Sistemi za detekcijo napadov

Domen Kožar, Andraž Brodnik

14. januar 2013

Kazalo

1 Povzetek

Teoretični del zajema razlago sistemov za detekcijo napadov (IDS) in preprečevanje napadov (IPS) ter njihove podskupine. Osredotočili se bomo na sisteme, ki opazujejo promet na omrežnem vmesniku (NIDS/NIPS).

V praktičnem delu pa smo namestili in nastavili sistem Snort.

2 Uvod

Namen seminarske naloge se je seznaniti s sistemi za zaznavanje vdorov ter s sistemi za preprečevanje vdorov. Kako sestaviti osnovno politiko (policy) za tak sistem, kakšne napade lahko detektiramo, ter priporočljive obrambne mehanizme.

Dandanes se srečujemo z novicami o nepoblaščenih vdorih v informacijske sisteme. Takšni vdori lahko uničijo podjetje ali zasebnost uporabnikov, kar pomeni, da je racionalno investirati nekaj tehničnih ur v postavitev sistema, ki bi lahko (ni pa nujno) taksen vdor preprečil ali pa zaznal poskus vdora. To nam koristi, da vidimo na kaksen način je napadalec napadel naš sistem, ter kaj je storil.

Kljub temu, da je naša varnostna politika v skladu z dobro prakso (menjava gesel, dvonivojska avtentikacija, požarni zidovi, varne aplikacije, tuneliranje prometa, up-to-date strežniki), ne smemo biti preveč zadovoljni s sami sabo ter moramo postavit tudi sistem za detekcijo in/ali preprečevanje napadov.

Naj bralca opozorimo tudi na dejstvo, da 'nepravilno' konfiguriran IDS ali IPS sistemi lahko globoko posežejo tudi v zasebnost posameznika, kar ni v skladu z ustavo Republike Slovenije in drugimi pravnimi akti. IDS in IPS sistemi se lahko uporabljajo tudi kot DPI (deep packet inspection) sistemi, kar pomeni, da ne gledamo samo glav paketnih protokolov ampak tudi aplikacijski nivo (aplikacijski protokol oz. vsebino), zato nastavljajmo IDS in IPS sisteme odgovorno, podatke pa shranjujmo z največjo skrbjo.

3 Teoretični del

3.1 IDS

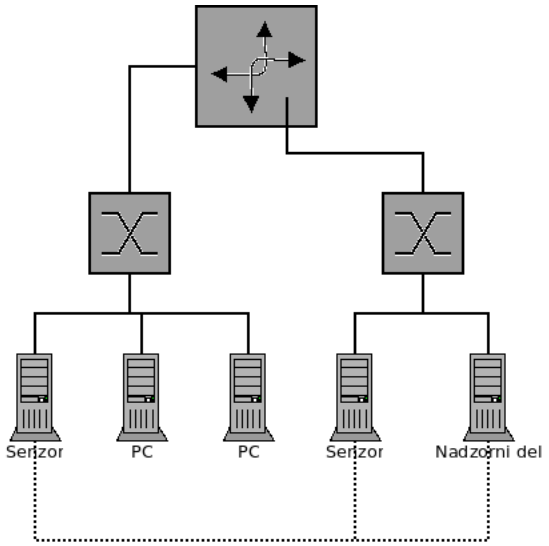
Sistemi za detekcijo napadov (intrusion detection system), krajše IDS so sistemi, katera naloga je analizirati podatke na omrežju ali sistemu samem ter zaznati poskuse vdora ali pa vdor sam.
Naj omenimo, da so tej sistemi namenjeni ponudnikom storitev (podjetjem, inštitucijam, posameznikom) v večini niso namenjeni omrežnim operaterjem, razen če želimo preprečevati napade na naše omrežne elemente. Ne moremo pa vsiljevati pravil za vse naše uporabnike. Včasih pa je bilo tega prometa za analizo preveč, a vendar so se časi spremenili in to ni več glavna omejitev.
Delimo jih na dve glavni skupini:

- NIDS (network intrusion detection system)
- HIDS (host intrusion detection system)

3.2 NIDS

Sistemi, katerim je glavni vir podatkov za analizo izključno omrežje se imenujejo NIDS sistemi (network intrusion detection system). Kar pomeni, da opazujejo ves dohodni in izhodni promet, nato pa indentificira sumljive vzorce, ki bi lahko kazali na napad na omrežje ali nek sistem.
Bralcu bo po vsej verjetnosti poznan program WireShark ali pa tcpdump. NIDS ponavadi delujejo podobno kot zgoraj omenjena programa. Program zajema vse paketke, ki jih vidi na omrežnem vmesniku, nato jih premerja s pravili v svoji bazi, sumljive pakete ali niz paketov pa zabeleži ali pa si ustrezno napiše informacije o njih.

