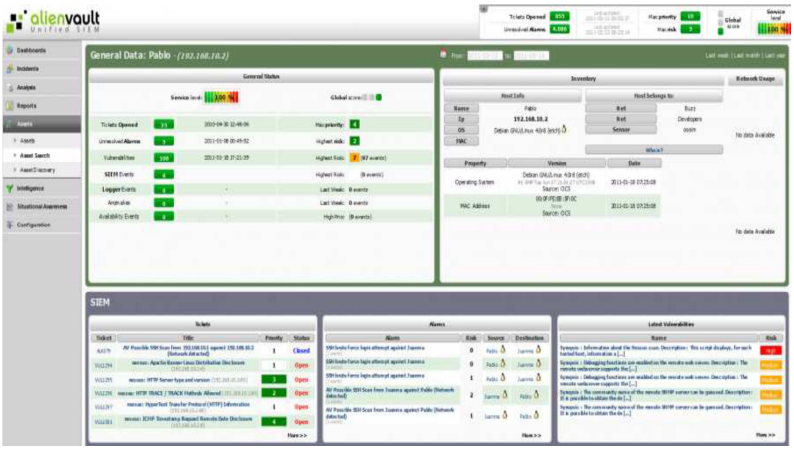


Slika 8. *Podaci o TCP vezama dobiveni programskim alatom Tcptrack*

5.3.8. Ntop

Ntop je programski alat namenjen ispitivanju mrežnog saobraćaja. On na pregledan način prikazuje podatke o zauzeću mrežnih resursa. Ovaj besplatan alat otvorenog koda osmislio je i implementirao Luca Deri.

Alat može raditi na dva načina. Prvi je interaktivni način rada u kojem se na korisničkom terminalu ispisuju podaci o stanju mrežnih resursa. Drugi način je onaj u kojem alat deluje kao web server koji podatke o zauzeću mrežnih resursa prikuplja i čuva u HTML formatu [5]. Sledeća slika daje primer upotrebe alata ntop unutar OSSIM - a.



Slika 9. *Upotreba alata Ntop u OSSIM-u*

5.3.9. Nagios

OSSIM sadrži i popularan alat za analizu mrežnog saobraćaja Nagios. Radi se o besplatnom alatu otvorenog koda kojeg je izvorno razvio Ethan Galstad 1999. godine.

Neke od važnijih funkcionalnosti koje Nagios pruža su sledeće :

- Praćenje i analiza mrežnih usluga (Simple Mail Transfer Protocol - SMTP , Post OfficeProtocol - POP3 , Hypertext Transfer Protocol - HTTP , Network News Transfer Protocol - NNTP , Internet Control Message Protocol - ICMP , Simple Network Management Protocol - SNMP , File Transfer Protocol - FTP , Secure Shell - SSH) ,
- Nadgledanje procesa i resursa na računaru na kojem je pokrenut alat,
- Nadgledanje ostalih sumnjivih aktivnosti,
- Nadgledanje pomoću posebnih skripti pokrenutih s udaljenih računara (eng. remotely – run scripts) preko NRPE (eng. Nagios Remote Plugin Executor) modula,
- Podrška za udaljeno nadgledanje preko protokola SSH i SSL (eng. Security Sockets Layer),
- Jednostavna nadogradnja sistema programskim dodacima,
- Izrada programskih dodataka za grafički prikaz podataka o stanju računarskih i mrežnih resursa,
- Paralelizirane provere pokrenutih usluga,
- Efikasan mehanizam dojave sumnjivih i ostalih aktivnosti (elektronska pošta , SMS itd),
- Mogućnost oblikovanja načina na koje će se određene aktivnosti obraditi s ciljem stvaranja proaktivnog pristupa u rešavanju problema te
- Automatsko beleženje prikupljenih informacija **[6]**.

5.3.10. Osiris

Osiris je jednostavan programski alat koji je prvenstveno namenjen otkrivanju neovlašćenih upada u računar na kojem je pokrenut (eng. *Host - Based Intrusion Detection System , HIDS*). Premda je zastupljen u OSSIM distribuciji, većina funkcionalnosti koje obavlja preklapa se sa onima programskog alata OSSEC.

5.3.11. Snare

OSSIM distribucija sadrži i grupu programskih agenata otvorenog koda koji se koriste za prikupljanje i beleženje podataka o radu s raznovrsnih operativnih sistema. oni omogućuju centralizovano sprovođenje složenih analiza rada međusobno udaljenih, ali povezanih računara s različitim operativnim sistemima. Prva verzija ovog programskog paketa objavljena je 2001. godine. Sami programski agenti unutar OSSIM distribucije služi za prikupljanje podataka i izveštaja dobijenih nadgledanjem pojedinih sistema te njihovo slanje centralnom serveru. Tamo se obavlja skladištenje i arhiviranje podataka te, po potrebi, njihova analiza i stvaranje normativnih izveštaja **[7]**.

6. Komercijalani IDS alati

Iako se teško odlučiti na kupovanje programske podrške, uz dobru i kvalitetnu prilično besplatnu ponudu, komercijalni proizvodi ipak imaju neke prednosti. Naime besplatni IDS proizvodi funkcionišu na elementarnom nivou i zahtevaju osoblje sa puno znanja i iskustva. Nadalje nisu toliko uputstvo orijentisani, pa je nekad uistinu teško raditi sa podacima koje generišu . Nasuprot tome, komercijalni proizvodi su često automatizovani i generišu " lepše " izveštaje, praktičniji su i omogućavaju jednostavniju instalaciju i upotrebu.

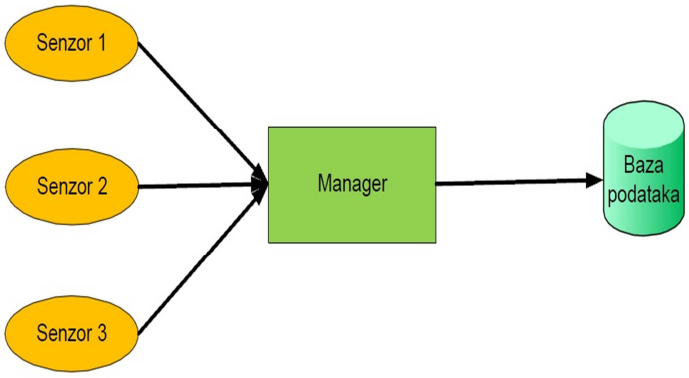
- **Prelude**
- **Cisco Intrusion Detection System Appliance IDS-4200**
- **ArcSight**

6.1. Prelude

Prelude IDS je hibridno IDS okruženje tj program koji svim bezbednosnim programima, bez obzira jesu li otvorenog ili zatvorenog koda, omogućuje prijavljivanje incidenata na centralizovano mesto. Kako bi to omogućio Prelude IDS se oslanja na IDMEF (eng. *Intrusion Detection Message Exchange Format*) standard IETF-a (eng. *The Internet Engineering Task Force*) koji definiše format podataka i procedure za njihovu razmenu između IDS - a i sistema s kojima IDS - ovi interagiraju. " Hibridno " u opisu programa Prelude znači da je moguće kombinovati podatke dobijene iz bilo kojeg programa za bezbednost.

6.1.1. Arhitektura

Prelude okruženje sastoji se od dva osnovna dela : senzora i menadžer . Senzori su svi programi koji prikupljaju podatke o bezbednosnim događajima. Manadžer je deo programa koji prima podatke od senzora te ih skladišti ili ih prosleđuje drugoj instanci menadžera.

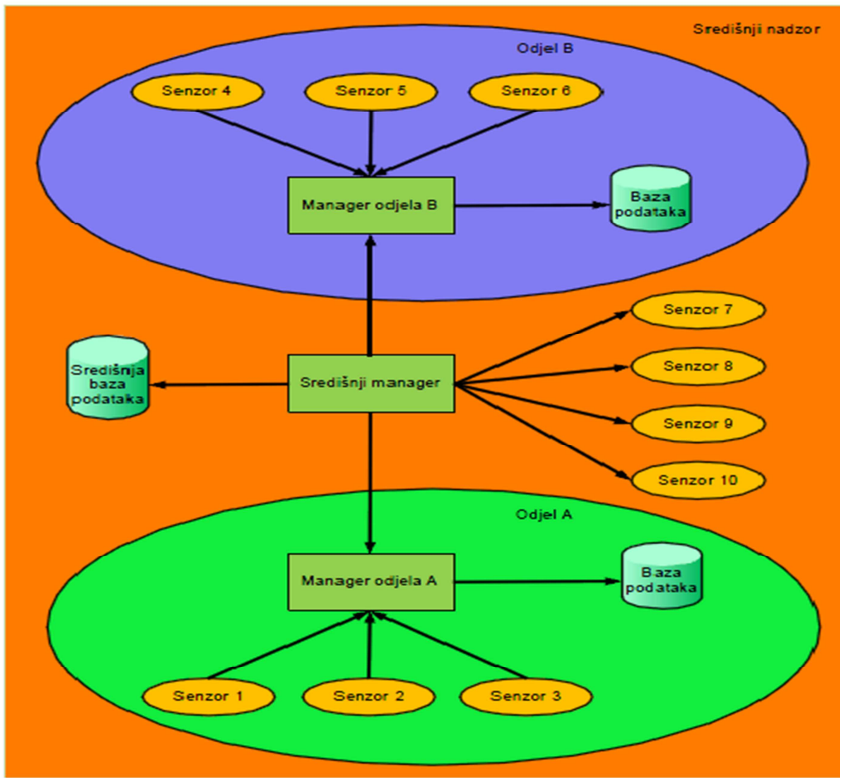


Slika 10. *Prikaz najjednostavnije arhitekture Prelude sistema*

Prelude je moguće instalirati u nekoliko načina rada odnosno arhitektura. U najjednostavnijem načinu sve komponente (senzori i prelude - menadžer) nalaze se na istom računaru. Takva arhitektura prikazana je na slici 10. Osim ovog jednostavnog načina moguće je okruženje postaviti tako da jedan prelude menadžer prosleđuje sve informacije drugom prelude - menadžeru . Na slici 11. prikazan je primer složenije arhitekture u kojoj menadžeri iz odeljenja A i B prosleđuju sve informacije centralnom menadžeru. Menadžeri odeljenja A i

B deluju nezavisno jedan o drugom. Podaci prikupljeni u odeljenju A nisu vidljivi menadžeru u odeljenju B i obrnuto.