3.2.1 Komponente



Slika 1: Primer postavitev komponent

Sami NIDS sistemi so ponavadi razdeljeni na 2 dela:

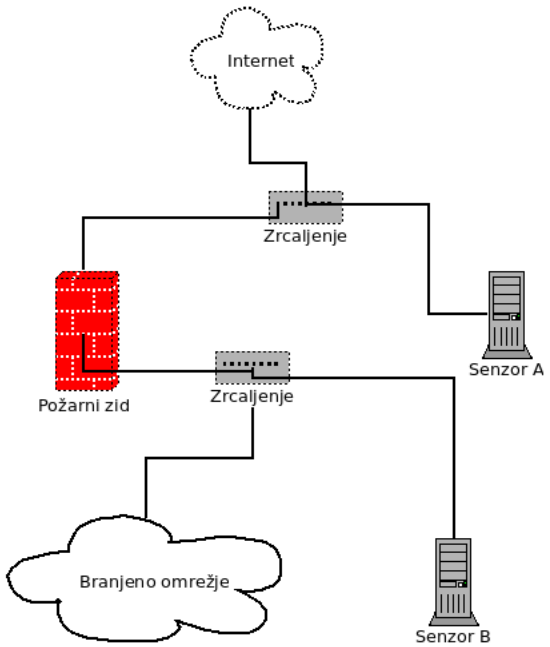
- Senzor
- Nadzorni del

Senzorji so programi, ki zajemajo paketke na določenem delu omrežja jih analizirajo nato pa pošljejo nadzornem delu.
Takšna delitev ima več prednosti. Prva prednost je, da se dogodki/alarmi počiljajo naprej torej so reproducirani, tako je praktišno nemogoce za napadom pobrisati sledi. Hkrati lahko zajemamo več omrežij, ki so na različnih lokacijah, brez, da bi promet preosmerili na centralno lokacijo. Oz. lahko dvignemo več instanc (slovenski prevod primerek je neprimeren) senzorjev na enem samem strežniku, kar nam omogoča bolšo uporabo računskih virov na strežniku. To nam seveda oteži samo nastavitev (konfiguracijo) sistema, kar pomeni, da je takšna postavitev smiselna za večja podjetja.

3.2.2 Detekcija napadov

Kako poteka sama detekcija napadov? NIDS sistemi imajo vgrajeno bazo odtisov napadov. Če je dogajanje na mreži podobno, opisanem napadu v bazi bo NIDS sistem to napisal v dnevnik ali pa si celo shranil napad (v pcap datoteki). NIDS sistemi so sumljičavi, glede velikosti, samega protokola in količine paketov. NIDS sistemi imajo vgrajene tudi logične avtomate, torej se zavedajo tudi paketov pred nekim paketom in po njem.

3.2.3 Lokacija v omrežju



Slika 2: Primer dveh lokacij senzorjev: Senzor A pred požarnim zidom in senzor B za njim

NIDS sisteme lahko postavimo na več delov v sistemu. Najbolj pogosto je za požarnim zidom. Z tem zajemamo vse pakete, ki jih požarni zid ni zavrgel. To nam sicer ne, da celotnega uplogleda v vse napade. Nam pa zmanjša količino podatkov, ki jih moramo obdelati. Napadi, ki nam ne

pridejo skozi požarni zid načeloma niso nevarni.
NIDS senzorje imamo lahko razdeljene tudi na različnih fizicnih lokacijah ali logicnih lokacijah, ker ni mogoče vse realizirati z enim senzorjem ali pa si to ne želimo (zahtevana je separacija).

3.2.4 Omejitve NIDS

Lažni alarmi Pomembna mejitev je pogostost lažnih alarmov. Noben NIDS sLažni alarmiistem ne more preprečit pojavljanje lažnih alarmov. V večino NIDS sistemom je mogoče dodati tudi vzorce lažnih alarmov. V bazo dodamo vzorec, ki nam sproži alarm vendar NIDS sistem najde pravilo, da je to lažni alarm.

TCP tok/IP fregmentacija Sestavljanje TCP toka podatkov (stream)/Sestavljanje IP paketkov (zaradi defregmentacije)
Kot omenjeno kdaj analiziramo celotne TCP toke podatkov, da lahko zaznamo nevarnost. To pomeni, da moramo shranjevati paketke. Pri napadih mnogokrat ne zaključimo toka podatkov, kar pomeni, da se more pri shranjevanju tokov podatkov NIDS obnasati zelo specifično. Podobne probleme imamo tudi pri IP paketih. Kot vemo imamo omejen polnilnik. Predstavljate si lahko koliko paketkov lahko shranimo na 10Gbit/s vmesniku.

3.2.5 Stvari, ki jih moramo premisliti

Operacijski sistem Pomemben je operaciski sistem. Ponavadi jo NIDS sistemi navoljo za vse sisteme, tako Windows NT kot Unix ter ostale. Neki sistemi lahko tečejo na več različnih, tako ni nič nenavadnega da imamo senzor na OpenBSD, managment pa na Windows NT sistemu. Pomembno je da sistemski administratorji poznajo platformo na kateri teče NIDS ali del njega, saj je pomembno, da je ta ustrezno zasčitena.

Podprti omrezni vmesniki Pomembno je da se zavedamo, da niso podprti vsi omrežni vmesniki. Vecina sistemov podpira samo ethernet vmesnike.

Alarmi Pomembno je, da se odločimo kakšne načine obveščanja se bomo poslužili. Ponavadi so detekcije (alarmi) razdeljeni v različne tipe ali nivoje. Glede na katere določimo kanal obveščanja. Nekatere napade lahko zgolj napišemo v dnevnik, za nekatere pošljemo e-posto skrbniku ali operatorju v nekaterih primerih pa pošljemo SMS.

Pisanje dnevnikov in poročil Vsi NIDS sistemi pišejo alarme v dnevnike, torej lahko za nazaj pogledamo, kaj se je dogajalo. Določeni sistemi pa omogočajo tudi pošiljanje avtomatsko generiranih porocil (npr. dnevnih).

Vzdrževanje Pomemben segment je tudi vzdrževanje. Vprašanja glede tega so. Ali se nam baza odtisov, sama posodablja ali pa jo moramo avtomatsko posodobiti? Kako se posodablja jedro? Moramo za posodobitve baze plačevati. Koliko stane licenca? Imamo več alternativnih virov odtisov? Je okoli odtisov v bazi zbrana skupnost ali pa celo akademsko okolje. Koliko fleksibilni so ti mehanizmi. Koliko pogosto moramo posadabljati. Stvar zavisi od naših izkušenj in od izbranega sistema. Ni pa enoličnega odgovora za takšna vprašanja.

Izgled nadzornega sistema Pomemben je izgled nazornega dela. Ali je nadzorni del prijazen in ga hitro razumemo. Nadzorni deli niso namenjeni konfiguraciji, ampak predvsem prikazu alarmov, a lahko kjub temu omogočajo preproste nastavitve. Vecina komercialnih sistemov nam ponuja grafične vmesnike. Realizirane za različne platforme ali pa kar spletni vmesnik. Določeni sistemi nam dajo na razpolago programerski vmesnik (API), nato pa vzamemo drug projekt za prikaz rezultatov.

Skalabilnost Skalabilnost je pomemben aspekt, če nacrtujemo sistem za omrežja, ki so velika ali pa se bodo razširila. V tej točki, nas zanima ali NIDS sistem izkorišča vse procesorje na strežniku oz. kako to doseči ali lahko paktetke pošiljamo naprej. Drugim računalnikom in tako dosežemo neko kolektivno intelegenco. Koliko paketkov lahko realno zajamemo in obdelamo, preden jih začnemo spuščati (ne analizirati).

3.3 NIPS

Veliko smo napisali o NIDS sistemih, vendar kaj so NIPS (network intrusion prevention system) sistemi? NIPS je sistem za preprečevanje napadov, ki glede na alarme NIDS vgrajenega sistema izvedejo določene akcije, ki ta napad preprečijo ali pa ga ustavijo.

Bralca naj te termini ne zmedejo saj se jih ne uporablja strikno. Korak od IDS do IPS sistema je zelo majhen zato se veliko IDS sistemov deklarira kot IPS.

3.3.1 Načini preprečevanja

Spreminjanje pravil v požarnem zidu Najbolj pogosto NIPS sistemi spreminjajo pravila požarnega zidu. Če je NIPS sistem vgrajen v požarni zid potem je to sila preprosto. Drugače pa mora komunicirati preko malih programov ali pa protokolov za oddaljen nadzor (SSH, netrpc).

Kot primer si predstavljamo, da NIPS sistem zazna napad napise pravilo v požarni zid, da vse pakete, ki prihajajo iz izvornega IP naslova napadalca zavrze za določeno časovno obdobje (npr. 20 minut).