Ovakva arhitektura svakom odeljenju omogućava nadzor nad sopstvenim podsistemom, ali istovremeno obezbeđuje i središnji nadzor celokupnog sistema. Svako odeljenje u ovom prikazu može se sastojati od proizvoljnog broja računara tj svaki senzor prikazan na šemi može se nalaziti na zasebnom računaru ili može predstavljati sve senzore na jednom računaru. Takođe je moguće unakrsno povezati dva menadžera kako bi se osigurao integritet podataka u slučaju kompromitacije jedne instance menadžera. Osim prikazanih postoji i mogućnost tzv. obrnutog prosleđivanja. Ono je potrebno kada prelude menadžer koji se nalazi u jednoj mreži ne može pristupiti menadžeru u drugoj mreži. Primer ovakve situacije je mreža zatvorena vatrozidom (eng. *Demilitarized Zone - DMZ*) koji blokira spajanje na spoljnu mrežu. U takvoj se situaciji prelude - menadžer iz spoljne mreže može spojiti na menadžer u unutrašnjoj mreži i od njega zatražiti snimanje informacije.



Slika 11. *Prikaz složene arhitekture*

6.2. Cisco Intrusion Detection System Appliance IDS-4200

Cisco IDS može analizirati kako zaglavlje tako i sadržaj svakog paketa. Cisco IDS može analizirati pojedinačne pakete tako i kompletan tok napada, dok održavanje toka stanja, omogućava detekciju napada multi - paketa. Cisco IDS koristi *rulebased expert system* za ispitivanje paketa informacija te da se utvrdi tip napada, bilo da je jednostavan ili složen.

Cisco IDS je zasebni proizvod gde se svi prikupljanji podaci i analiza vrši na hardversku platformu. Ove jedinice treba da budu postavljene na strateškim tačkama u celom mrežnom sistemu gde se ispituje mrežni saobraćaj. U odgovor na napad, Cisco IDS ima nekoliko opcija koje uključuju generisanje alarma, evidentiranje alarm događaja i prekidanje TCP sesija. Cisco IDS može daljinski upravljati na dva načina. Prvi je preko Veb stranicama preko transportnog sloja bezbednost veze. Drugi je kroz CLI preko Secure Shell (SSH) veze.

6.3. ArcSight

ArcSight proizvod obuhvata upravljanje bezbednošču, proizvod je dizajniran da prati, analizira i izveštava o mrežnim anomalijama identifikovanih od treće strane uređaja za nadgledanje mreže (npr. IDS Senzori ili IDS Skeneri, firevall).

ArcSight onda obezbeđuje IDS drugog reda u tome što pruža širok poduhvat nadgledanja pod- mreže koje nisu homogene mreže monitora. Kao takav , ArcSight nudi rešenje za upravljanje sve mrežne događaje i / ili aktivnosti u preduzeću iz centralizovanog gledišta.

ArcSight omogućava pouzdanim korisnicima da prate događaje, koreliraju događaje za dubinsko istraživanje i analizu, i rešili događaje sa povećanim automatizovanim postupcima i akcijama.

ArcSight Konzola je centralizovana prikaz u preduzeću koje pruža u realnom vremenu praćenje, dubinski istražne mogućnosti i automatizovane reakcije I odgovarajuće rešenje na događaje.Konzola pruža administratorima , analizator administratora, i operatore sa intuitivnim interfejsom za menadžera obavljanja bezbednosne funkcije upravljanja koji uključuje pregledavanje revizorske podatke.

ArcSight Manager je motor visokih performansi koji upravlja, unakrsne- korelacije, filtere, i obrađuje nastanak štetnog događaja u okviru preduzeća. ArcSight Menager se nalazi

u centru ArcSight i deluje kao veza između ArcSight konzolu, ArcSight bazu podataka, i ArcSight SmartAgent.

ArcSight Baza je mehanizam za logićan pristup, određene šeme , tabelarni prostor , podela i raspored diska. ArcSight Baza skladišti sve snimljene događaje i čuva sve informacije o konfiguraciji, kao što su korisnici sistema, grupe, dozvole, i definisana pravila, zone, sredstva, izveštaji, i preferencijama u Oracle bazi podataka.

ArcSight SmartAgent prikuplja i obrađuje događaje generisane od strane bezbednosnih uređaja širom preduzeća, kao što su ruteri, email logovi, anti -virus, firevall proizvodi, IDS, serveri kontrole pristupa, VPN sistemi, anti- DOS uređaji, logovi operativnih sistema, kao i drugih izvora gde Informacije o bezbednosnim pretnjama su otkriveni i objavljeni.

7. SNORT

Snort je alat otvorenog koda za mrežno otkrivanje i sprečavanje upada u sistem - NIDPS (eng. Network Intrusion Detection and Prevention System). Izvorno ga je izdala firma Sourcefire na čelu s Martinom Roeschomp od 1998. godine. Alat ima sposobnost izvođenja analize prometa u realnom vremenu i zapisivanja paketa u IP mrežama. Isprva je bio poznat pod nazivom " lagana " tehnologija otkrivanja upada, a razvio se u zrelu IPS tehnologiju punu bogatih osobina i postao skoro pa standardom za otkrivanje i sprečavanje upada u sistem. Ima gotovo 4.000.000 preuzimanja i 400.000 registrovanih korisnika. Postao je najrasprostranjenija tehnologija za sprečavanje upada u svetu.

7.1. Istorija Snort-a

Snort je besplatan alata koji je zbog toga što je pisan kao program otvorenog koda (eng. *Open source*) postao vrlo moćan i neizostavan element u bezbednosti sistema i mreža. Napisao ga je Martin Roesch 1998. godine i tada je imao samo 120 linija programskog koda. On je krenuo u izradu alata koji bi služio kao "slušalac" (eng. *Sniffer*) mrežnog saobraćaja. Ime je dobio po engleskoj reči koja u prevodu znači šmrkati, njušiti. Ovakav alat mu je bio potreban za proveru prometa na njegovom modemu i proveru grešaka. Roesch nije bio zadovoljan alatom "tcpdump", koji je na izlazu davao cele pakete u heksadecimalnom zapisu

te su se morali ručno dešifrovati. Zbog toga je napravio alat Snort, u kojem je bilo automatsko dešifrovanje i predprocesiranje.

Mesec dana od nastanka alata Martin Roesch podelio je svoj rad sa zajednicom otvorenog koda (eng. *Open Source Community*). Nakon vrlo kratkog vremena članovi ove zajednice primetili su alat Snort i dodate su nove mogućnosti koje su automatizovali praćenje saobraćaja. Martin Roesch napisao je jednostavan jezik za obradu pravila s kojima je bilo moguće raditi prepoznavanje potpisa paketa. Ovim poslednjim dodatkom privučena je velika pažnja mnogobrojnih članova koji su želeli da doprinesu daljem razvoju i usavršavanju alata. Početkom 1999. godine izašla je verzija Snort 1.0. koja nije imala predprocesore. U jesen 1999. godine Snort je već imao nekoliko novih tehnologija pa je tako dobio dekoder i modularni sistem za otkrivanje upada. Ova osnova alata Snort ostala je do danas. Sledeći nedostatak koji se rešavao je taj što se za svaku varijaciju HTTP (eng. *Hypertext Transfer Protocol*) protokola moralo raditi novo pravilo i onda se dodao predprocesor koji je rešio problem. Tokom sledećih meseci razvijali su se modularni sistemi za izlaze i za predprocesore te pravila. Poslednja samostalna verzija alata je Snort 1.7. jer od tada postaje deo alata kompanije Sourcefire koju je osnovao sam tvorac alata Snort, Martin Roesch.

Trenutno su u alat ugrađene komercijalne mogućnosti koje ga uvode u novi svet velikih poslovnih mreža. Kompanija Sourcefire je počela proizvoditi hardverska rešenja za otkrivanje upada u sistem koja u sebi imaju ugrađeni Snort te je napravila i sistem za podršku korisnicima. Iako je ovaj celi napredak velik, Snort je i dalje ostao besplatan program kojega se može skinuti s web stranice: *www.snort.org*

Preuzimanjem besplatne verzije alata Snort ne dobija se korisnička podrška i neki napredni moduli za velike poslovne mreže. Vrlo brzi razvoj Interneta donosi sve veće brzine prenosa pa tako u jednom vremenskom intervalu prođe jako puno paketa kroz mrežu i teško je te sve pakete podataka nadzirati u stvarnom vremenu. Moglo bi se dogoditi da neki paketi podataka prođu kroz mrežu, a da nisu provereni. Napredak alata Snort krenuo je od toga da se tokom nadzora mreže proveravaju samo ona pravila koja mogu doći u obzir s obzirom na vrstu upada u sistem. Ovakvim načinom se ne mora prolaziti kroz sva pravila i nepotrebno gubiti vreme na ona koja ne mogu doći u obzir jer je novih upada svakim danom sve više, a time i novih pravila.

Snort je izvorno napisan za sistem Linux, a s vremenom je počeo razvoj i za druge operativne sisteme. Danas je dostupan za sledeće platforme: Linux, OpenBSD, FreeBSD, NetBSD, Solaris (Sparc i i386), HP-UX, AIX, IRIX, MacOS i Windows.

7.2. Snort danas

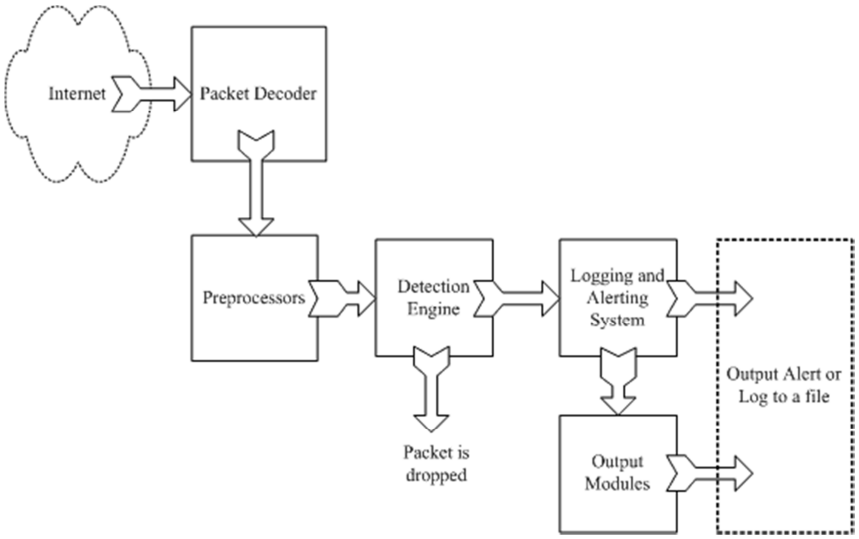
Snort može analizirati mrežni saobraćaj na IP mrežama i stvarati zapise u realnom vremenu, radi pomoću pravila, ali postoje i dodaci (eng. *plug-ins*) koji mogu pronaći nepravilnosti u jednom paketu. Pravila pomoću kojih Snort otkriva neovlašćene upade u sistem zapisuju se u tekstualnim datotekama tako da ih korisnici mogu sami prilagođavati (nadopunjavati, brisati) prema svojim potrebama. Pravila se grupišu u kategorije kako se za vreme nadziranja mreže ne bi trebalo prolaziti kroz sva pravila, nego kroz određenu kategoriju. Svaka kategorija spremljena je u svoj zasebni tekstualni dokument.