Poslužimo se lahko seveda tudi filtriranje glede na TCP/UDP port.

Ta način se uporablja v večini primerov.

Spreminjanje usmerjanja Če imamo možnost ali pa veliko prometa lahko uporabimo zgornji način malo drugače. V primeru, da so napadi res hudi. Lahko usmerjevalni protokol nastavimo tako, da nastavimo neveljavno pot (null route) za napadalčev naslovni prostor.

Obveščanje/nadzor ostalih storitev Možno je tudi pisanje posebnih pravil, ki začasno spremenijo nastavitve aplikacijskih strežnikov in drugo.

3.3.2 NIDS/NIPS produkti

Tipične sistemi v praksi so:

- Snort (odprtokodni sistem)
- Cisco IPS/IDS
- Niksun NetDetector

- ISS RealSecurea

Naj omenimo najbolj pogosta:

Slika 3: Snort

Snort Snort[7] je odprtokodni NIDS/NIPS sistem, ki ga razvija podjetje Sourcefire[6] s svojimi 560 zaposlenimi. Prvotno je leta 1998 sistem Snort napisal Martin Roesch, ki je sedaj tehnični direktor podjetja. Snort je čez leta postajal čedalje bolj pomemben v informacijski industriji, uspeh pa je zgled za celotno odprtokodno skupnost.

Snort je izdan pod GPL licenco, napisan je v C programskem jeziku ter deluje na različnih operacijskih sistemih.

Snort ima močno komercialno podporo. Glavna storitev je ponudba pravil za snort konfiguracijo.

Ponavadi ga namestimo na GNU/linux sistem. Pri tem je lahko strojna oprema poljubnih proizvajalcev, se pozna če imamo PCI-X režo za omrežni vmesnik. Strežnik ima lahko več jeder za obdelavo podatkov. Snort je mogoče poganjati samo na enem jedru, vendar lahko procesi med sabo dobro komunicirajo (clustering), saj je snort narejen z mislijo obdelave veliko podatkov. Za neko srednje podjetje se priporoča strežnik, ki ima dva omrežna vmesnika ter 4 jedra ali več.

Dva omrežna vmesnika sta nujna saj na enem zajemamo promet, na drugem shranjujemo paketke na NAS, izvajamo IPS akcije/sankcije ter nadzorujemo sistem. Taksnih postavitev smo navadno vajeni pri omrežnih elementih, kjer imamo za nadzor ločene komunikacijske kanale.

OISF[8] je začel razvijati novo implementacijo, ki uporablja Snort pravila imenuje se Suricata. Glavne prednost je da lahko uporaljablja več jeder ter privzeto podpira IPv6.

Slika 4: Cisco

Cisco Cisco ponuja te sisteme v ASA seriji 5000[4], kot je razvidno na njihovi strani. Najbolj zmogljivi sistem lahko obdela 400Mbit/s podatkov, to pomeni z hkratnim delovanjem tako požarnega zidu, kot IPS sistema.
Cisco ima stvari dobro integrirane med vsemi napravami. To omogoča njihov sistem SIO (Security Intelligence Operations). Hkrati pa nam osvežujejo pravila.

3.4 HIDS

HIDS sistemi (host intrusion detection system) so sistemi za detekcijo napadov, ki delujejo na samem strežniku ali pa napravi, ki ponuja neko storitev odjemalcem. Torej HIDS sistem zanima samo gostitelj (host). Zaznava tudi paketke iz omrežja, vendar samo tiste, ki so namenjeni gostitelju. Ne zaznava pa samo paketkov. To pomeni da imamo senzorje na vsaki napravi, ki jo želimo ščititi. Te senzorje ponavadi imenujemo agenti. Zakaj bi postavili se HIDS na strateško pomembnih napravah? Primarno, ker gre za drug nivo zaznave in preprečevanja.

3.4.1 Omejitve HIDS

Glavni problem HIDS sistemov je, da so tesno povezani z platformo na kateri delujejo oz. sam operacijski sistem. Kot prej omenjeno moramo imeti na vsaki napravi nameščenega agenta. Ta agent ponavadi vedno pošilja vse podatke naprej nadzornemu sistemu. Nikoli se ne zanašamo, da je sistem pod našim nadzorom in v njega se ni bilo vdrto.