7.3. Komponente Snort - a

Snort je logički podeljen na više komponenata koje međusobno sarađuju kako bi otkrile napad u mreži. Alat se sastoji od sledećih komponenata :

- Dekoder paketa (eng. *Packet decoder*),
- Predprocesori (eng. *Preprocessors*),
- Mehanizam otkrivanja (eng. *Detection engine*),
- Sistem stvaranja zapisa i obaveštenja (eng. *Logging and alerting system*),
- Izlazni moduli (eng. *Output modules*).

Slika 12, prikazuje raspored komponenata alata Snort i tok akcija nad paketom koji se analizira. Nakon preuzimanja paketa iz mreže isti se prosleđuje dekoderu. Uzimanje paketa obavlja poseban deo programa, a nakon toga se obavlja i samo dešifrovanje. Paketi podataka nakon obrade u dekoderu dolaze u predprocesore, koji služe kako bi pakete podataka sortirali ili promenili pre nego što dođu do mehanizma za otkrivanje. Mehanizam za otkrivanje najvažniji je deo alata Snort i on mora otkriti maliciozne pakete (koje su napadači stvorili). Koristi pravila koja su učitana u interne strukture podataka. Kad dođe do poklapanja paketa s nekim pravilom, obavlja definisane radnje nad tim paketom ili se on jednostavno odbacuje. Radnja koja se može obavljati nad paketom je zapisivanje paketa ili stvaranje obaveštenja, a mogu se i obe radnje izvoditi simultano. Zavisno o tome što mehanizam za otkrivanje otkrije, sistem za kreiranje zapisa i obaveštenja obavlja zapis podataka u datoteku ili se stvara odgovarajuća obaveštenje. Izlazni moduli mogu obavljati različite radnje zavisno o tome kako se želi sačuvati izlaz iz sistema stvaranja zapisa i obaveštenja. Mogu se zapisivati podaci u datoteke, slati poruke, zapisivati u bazu podataka i slično.



Slika 12. *Komponente alata Snort*

7.3.1. Paket dekoder

Paket dekoder uzima pakete sa različitih vrsta mrežnih interfejsa i priprema pakete za predprocesiranje ili da bi bili poslani na sistem za detekciju. Interfejsi mogu biti Ethernet, SLIP, i PPP. Prvo što radi Paket dekoder je dekodiranje okvira sloja Linka podataka (kao što je Ethernet, TokenRing ili 802.11), zatim dekodiranje IP protokola i onda TCP ili UDP paketa. Kada je završio sa dekodiranjem, Snort ima sve informacije o protokolu na svim potrebnim mestima, spremno za dalje procesiranje. Na mrežnom sloju (OSI model, sloj 3) i iznad, Snort može dekodirati samo podatke koji su deo TCP/IP protokol modela. Ovo znači da ne može dekodirati podatke ili detektovati upade koji su prenošeni preko drugih mrežnih protokola, kao što je Novelov IPX/SPX ili Apple Talk. Ali za većinu mreža, to nije velik problem pošto mnogo mrežnih operativnih sistema su već migrirani na protokol TCP/IP i ne treba brinuti da će neko napadati našu mrežu sa interneta koristeći IPX protokol, jer jednostavno to nije moguće (internet je baziran na TCP/IP, ne na IPX ili nekom drugom protokolu.).

7.3.2. Predprocesori

Predprocesori su komponente ili dodaci (eng. *plugins*) koji mogu biti korišćeni sa Snortom da bi sortirali ili modifikovali pakete pre nego sistem za detekciju uradi neke

operacije da bi ispitao da li je paket od napadača. Neki predprocesori vrše i detekciju tako što traže anomalije u paket zaglavljima i generišu uzbune. Predprocesori su veoma važni za IDS pošto pripremaju pakete koji će biti analizirani. Napadači koriste različite tehnike da prevare IDS. Kao na primer, možemo kreirati pravilo koje treba da pronađe potpis “scripts/iisadmin” u HTTP paketima.

7.3.3. Sistem detekcije

Sistem za detekciju je najvažniji deo Snort-a. Njegova odgovornost jeste da otkrije pakete koji dolaze od strane napadača. Sistem za detekciju koristi Snort-ova pravila za ovu svrhu. Pravila su učitana u interne strukture podataka ili lance podataka sa kojima se svi paketi upoređuju. Ako se paket podudara sa nekim od pravila, određena funkcija nad paketom se izvrši, ili se u drugom slučaju paket odbacuje. Paket se odbacuje u smislu da se više ne zadržava u sistemu IDS-a i na njemu se ne vrši daljnja proveranja. Akcije mogu biti logovanje paketa ili generisanje uzbune, ili oboje. Sistem za detekciju je vremenski ograničen deo Snort-a. Zavisno od toga koliko je jaka mašina ili koliko je pravila definisano, moguće je da se desi da je potrebno različite količine vremena za različite pakete. Ako je mrežni saobraćaj velik, i ako Snort radi u NIDS modu, moguće je da neki paketi otpadnu. Opterećenje sistema za detekciju zavisi od sledećih faktora:

- Koliko je pravila definisano;
- Jačina mašine na kojoj je Snort;
- Brzina interne sabirnice koja se koristi u mašini na kojoj je Snort;
- Koliki je saobraćaj na mreži.

7.3.4. Sistem za logovanje i uzbune

Zavisno od toga šta sistem za detekciju nađe unutar paketa, paket se loguje ili se generiše uzbuna. Logovi se čuvaju u jednostavnim tekst fajlovima, tcp-dump vrsti fajlova ili u nekoj drugoj formi. Svi log fajlovi su sačuvani u /var/log odnosno c:\snort\log kao podrazumevanim lokacijama. Možemo koristiti **-l** opciju komandne linije da bi modifikovali lokaciju gde se generišu logovi i uzbune. Mnoge dodatne opcije logovanja će biti modifikovane baš korišćenjem ovog parametra pri pozivu Snort-a.

Sistem za uzbune kod Snort-a nas može obavestiti kada mrežni promet odnosno neki paket se podudara sa kriterijumima definisanim nekim pravilom. U sledećem primeru možemo videti kako nam može reći sistem za logovanje i uzbune da se dogodila povreda sigurnosti:

01/20-22:34:35.218093 [**] [1:469:1] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 192.168.1.68 ->172.16.34.18

Na ovaj način nam Snort govori da neko napolju koristi Nmap (alat za skeniranje mreže), da ping-a sisteme što je siguran znak da sledi skeniranje portova. Prethodna uzbuna takođe sadrži:

- Datum i vreme (zajedno sa sekundama i mikrosekundama);
- SID (Snort ID), identifikator koji nam govori koje pravilo je u pitanju. Ovo je napisano u formatu[*sig_generator:sig_id:sig_revision*] gde je:
    - **sig_generator** nam govori koji deo Snort-a je generisao uzbunu
    - **sig_id** je Snort-ov signature ID, koje nam govori koje pravilo je pokrenuto
    - **sig_revision** je reviziski broj tog pravila
- Kratka tekstualna poruka
- Klasifikacija i prioritet napada
- Protokol paketa koji je pokrenuo pravilo
- IP adrese od pošaljioca i primaoca paketa

Sistem za logovanje ne generiše nikakve alarme. On nečujno loguje sve informacije koje se nalaze u paketu koji se koristi za napad. Nekad je poželjno da se napad loguje bez generisanja uzbune.

7.3.5. Izlazni moduli

Izlazni moduli ili izlazni dodaci (eng. *Output plugins*) mogu raditi različite operacije zavisno od toga kako želimo da sačuvamo izlaz generisan sistemom logovanja i uzbuna Snort-a. U osnovi ovi moduli kontrolišu vrstu izlaza generisanih sistemom logovanja i uzbuna. Zavisno od konfiguracije, moduli izlaza mogu raditi sledeće stvari:

- Jednostavno logovanje na /var/log/snort/alerts (c:\snort\log\alerts.ids) fajl ili neki drugi fajl,
- Slati poruke na sistemski log operativnog sistema,
- Logovanje na bazu podataka kao što je MySQL ili Oracle,
- Mogu generisati eXtensible Markup Language (XML) izlaz,

- Modifikovati konfiguraciju na router-ima i firewall-ima,
- Mogu slati Server Message Block (SMB) poruke Microsoft Windows bazirane mašine.

Od verzije 1.6, Snort koristi izlazne module, da bi zapisivao podatke iz sistema za logovanje i uzbune Snort-a. Snort-ovi sistemi za logovanje i uzbune zapravo ne zapisivaju podatke. Oni šalju podatke na odgovarajući izlazni modul, koji vrši formatiranje i pisanje logova. Moguče je napisati vlastiti izlazni modul za specifično logovanje i uzbunivanje. Snort može biti pokrenut na više načina:

- Kao deamon ili serivs, koji konstantno radi kao pozadinski proces koji vrši pregledavanje mreže u potrazi za uljezima,
- Sa komandne linije kao snifer paketa,
- Kao naprednija opcija paket sniffer-a koji može da uhvati podatke, i da ih uporedi sa određenim pravilom.

Neki Snort-ovi izlazni moduli su bolje prilagođeni radu u deamon načinu rada dok su neki bolje prilagođeni paket sniffer načinu rada. Svaki izlazni modul se poziva malo drugačijom sintaksom zavisno od toga da li je pozvan sa komandne linije ili konfiguraciskog fajla. Uobičajeno izlazni moduli šalju svoje podatke na /var/log/snort (Linux) odnosno c:\snort\log (Windows), i ako različiti direktorijumi mogu biti postavljeni zavisno od potrebe.

Različit direktorijumi za logovanje može biti specifiran korišćenjem različitog parametra **-l** u komandnoj liniji. Svi logovi i uzbune u ASCI formatu de biti prosleđeni u direktorijum specifikovan sa **-l** parametrom, a ako nije postavljen taj parametar onda će biti prosleđeni u standarnu lokaciju (/var/log/snort, c:\snort\log). Bilo koja opcija koje se specifikuje na komandnoj liniji će transparentno preskočiti postavku za istu opciju u okviru snort.conf fajla.

7.4. Tipovi Izlaznih modula za uzbune

Snort poseduje par različitih načina da generiše uzbune u zavisnosti koliko detalja želimo da vidimo u našim uzbunama i gde želimo da pošaljemo te uzbune. Počet ćemo sa nekim jednostavnijim opcijama i na kraju završiti sa nešto kompleksnijim. Konfiguracija izlaznih modula se radi u okviru snort.conf fajla (da ponovimo da se dati fajl nalazi na lokacijama /etc/snort.conf za Linux i c:\snort\etc\snort.conf za Windows). U tom fajlu opcije za konfiguraciju izlaznih modula su pod „Step #3: Configure output plug-ins“

A nabrojat ćemo neke osnovne izlazne module za uzbune i njihove osobine:

- **alert_fast**
  Ovo je često najbolji izbor za odabir izlaznog modula. Alert_fast modul će ispisivati uzbunu u jendoj liniji bez obzira koji mu fajl specifikujemo. Kao što mu ime govori, brz je zato što ne troši cikluse procesora na konvertovanje zaglavlja paketa u čoveku čitljiv format. Sintaksa koju koristimo u konfiguraciskom fajlu da bi rekli Snort da koristi ovaj modul je **output alert_fast: snort.log** Sintaksa koju koristimo kad pozivamo Snort iz komandne linije je **-A** fast Ovaj oblik izlaza uzbune nam govori osnovne informacije o napadu kao što su tip upada, klasifikacija napada, te izvorišna i odredišna adresa i port kao i vreme upada. Ne mnogo drugih informacija se snimi koristeći ovaj modul upravo zbog toga je tako brz.

- **alert_full**
  Ovaj oblik izlaza uzbune nam daje znatno više informacija o napadu, ali ima i svoju cenu što je smanjenja perfomansi i ne bi se trebalo koristiti nigde osim u slučaju slabo opterećenih mreža. Ako koristimo ovu opciju na mreži gde ima mnogo saobraćaja, Snort će izgubiti neke pakete. I ne samo to, biče ispisani gigabajti informacija na HDD, smanjivat će se slobodni prostor i doći će do I/O zagušenja na sistemu. Da bi koristili ovaj način preuzimanja uzbuna, potrebno je editovati snort.conf file sa sledećom linijom: **output alert_full: alert.full**. Sintaksa koju koristimo kad pozivamo Snort iz komandne linije je **–A** full.

- **alert_syslog**
  Slanje uzbuna na log fajl je odlično, ali metode koje su prethodno bile napomenute imaju manjak fleksibilnosti. Neretko se dešava da informacije o upadu želimo da imamo na istom mestu gde i ostale informacije o sistemu. U suštini, ovaj oblik izlaza je vrlo sličan već napomenutom alert_fast izlaznom modulu osim što ima dodatne mogućnosti slanja uzbune u sistemski log. Zanimljiv je uticaj ovog modula na sigurnost u slučaju da koristimo daljinski log server (dosta česta implementacija u svetu Linux-a), gde dajemo napadaču manje mesta za skrivanje tragova jer mora da, pored jednog servera u kojeg je ciljno upao, komprimutuje i log server na kojem je spašen tok aktivnosti upada. I ovaj oblik specifikujemo u okviru konfiguracionog snort.conf fajla: alert_syslog: <log_sistem> <prioritet>, gde <log_sistem> najčešće može biti LOG_AUTH, a prioritet LOG_ALERT. Iz komandne linije nije moguće preusmeriti izlaz uzbuna u sistemski log.

- **alert_CSV**
  CSV znači Comma Separated Values, je gotovo univerzalni format za unošenje podataka u baze podataka ili spreadsheets. Ako želimo da procesiramo Snortove uzbune u bazi podataka koja nije podržana od Snort-a, kao MS Access, ili da ih unosimo u spreadsheet kao što je Excel, onda je alert_CSV izlazni modul rešenje.

7.4.1. Izlazni moduli za logovanje

Moduli logovanja se koriste za logovanje informacija paketa. Moduli za logovanje su obično pozvani od strane modula za uzbune u cilju da se loguje informacija paketa koji je povezan sa određenom uzbunom, ali takođe mogu biti pozvani od strane nekih predprocesora,

ili pozvani nezavisno sa komandne linije. Snort hvata mrežni promet i aplicira predprocesor na promet u cilju da normalizuje promet u format koji je prepoznatljiv sistemu za detekciju. Ako predprocesor odluči koji promet je zanimljiv, on može pozvati izlazni modul da pošalje uzbunu, ili da loguje paket, ili da uradi oboje (zavisno od toga kako je modul napisan). Snort onda upoređuje mrežni transport sa listom određenih pravila. Ako se paket podudara sa određenim pravilom, onda Snort generiše uzbunu koristeći modul uzbune (koji je specifikovan u snort.conf ili u komandnoj liniji). Kada se uzbuna generiše, Snort takođe zove određeni modul za logovanje da bi logovao detalje paketa koji su zadovoljili kriterijume određenih pravila. I sada kada vidimo uzbune u svojim logovima, možemo iči nazad i ispitati informacije u paketima koje su odgovorne za generisanje uzbune.

Možemo izabrati format zavisno od brzine koja nam je potrebna, da olakšamo postprocesiranje, da mašina ima bolju preglednost, ili ljudi da imaju bolju ćitljivost. U zavisnosti od toga imamo tri principijalna osnovna oblika izlaza modula za logovanje:

- **Tcpdump binarno logovanje**
  Tcpdump binarni je vrhunac u brzom i fleksibilnom post-procesiranju. Ova gotovo univerzalna metoda ne zahteva mnogo procesorske snage za konverziju u tekst. Kada se koristi tcpdump stil logovanja, Snort lako radi sa veoma opterčenim mrežama bez gubitka paketa. Binarni podaci su dobri za mašine, ali su nečitljivi za ljude.

- **ASCII logovanje**
  Ako imamo poteškoće sa interpretacijom jezika koji se sastoji samo od nula i jedinica, onda trebamo razmotriti ASCII opcije logovanja. Potrebno je više procesorske snage nego kod binarnog logovanja, ali su zato razultati dosta čitljiviji za ljude.

- **Logovanje u bazu podataka**
  Logovanje direktno u bazu podataka je jedna od najkorisnijih opcija za reportiranje Snort podataka, i za deljenje sa drugima. Iako tabele baza podataka nisu lake za čitanje bez softverskog klienta, softver klienata ima mnogo, i programa da bi se preveli unosi baze podataka u slikovite web stranice.

- **Unificirano logovanje**
  Ostali oblici logovanja imaju jedan veliki problem koji se ogleda u tome da je kompleksno podešavati bazu, a i baza je neretko na istom računaru instalirana gde i sam senzor pa uzima nepotrebno dragocene cikluse procesora. Jedno od rešenja za to je korišćenje unificiranog izlaza logovanja koje se ne trudi da prebaci izlaz direktno iz Snort-a u čoveku čitljiv format i samim tim smanjuje opterečenje za procesor i ubrzava rad. Prebacivanje u čoveku čitljiv format vrši se preko spoljašnih delova softvera (**Barnyard**) i u kombinaciji sa ovim načinom logovanja čine jedan jednostavan i vrlo efikasan metod davanja izlaza iz Snort-a. Snort u budučnosti će

sigurno forsirati ovaj metod logovanja jer je izuzetno jednostavan i nema efekta na perfomanse.

7.5. Način rada i mogućnosti

Načini rada alata Snort su :

- **Sniffer** - najjednostavniji način rada koji se pokreće na početku kako bi se upoznali s alatom . U ovom načinu rada alat Snort prati promet paketa podataka na mreži i obavlja rezultat u komandnoj liniji . Pokretanje ovog načina rada obavlja se naredbom " *Snort - v* " , a na ekranu se dobije ispis prometa paketa u mreži . Naredbe koje se još koriste u ovom načinu su :
  " *Snort - w* " - ispisuje spisak mrežnih kartica u mreži koju alat Snort nadzire,
  " *Snort – v-i* " - pokretanje alata Snort , * označava mrežnu karticu s kojom se nadziru paketi.

- **Packet Logger** - način rada u kojem se ispituju mogućnosti stvaranja zapisa paketa alatom Snort sledećom naredbom " snort - dev - l Snortpath \ log ". Bolja mogućnost od ove bila bi zapisati pakete u binarnu datoteku jer se s njima može brže raditi , a naredba za provođenje toga je " snort - bl Snortpath \ log ". Na računaru se mogu pronaći , otvoriti i proveriti datoteke koje je alat Snort stvorio. Kad se paketi zapišu u binarnu datoteku, mogu se čitati sa bilo kojim alatom koji podržava " tcpdump " format.

- **Mrežni IDS** je najsloženiji način rada i za njegovo ispravno funkcionisanje potrebno je postaviti neke opcije : mrežne postavke (eng. *Network Settings*), postavke pravila (eng. *Rules Settings*), postavke izlaza (eng. *Output Settings*) i podešavanja uključivanja (eng. *Include Settings*). Mrežne postavke su mesto gde se određuje koji deo IP adrese treba nadzirati. Osnovna postavka konfiguracije datoteke obavlja se naredbom " varHOME - NET any " pomoću koje alat Snort nadzire celu mrežu na koju je priključen. U podešavanjima pravila alatu Snort se pokazuje lokacija gde se nalazi datoteka koja sadrži pravila (naredba " varRULE_PATH SnortPath \ rules "). Postavke izlaza služe kako bi se odredilo kako će izlazne informacije biti predstavljene, a to su zapisivanje s naredbom " output logdir "(mesto gde je instaliran Snort, npr. c\snort\log) i obaveštenje naredbom " output alert_fast: alert.ids ".

7.6. Šta je senzor Snort-a i gde postaviti senzor

Postavlja se pitanje koje je najbolje mesto da se postavi senzor za jedan mrežni sistem za detekciju kao što je Snort. Snort inače radi na nivou jednog mrežnog interfejsa, mrežne kartice (engl. NIC - *Network Interface Card*) koja mora biti postavljena u zajedničkom načinu rada, tako da pored mrežnog saobračaja koje šalje na ciljne lokacije, isti taj saobraćaj prosleđuje na Snort. Ta mrežna kartica u tom trenutku može da se nazove senzor Snort - a . Mrežna kartica inače prosleđuje sav saobraćaj na Snort ali daljim konfugirisanjem Snort - a

moguće je ograničiti saobraćaj na koji naš mrežno bazirani sistem za detekciju treba da obrati pažnju. Konkretno to se radi postavljanjem varijable var HOME_NET u datoteci snort.conf.

7.7. Lokacija za snort

Odgovor na pitanje gde postaviti senzor za Snort zavisi od zahteva određene organizacije, i šta želi da detektuje. Jedan način određivanja pozicije Snort - a jeste pozicija firewall - a, odnosno, postavljanje Snort senzora ispred ili firewall - a. Postavljanje IDS - a ispred firewall-a će omogućiti detekciju svih napada, bez obzira da li bi mogli proći kroz firewall. Postavljanjem IDS-a ispred generiše se više logova nego postavljanjem iza firewall-a, zato što do IDS-a dolazi nefiltriran saobraćaj, odnosno saobraćaj koji nije prošao preko firewall-a. Ako želimo nadzor prometa koji je prošao kroz firewall onda IDS treba postaviti iza istog. U mogučnosti, najbolje je postaviti IDS senzore ispred i iza firewall-a. U ovom slučaju imamo uvid u čitav saobraćaj upućen prema mreži i saobraćaj koji je prošao kroz firewall.

Pozicija IDS-a u odnosu na firewall	Prednosti	Mane
Ispred firewall-a	Viši nivo svesnosti sa kojim upadima se suočavamo	Ako firewall ima uključen NAT, praćenje saobraćaja je otežano; veče opterećenje IDS-a
Iza firewall-a	Samo ono što prođe kroz firewall, će biti pregledavano; manje opterećenje IDS-a	Niži nivo svesnoti o okruženju i o mogućim upadima.

Tabela. 1. *Pozicija IDS-a u odnosu na firewall*

7.8. Optimizacija Snort-a

7.8.1. Smanjivanje pogrešnih upozorenja

Alat Snort se može konfigurisati kako bi se smanjio broj pogrešnih upozorenja. jedan od načina, a koji je ujedno i najjednostavniji, je isključiti nepotrebna pravila. Svaki korisnik može dodavati nova pravila i brisati ona koja su za njega nepotrebna. Uz alat Snort dolazi jako puno pravila, a njihov broj se povećava dodavanjem sopstvenih. Najčešće je otprilike samo pola pravila dovoljno za funkcionisanje sistema. Najbolje mesto gde bi se trebalo početi

sa brisanjem pravila je kad se pogleda mapa sa rasporedom uređaja u mreži. U velikim mrežama postoji puno mrežnih dijagrama pomoću kojih se može videti gde se u mreži nalaze određene mrežne tehnologije, hardver, operativni sistemi i aplikacije.

Najbrži način da se vidi da li postoje neželjeni sistemi ili računari u mreži je skeniranje mreže. To daje prikaz isti onakav kakav ima napadač i može biti vrlo korisno, jer daje na uvid kako napadač namerava upasti u sistem. Postoje brojni alati za to, a jedan od njih je NMAP (eng. *Network Mapper*) koji daje realnu sliku kako mrežu vidi napadač.

Još jedan način kako se može dobiti uvid u sisteme u mreži i njihove propuste je alat za ispitivanje bezbednosti mreže. Primer takvog alata je OpenVAS koji može dati listu otvorenih uređaja na mreži i detalje koje aplikacije rade s tim uređajima.

Ako se koristi operativni sistem Windows 7 u mreži onda se mogu isključiti sva pravila za druge operativne sisteme tako da se ispred njih stavi znak # . Veliko je gubljenje vremena da alat Snort pretražuje sva pravila ako nema potrebe za tim.

7.8.2. Stvaranje novih pravila

Ponekad je potrebno napraviti nova pravila, a situacije koje to zahtevaju mogu biti neke od sledećih :

- Primećivanje nekog nestandardnog ponašanja u mreži kao što može biti prenos velike količine podataka bez nekog vidljivog razloga. U tom slučaju treba te podatke analizirati, pokupiti uzorke prometa sa alatom Snort te, ako je to potrebno, napraviti odgovarajuće pravilo.

- Napravljen je novi oblik napada na mrežu i pravila koja su zapisana u datoteci alata Snort ne sadrže ovaj novi napad te se napad ne može registrovati. Potrebno je stvoriti novo pravilo, barem privremeno dok ne izađe službena verizija koja sadrži ekvivalentno pravilo. Ovaj slučaj nije čest jer se retko dogodi da se mora samostalno stvoriti pravilo pre nego je izdata službena verzija.

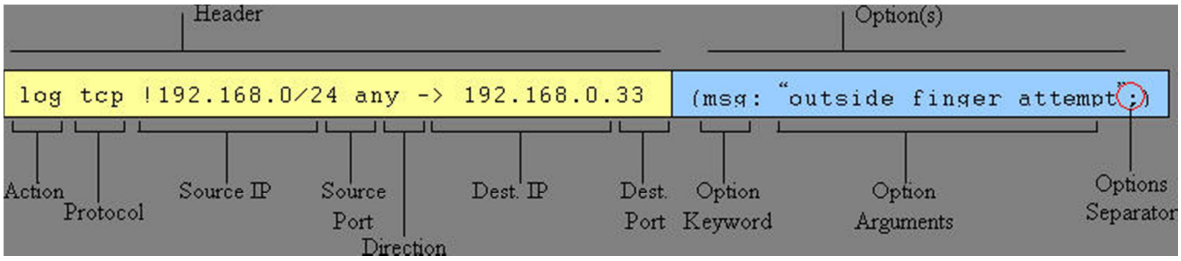
Alat Snort ima više tipova pravila, a to su:

- Pravila protokola (jer alat Snort može otkriti protokol IP u mreži),
- Pravila vezana uz uređaje u računarskoj mreži jer se pretražuje određeni uređaj. U ovim pravilima podaci u paketu nisu važni, nego samo protokol, odnosno broj priključka (eng. *port*). S ovom vrstom pravila treba biti pažljiv jer se može preopteretiti alat Snort kad se koristi sa uobičajenim priključkom.
- Aplikacijska pravila se koriste zbog toga što se ponekad neobično ponašanje dogodi u aplikaciji ili protokolu i to onda treba otkriti.

Pravila alata Snort imaju osnovni format koji se može proširivati za neke određene potrebe, a izgleda ovako:

```
<snort action> <protocol> <src IP> <src PORT> <direction> <dst
IP> <dst port> (msg:"Tell the user what I'm tracking"; <optional
classtype>;<optional snort ID (sid)>; <optional revision (rev)
number>;)
```

i može se videti na slici u nastavku



Slika 13. *Zaglavlje i opcije pravila alata Snort*

Ključne reči iz osnovnog formata pravila znače :

- **Snort action** - radnje koje preduzima alat Snort. Ovde može stajati jedna od tri ključne reči : Alert (obaveštenje) šalje obaveštenje na osnovu potpisa, Log (zapis) ne stvara se obaveštenje nego se samo zapisuje ovaj događaj u određenu datoteku te Pass (prolaz) govori sistemu za otkrivanje da se propuste oni paketi koji zadovoljavaju uslove o potpisu.

- **Protocol** - u ovom delu se označava alatu Snort koji protokol se treba nadzirati. Neki od najčešćih protokola su TCP (eng. *Transmission Control Protocol*) , UDP (eng. *User Datagram Protocol*) i ICMP (eng. *Internet Control Message Protocol*) ili može biti uopšteno IP protokol.

- **Source IP** - grupa IP adresa iz kojih alat Snort može videti od kuda je uspostavljena veza.

- **Direction** - pokazuje da li je veza uspostavljena iz izvorne ili odredišne IP adrese.

- **Destination port** - odredišni priključak prometa : 80 za HTTP , 21 za FTP (eng. *File Transfer Protocol*) i 23 za telnet veze.

- **Message** - polje gde se uz obaveštenja mogu ostaviti komentari, a informacija se prikaže na mestu za obaveštenja.
- **Class type** - oznaka prioriteta. Ako postoji više pravila koji se slažu s delom mrežnog saobraćaja, ovaj parametar pomaže da se definiše koje će se pravilo koristiti.

- **Snort Identification (SID) Number** - jedinstveni broj koji je dodeljen određenom pravilu. Pri stvaranju vlastitog pravila dodeljuje se broj SID koji je iznad 1000 000.

Ovime se stvara vidljiva razlika između pravila alata Snort i pravila koja je korisnik sam stvorio.

- **Revision number** - opcionalni deo, a koristan je ako se stvara više verzija istog pravila te ako se prate promene pravila za celi sistem IDS.

7.9. Izbor operativnog sistema za Snort

Kada se počinje građevisnke projekat, dobro je znati alate koji su potrebni i kakvi su temelji. Isto se odnosi i na Snort (naziv odmilja " *The Pig* " zbog maskote svinje). ovaj odeljak se odnosi i na Linux i Windows platforme za Snort, hardverske zahteve, i na uputstva za dobar odabir mreže. Snort se distribuira i kao source code (kojeg korisnici mogu kompajlirati) i u vidu binarnih fajlova koji su kompajlirani pre svega za Linux, MacOS, FreeBSD, ali već neko vreme i na Windows-u. Ne postoji bitni problem pokretanja Snort-a na verzije posle Windows 2000 operativnog sistema , što još jednom govori o sve većoj raspostranjenost open source rešenja na sve više operativnih sistema. Već kompajlirani paketi (za Linux i Winodows samo) su dostupni sa sajta www.snort.org. Linux binarni paketi se distribuiraju u RPM format pakovanja, koji se često koriste sa RedHat Linux i Fedora Project (zajednica podržana od RedHat Linux), i Mandrake Linux. Windows binarna verzija su jednostavni EXE fajlovi. Za Linux, kompajliranje iz source koda je preporučen metod instalacije. Za Windows, binarni je najbolji način instalacije. Snort radi na mnogobrojnim operativnim sistemima, svaki sa svojim prednostima i manama. Tabela daje kratak uvid u neke od ovih sistema kao i prednosti i mane.

Operativni Sistemi	Prednosti	Mane
Linux	Fleksibilnost, perfomanse, cena.	Previše fleksibilno (mnogo toga traži podešavanje)
Windows	Prijateljski orijentisan prema Korisnicima.	Slab menadžment, mogući problemi s perfomansama
FreeBSD	Stabilnost	Ograničen izbor hardvera
OpenBSD	Sigurnost	Ograničen izbor hardvera
Sun Solaris	Sklabilnost	Skupoceni Hardver
MacOS X	Prijateljski orijentisan prema korisnicima.	Ograničen izbor hardvera

***Tabela. 2**. Prednosti i mane*

U našim prilikama, izbor operativnog sistema može se najčešće suziti između Linux i Windows operativnih sistema. Preporučljivo je koristiti Linux distribuciju koja koristi kernel

verziju 2.4 ili novije verzije (kernel je srce Linux operativnog sistema). Linux Kernel verzija 2.4 pruža bolje perfomanse i bezbednost nego starije verzije. Linux Kernel 2.6, pruža bolju nadogradivost, bolje I/O perfomanse, i ima podršku za nove procesore kao što su AMD 64-bit Opteron.

Prednost Linux-a kao Snort IDS senzora može biti sumirano u četiri stvari: perfomanse, podesivost, patch-iranje i cena. Linux-ova TCP/IP podrška (posebno u verziji 2.4 i novijim verzijama kernela) odlično radi posao sa velikim količinama podataka koje prolaze kroz Snort IDS mrežnu karticu. Za razliku od Linux-a, standardna Windows instalacija sadrži u sebi mnoge nepotrebne stvari za funkcionisanje Snort IDS-a. Neke od tih stvari se mogu ukloniti dok recimo GUI (grafički interfejs) ne može, i na kraju krajeva šta čini Windows Windows-om. Sa Linux-om je moguće ne instalirati u potpunosti GUI, i druge aplikacije koje nisu potrebne za Snort i tako dati puno više prostora za Snort IDS. Linux za razliku od Windows-a je besplatan, kao i Snort. Velikim kompanijama je poterbno više Snort senzora, što znači i upotrebu više operativnih sistema. Korišćenjem besplatnih operativnih sistema može se uštediti značajna količina novca.

Snort IDS radi na Microsoft Windows 2000 Professional Serveru, XP Professional, Windows 2003 Serveru Winodws 7. Zbog načina na koji Snort koristi mrežnu karticu (samo osluškuje i ne pravi nikakve konekcije), Professional versions 2000 i XP rade dobro (i može se uštedeti novac na serverskom licencsiranju). Snort može da radi na Windows 95, 98, ME, NT, i XP Home, ali nije preporučljivo. Ove verzije nemaju podršku za najnoviji hardver, nisu stabilne, i nemaju dobru zaštitu kao novi sistemi. I pored toga, Windows 95, 98, i NT nemaju više podrške u vidu zakrpi. Korišćenje Windows sistema kao platforme za Snort je prednost ako je u čitavoj organizaciji Windows dominantan operativni sistem. Još jedna stvar u kojem Windows može biti pogodan je postojanje solidne podrške za različite proizvođače hardvera. Još jedna prednost Windows-a jeste lakoća korišćenja upotrebom GUI-a. Kod Windows-a se koriste point-and-click aplikacije što u mnogome pojednostavljuje stvari za neiskusne Snort IDS korisnike, dok kod Linux-a, se mora koristiti liniski interfejs ili se treba instalirati GUI naknadno. Ako se misli samo ne tehničke elemente onda Snort IDS na Linux je pravo rešenje. Linux sistemi su brzi, pouzdani i lako se konfigurišu. Ali, ipak nije sve u tehničkim stvarima. Takođe, i poznavanje i iskustvo stečeno sa radom sa određenim sistemom kao Windows, stavlja Windows na prvo mesto. Ako odlučimo da Snort vratimo na Windows platformi, takvo rešenje ne predstavlja funkcionalno manju efikasnost kao na Linux platformi Snort ima iste opcije na obe platforme.

7.10. Primena Snort-a u organizacijama

U organizacijama je potrebna zaštita od upada zbog zaštite onoga čime se organizacija bavi, privatnih podataka, intelektualnog vlasništva, identiteta zaposlenog i drugih stvari. Ponekad sami zaposleni učestvuju u zločinima, a da toga nisu ni svesni. Korišćenjem određenih alata za slušanje ili ilegalno preuzimanje muzike i filmova na računare na poslu mogu dovesti do ugrožavanja sigurnosti računarskih sistema.

Mnogi alati koji služe za deljenje datoteka, kao što su Kazaa i Gnutella , koriste se za deljenje sadržaja koji su zabranjeni (jer se povezuju sa drugim računarima u svetu i tako čine savezne zločine). Svaka organizacija ima različite razloge za instalaciju alata Snort i različite ciljeve u nadziranju svojih mreža. Multimedijalni saobraćaj postaje značajan problem u mnogim mrežama gde korisnici preuzimaju takav sadržaj na svoje računare. Postoji primer kad je škola za umetnost i dizajn Ringling uspela smanjiti mrežni saobraćaj za 80% koristeći alat Snort za identifikaciju i sprečavanje P2P (eng. *Peer-to-peer*) aktivnosti. Mnoge škole i fakulteti želeli bi sačuvati svoju propusnost na mreži u svrhu edukacije i isto tako ukinuti narušavanje autorskih prava zbog P2P aktivnosti. Osim potrošnje propusnosti, razvijeni su i novi napadi koji su jedinstveni za multimedijalni sadržaj (npr. Trojan.Brisv.A) . Broj napada na multimediju se povećava povećavanjem broja mediskih formata i protokola koji su razvijeni i uvedeni.

Škole mogu koristiti Snort kao obrazovni alat i kao alat za otkrivanje upada u sistem. Juta Vallei State College ima uključen alat Snort u preddiplomski mrežni program kao deo jeftinog laboratorija koji radi s ovim alatom otvorenog koda. Koledž je razvio laboratorijsku vežbu u kojoj koriste alat Snort kako bi mogli uočiti mesto upada. Snort može biti integrisan i u druge delove IT (eng. *Information technology*) nastavnog plana i programa. Jedan dodatak koji bi se mogao uklopiti u nastavni plan i program je alat Barniard. Radi se o alatki otvorenog koda koji preuzima odgovornost za punjenje baze podataka sa izlazom iz alata Snort. Usmeravanje izlaza alata Snort pruža stvarne prilike za pisanje dodataka i programa za obaveštavanje. Veliki broj lažno pozitivnih obaveštenja koje stvara alat Snort dovode do toga da se puno podataka mora zapisati u bazu podataka. Takođe, jedna od obrazovnih mogućnosti za integrisanje alata Snort u nastavni plan i program je pisanje izlaznih datoteka.

Za pisanje srednje - složenog izlaznog dodatka trebalo bi programeru s umerenim veštinama dve do četiri nedelje. Razvijanje izlaznih dodataka je zadatak primerene veličine za grupu poslediplomskih studenata kako bi demonstrirali svoje veštine u pisanju programa i razumevanju mrežnog otkrivanja upada.

7.11. Poređenje sa drugim IDS / IPS alatima

Snort je najpopularniji i najbolje ocenjeni alat otvorenog koda koji se koristi kao sistem za mrežno otkrivanje upada. Odlično je oružje u borbi protiv mrežnih upada pa može konkurisati komercijalnim IDS alatima. Navedeni alat je alat otvorenog koda za mrežno otkrivanje i sprečavanje upada u sistem i koristi jezik koji se zasniva na pravilima. Snort kombinuje prednosti rada sa potpisima i rada s nepravilnostima. Ima i jako dobru korisničku podršku, a to je često glavni razlog za izbor tog alata. Snort se razvija već više od deset godina i postoji velika količina dostupnih dokumenata koji pomažu početnicima u radu s alatom. Danas može raditi na svim najpopularnijim operativnim sistemima.

Licence za hardver i operativni sistem kompanijama stvaraju veliki trošak te onda traže jeftinije programsko rešenje za mrežnu bezbednost. Rešenje ovog problema može biti alat koji radi na svim operativnim sistemima i koji je besplatan. Snort se može koristiti i kao skriveni alat jer se može postaviti tako da bude skriven na mreži, te na taj način bude zaštićen od napadača. Kada je ovako skriven može nastaviti s radom i zapisivati upade ako mreža postane ugrožena. Postavljanjem alata Snort u ovakav mod rada, da mu se iz mreže ne može pristupiti, omogućava da se zapisivanje nastavi čak i tokom upada. Ovakvim načinom pomaže se u pripremi mreže od budućih napada takve vrste što uveliko povećava njegovu praktičnu vrednost.

Drugi alat za mrežni IDS je alat Bro, koji je stvorio Vern Pakon u institutu *Lavrence Berklei National Labaratory and International Computer Science Institute*. Radi se o vrlo naprednom alatu koji se koristi kao mrežni IDS. Bro je alat otvorenog koda, koji je napisan za platformu UNIKS i služi za mrežno otkrivanje upada. Mogućnosti navedenog alata slične su onima koje ima alat Snort. Bro nadzire mrežni saobraćaj koristeći skup pravila otvorenog koda kako bi mogao tražiti sumnjivi promet. Od kad je alat Bro dobio mogućnost da ga se može samostalno prilagođavati, administratori mogu određivati i proveravati mogućnosti koje će se postaviti na mrežu. Otkrivanjem novih oblika mrežnih napada, pravila se mogu menjati i dodavati tako da imaju sadržane potpise tih novih napada (zbog čega se mreža može bolje nadzirati). Kao i alat Snort, Bro može otkriti nepravilnosti u mrežnom prometu i te nepravilnosti zapisati u za to određene datoteke. Naknadno se zatim istražuju uočene nepravilnosti te formira obaveštenje (npr e-mail poruka administratoru sistema).

Platforma alata Bro je sistem otvorenog koda i pokreće se na vrlo stabilnom operativnom sistemu UNIKS. Prema tome, alat Bro nije namenjen kao rešenje za preduzeća koja ne koriste operativni sistem UNIKS (jer zahteva posebna znanja o tom sistemu). Ovo može biti

ograničavajući faktor u donošenju odluka o njegovom korišćenju u preduzećima. Osnovna prednost alata su velika sposobnost da u rukama stručnjaka postane alat za zaštitu mreža u stvarnom svetu. Bro može pružiti izvrstan mrežni IDS sistem za preduzeća za jako nisku cenu. Dodatna prednost mu je i što je prilagođen tako da može koristiti fajlove sa potpisima alata Snort. Bro otkriva upade tako da poredi mrežni saobraćaj sa prilagodljivim skupom pravila za opis događaja koji se smatraju problematičnima, baš kao i Snort. Ako se s alatom Bro otkrije nekakvo odstupanje, može se izvoditi zapis toga u datoteku, stvarati obaveštenje operatoru u realnom vremenu ili pokrenuti izvođenje određene naredbe. Bro bi mogao biti snažniji alat od Snort-a za mrežni IDS zbog toga što je napravljen tako da je više mrežno orijentisan te može otkriti i prijaviti sve strane aktivnosti na mreži.

Alati Snort i Bro napravljeni su kako bi se u potpunosti integrisali u mrežu, potpuno iskorišćavanje mogućnosti koje nude ova dva alata zavisi o stručnosti operatera (administratora) u organizaicijama koji rade na njima i koji ih uključuju (i naravno kasnije održavaju) u sistem. Oba alata su otvorenog koda pa je kod na raspolaganju stručnjacima za korišćenje i dalji razvoj. U kombinaciji, ova dva alata pružaju drugi sloj zaštite koji je potreban za ispravno štićenje od slabosti sistema.

AIDE (eng. *Advanced Intrusion Detection Envirnonment*) je mrežni IDS alat koji radi na operativnom sistemu UNIKS. Glavni nedostatak ovog alata je manjak dokumentacije za podršku korisnicima, još uvek je u razvoju, ali ima potencijala da postane značajan alat. Ovaj alat bi trebao biti prvi izbor ako se traži mrežni IDS za operativni sistem UNIKS.

Shadow IDS je IDS alat otvorenog koda , nastao kao deo projekta koji je pokrenuo cider (eng. *Cooperative Intrusion Detection Environment*). Naval Surface Varfare Center je okosnica ovog projekta zajedno sa nekoliko drugih vladinih agencija. Ovaj projekat se razvija od 1998. godine te ima dobru dokumentaciju za podršku svojim korisnicima. Shadow koristi biblioteke "tcpdump" i "libcap". Njegova prednost u odnosu na alat Snort je ta što omogućava administratoru da postavi senzore kroz celu mrežu, tj može se koristiti kao distribuirani IDS.

8. Praktičan primer

8.1. Postavljanje snort-a u aktivan režim rada

Instalacija je urađena na virtuelnoj mašini. Korišćeno je sledeće:

- Virtuelna mašina VmWare Workstation sa instaliranim guest OS WinXP SP1
- WinPcap
- Snort

Instalacija Snorta je urađena na virtelnoj mašini radi portabilnosti i lakše manipulacije sa različitim stanjima instalcije. Na virtuelnoj mašini je instaliran i WinPcap (drajver koji omogučuje da mrežni interfejs radi u zajedničkom modu. Nakon instalacije prvo što je urađeno je podešavanje fajla snort.conf i najvažnije je bilo postaviti stavku var HOME_NET. Za Var EXTERNAL_NET dobro je ostaviti na any.

Snort može da radi u tri načina:

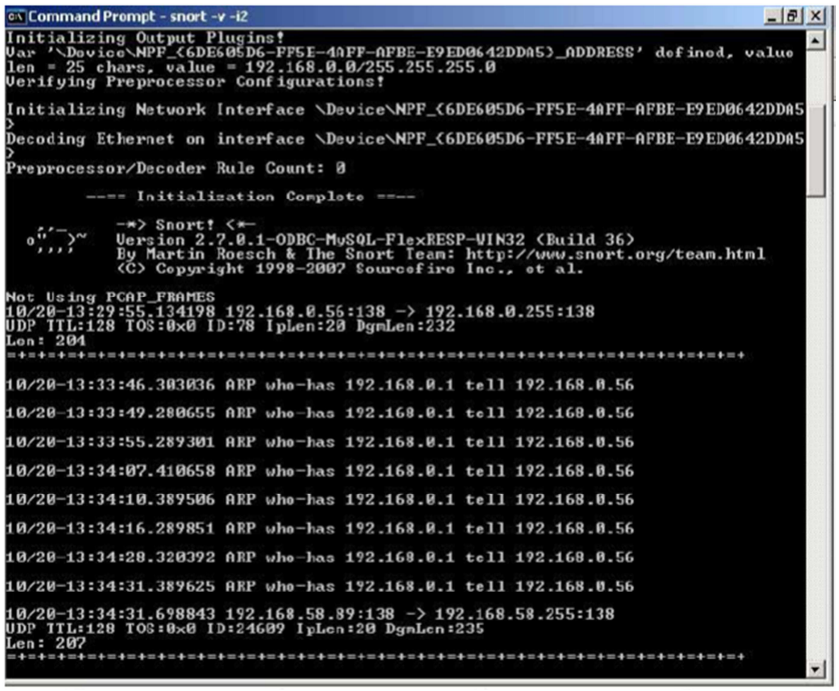
- Sniffer način
- Packet logger mode
- NIDS mod

8.2. Sniffer način rada

Za upoznavanje sa Snort-om koristi se sniffer način rada koji radi obični ispis paketa u komandnoj liniji. Da bi ispisali zaglavlja ICMP/TCP/UDP paketa (usput proverili da li je sve dobro instalirano i da li WinPcap dobro funkcioniše) koristimo :

Snort –v –i2 (sa i2 smo rekli da koristi lokalni mrežni interfejs, a ne generic adapter).

Na ekranu se dobije sledeći izgled:



Slika. 14. *Izgled izlaza Snort-a u komandnoj linij sa komadnom snort –v*

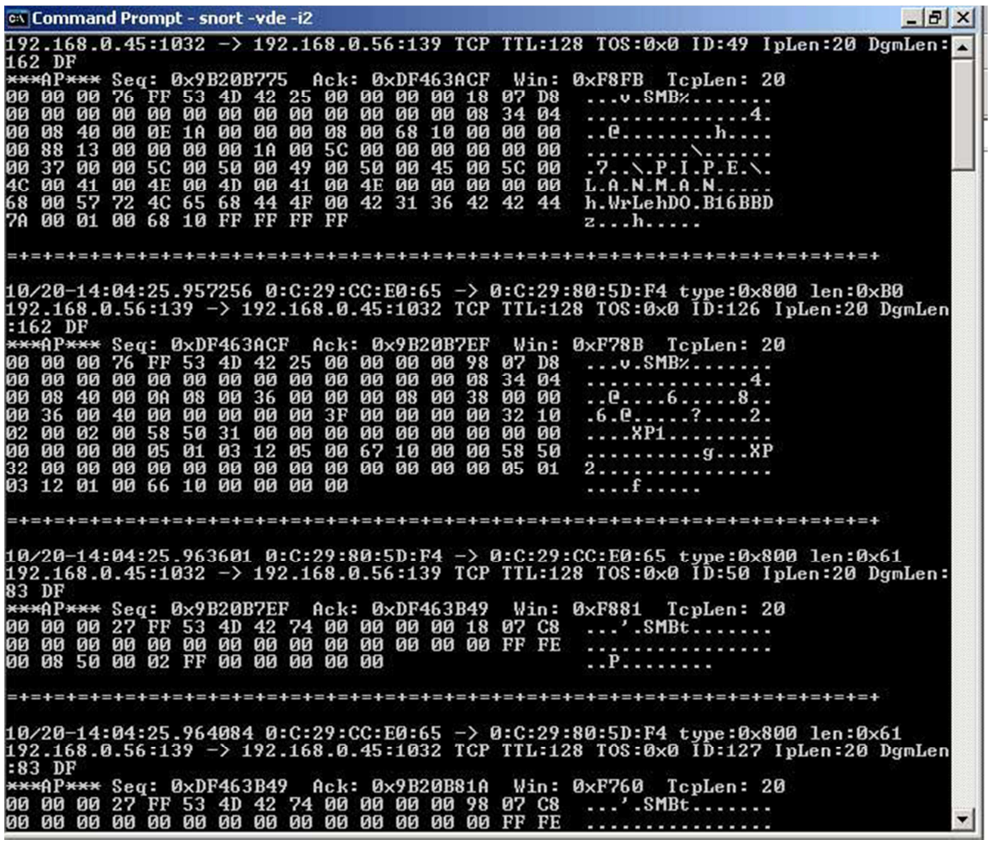
NAPOMENA: Ako imamo više mrežnih interfejsa na jednom računaru možemo ih izlistati korišćenjem komande :

- **snort –w**

Ako želimo da proverimo kako Snort uzima sadržaje paketa, a ne samo zaglavlja onda korsitimo:

- **snort –vde –i2**

Parametar **–d** prikazuje sadržaje paketa aplikacijskog sloja. Dobijemo sledeći izlaz



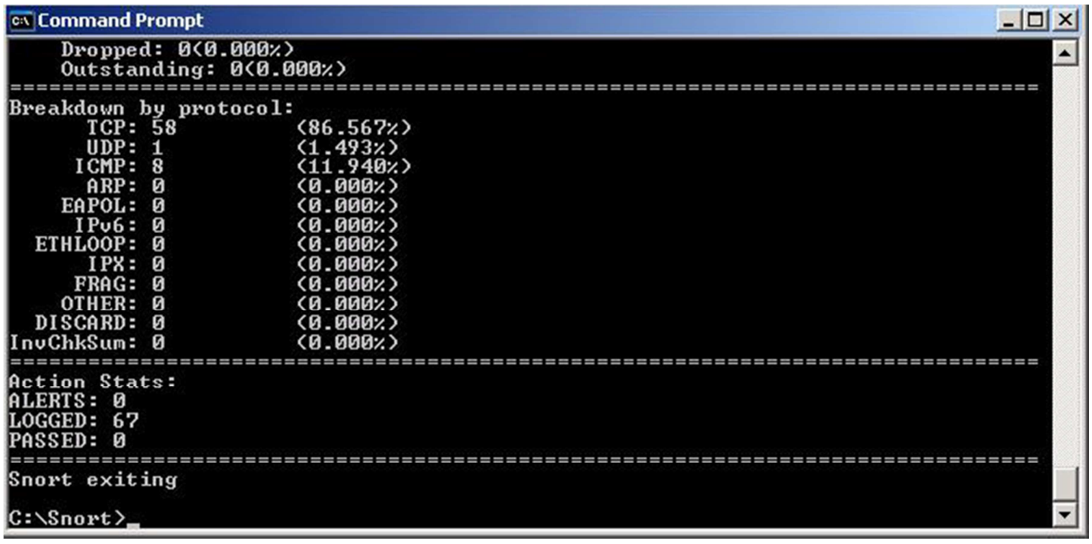
Slika 15. *Izgled izlaza Snort-a u komadnoj linij nakon komande snort –vd*

8.3. Paket log mod

Testiranje Snort-ovih mogućnosti logovanja moguče je sa -l (log) dodatkom komandnoj liniji. Testirano je sa sledećom komandom (pre toga je postavljen trenutni direktorijum u direktorijum c:\snort):

- **snort -dev -l./log –i2**

Nakon pokretanje ove komande proverili smo folder c:\snort\log i tamo našli faj l kojeg je kreirao.



Slika 16. *Izgled komandnog prozora nakon pokretanja snorta u log modu*

Na slici vidimo da je Snort prepoznao 58 paketa TCP protokola, 1 UDP i 8 ICMP ukupno je logovano 67 paketa. Takođe, testirani su i najbrži načini logovanja koristeći komandnu liniju, a to je binarni način (binary mode) koji se pokreče sa komandnim parametrom -b:

- **snort -dev -l./log –b –i2**

Razlika u korišćenju -b parametra je u tome što Snort loguje sav svoj izlaz u samo jedan fajl i specifikovanje parametra -d ili -e više nije neophodno pošto binary način kupi standardno cele pakete u log, a ne samo delove. Jednom kad su paketi logovani u binarni fajl, taj fajl možemo čitati sa bilo kojim drugim alatom za snifanje koji podržava tcpdump format (alati tipa tcpdump ili Ethereal, Wireshark i sl.). Snort takođe se može koristiti za čitanje paketa u binary formatu i da bi se to postiglo potrebno je samo specifikovati -r parametar u komandnoj liniji.

8.4. Nids mod

Da bi omogućili NIDS mod (mod mrežnog sistema za detekciju) u kojem neđemo svaki paket bezpotrebno logovati nego praviti selekciju na osnovu nekih pravila možemo pokušati sledeću komandnu opciju :

- **snort -dev -l./log -c snort.conf -A fast –i 2**

Ovo stavlja Snort da radi u svom najosnovnijem modu i da loguje pakete da zadovolje neka pravila definisana u snort.conf u log direktorijumu. U svrhu upoznavanja sa mogućnostima Snort-a mi ćemo koristiti samo jedno pravilo i kreirat ćemo fajl testconf.test u koji ćemo

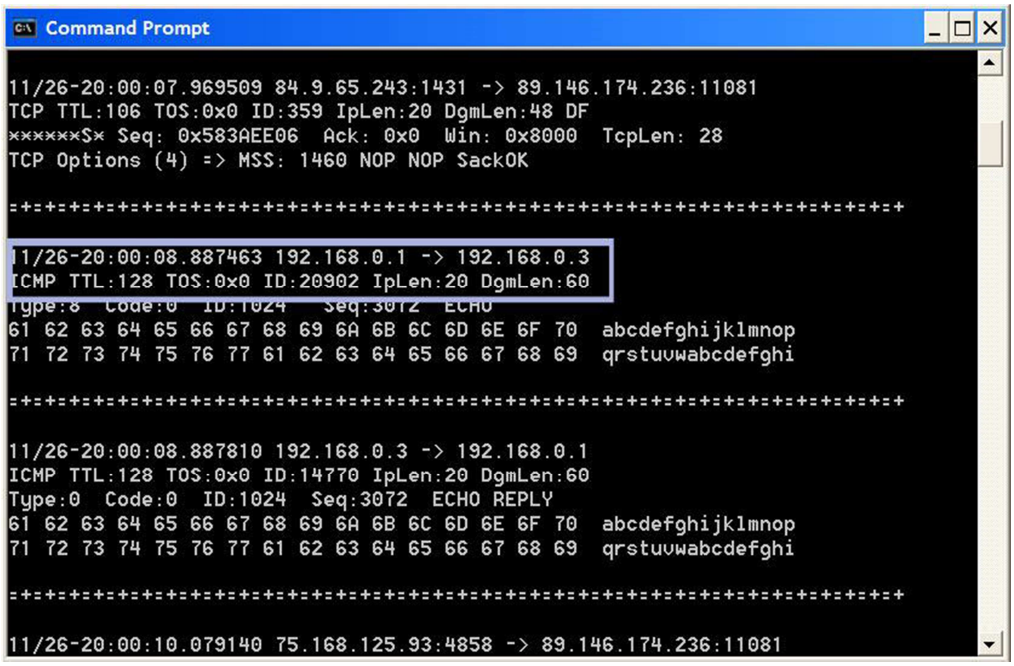
staviti to pravilo. Kao primer uzet ćemo pravilo koje će praviti uzbune za bilo koji ICMP saobraćaj:

- **alert icmp any any -> any any (msg:“Lokalni ICMP SAOBRACAJ TEST“; itype:8; sid:2000000;)**

Pokrenut čemo Snort sa komandom :

- **snort -dev -l./log -c testconf.test**

U svrhu testiranja, u komandoj liniji napravićemo ping prema nekoj mašini (u našem slučaju 192.168.0.3) sa komandom ping 192.168.0.3. U komandoj liniji gdje smo pokrenuli Snort vidimo da je Snort prepoznao ICMP saobraćaj:



Slika 17. *Testiranje : Prepoznavanje ICMP saobraćaja*

ZAKLJUČAK

Sistemi za detekciju upada spadaju u noviji trend alata za bezbednost računarskih sistema i računarskih mreža. Vremenom će se pokazati da sigurnost jednog modernog poslovnog okruženja ne može biti obaveza jednog alata (često samo firewall-a), nego i alata koji imaju mogučnost da prepoznaju same detalje sadržaja saobraćaja u mreži. Naravno, treba spomenuti da u sigurnosti veliku ulogu imaju ljudi koji de svojom sposobnošću morati da odole današnjem brzom toku razvoja kako sigurnosnih alata, tako i alata i metoda za narušavanje sigurnosti. Takođe, u doba kada dolazi do brzog razvoja računarskih i internet tehnologija, neretko neproverenih softverskih rešenja, te bitnim promenama koje je doveo internet sa svim svojim manama i prednostima, bezbednost de sve više tražiti svoje neizostavno mesto u računarskom informatičkom dobu. Internet je postao mesto velike razmene znanja, iskustva, kako iz dobrih tako i iz loših namera, i to je danas uzrokovalo da broj bezbednosnih pretnji ima stalan uspon. Do pojave OSSIM distribucije izrazito velik problem administratora računarskih sistema bio je vezan uz odabir najboljih i najefikasnijih bezbednosnih alata za zaštitu mrežnih sistema i povezivanje njihova rada u skladnu celinu. Velik deo vremena tako se gubio samo na usklađivanje različitih izveštaja i bezbednosnih obaveštenja, podešavanje rada velikog broja međusobno komplementarnih alata te dugotrajne analize često nepreglednih bezbednosnih izveštaja. Sve skupa otežalo je osnovni zadatak svih administratora - efikasnu zaštitu sistema koje administriraju.

Snort je jedan od prvih sistema za detekciju upada koji se razvio delimično zaslugama interneta. Od 1998. godine kada je prvi put napisan Snort koji je tad više ličio na obični prisluškivač mrežnog saobraćaja se razvio u jedan od vodećih alata za detekciju mrežnih bezbednosnih pretnji koji vrlo lako parira i mnogim drugim komercijalnim rešenjima iz oblasti sistema za detekciju upada. Zajednica koja radi u razvoju Snort-a je zasnovana na slobodnoj otvorenoj zajednici, inače tipičnoj za projekte Open Source, što znači da praktično svako ko ima znanje da unapredi sistem za detekciju Snort-a može slobodno da učestvuje u poboljšanju tog alata. Taj oblik evolucije znanja, mogućnosti, prilagođavanja u današnje internet doba da ovaj sistem za detekciju postane jedan od glavnih oružja od mnogobrojnih alata koji koriste propuste u implementaciji brojnog aplikativnog i sistemskog softvera kojeg proizvođači neretko više zbog novca nego kvaliteta stave brzo u tržište. Snort je vrlo prilagodljiv, modularan alat za detekciju pokušaja iskorišćavanja takvog neispitanog, nedovršenog ili slabo-testiranog softvera.

LITERATURA

[1] Wikipedia: arpwatch, http://en.wikipedia.org/wiki/Arpwatch,

[2] Wikipedia: P0f, http://en.wikipedia.org/wiki/P0f,

[3] Sourceforge: PADS, http://passive.sourceforge.net/,

[4] Wikipedia: Snort (software): http://en.wikipedia.org/wiki/Snort_%28software%29,

[5] Wikipedia: ntop, http://en.wikipedia.org/wiki/Ntop,

[6] Wikipedia: Nagios, http://en.wikipedia.org/wiki/Nagios,

[7] Wikipedia: Snare (software), http://en.wikipedia.org/wiki/Snare_%28software%29,

[8] B. Laing, Implementing a Network Based Intrusion Detection System, Sovereign House, Reading, 2000.

[9] R. Rehman, Intrusion Detection Systems with Snort, Prentice Hall PTR, Upper Saddle River, New Jersey, 2003.

[10] J. McGibney, Intrusion Detection Systems & Honeypots, Waterford Institute of Technology, Ireland, Prezentacija sa konferencije INET/IGC, Barcelona, 2004

[11] B. Laing, Implementing a Network Based Intrusion Detection System, Sovereign House, Reading, 2000.

[12] Craig Gosselin, Open Source Intrusion Detection and Prevention, http://www.infosecwriters.com/text_resources/pdf/Open_Source_CGosselin.pdf

[13] Sourcefire, Snort, http://www.snort.org/

[14] Sandeep Kumar. Classification and Detection of Computer Intrusions. Ph.D. Dissertation, August 1995.]