3.4.2 Načini zaznav napadov

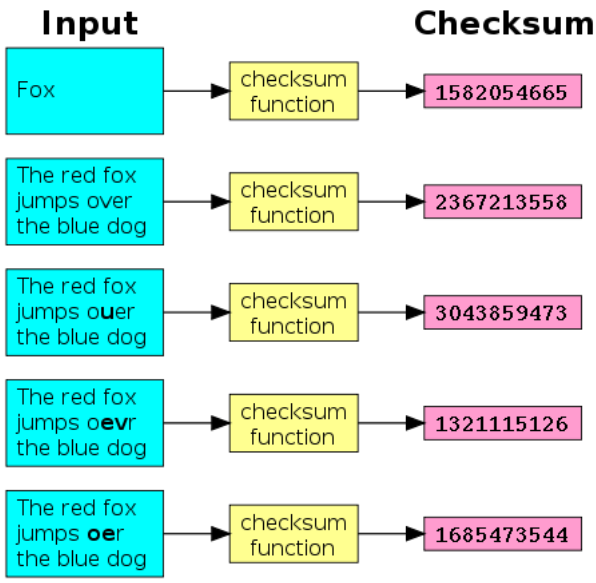
Poslušanje omrežnega vmesnika HIDS tudi posluša omrežni vmesnik na napravi. Podobno kot to dela NIDS, kar smo opisovali v prejšnih poglavjih.

Spremljanje integritete datotek Na predavanjih smo obravnavali tako imenovane zgoscevalne funkcije.

Te funkcije se uporabljajo tudi pri IPv4 paketih in drugje (za izračun kontrolne vsote). Funkciji podamo podatke ona pa vrne kluč oz. rezultat. Pomembno je, da se zavedamo, da se pri tem izgubi informacija podatka, ki smo ga ustavili v zgoščevalno funkcijo. Pri kriptoloških zgoščevalnih funkcijah je težko izračunati poljuben podatek, če poznamo kluč.
V UNIX operacijskih sistemih HIDS sistemi naredijo kontrolno vsoto večino datotek v /etc mapi. V /etc mapi se nahajajo konfiguracije pomembnih storitev, ki jih taka naprava/gostitelj ponuja. Kontrolne vsote pošlje naprej nadzornemu sistemu. To naredi vsakič, ko se datoteka spremeni, ce datotečni sistem to podpira oz. kronološko.
Nadzorni sistem primerja prejšno vrednost z novo, če se ne ujemata uporabnika o tem obvesti. Uporabnik nato presodi ali je to pričakovano (če je prislo do rekonfiguracije) ali pa je napadalec, ki je vdrl v sistem spremenil te datoteke. Zato seveda uporabljano SHA1 funkcijo, ki je kriptografsko varna, kar pomeni da napadalec ne more ustvariti datoteke, ki bi imela isti kluč. Kar se pri MD5 seveda da narediti zelo poceni (oblačne storitve).

To nam pride se kako prav pri sistemih/gostiteljih, kjer posegi v konfiguracijo niso pogosti.

Eden preprostejših sistemov, ki ponuja samo to metodo zaznavanja napadov je AIDE[5].



Slika 5: Graficna podoba kako deluje zgoščevalna funkcija

AIDE nam zgenerira bazo kontrolnih vsot datotek in map in jo shrani v datoteko. Nato se moramo sami potruditi in jo skopirati na drug sistem. V drugi iteraciji lahko baze med sabo primerjemo in si ogledamo rezultat sprememb. Logiko kopiranja in primerjanja moramo napisati sami.

Imamo pa tudi bolj pametne sisteme, ki te stvari urejajo sami. Primer sporočila, ki nam ga pošlje naprednejši sistem OSSEC iz naslova (ossecm@moon.brodul.org):

```
OSSEC HIDS Notification.
2013 Jan 11 04:44:51

Received From: moon->syscheck
Rule: 550 fired (level 7) -> "Integrity checksum changed."
Portion of the log(s):

Integrity checksum changed for: '/usr/bin/gpgsplit'
Size changed from '44664' to '44696'
Old md5sum was: 'edda24df7c85faec9d155faa7122e222'
New md5sum is : '4b47774d265adf26db424436d59501dd'
Old sha1sum was: 'e49f84977fb1c1ad10796f5071abda7b3c6810d6'
New sha1sum is : 'd80551f82582f2758cc2c22ffc15428f0399ba49'
```

Spremljanje dnevnikov HIDS sistemi spremljajo dnevnike (logs). Tukaj ponavadi zaznajo največ poskusov napadov. Gledajo posamezne storitve, ki so potrebne za dostop do sistema/gostitelja. Opazujejo Telnet in ssh. Ali so poskusi prijav skozi te sisteme uspešni ali pa niso. Tako lahko detektirajo različne napade tako napade z pomočjo grobe sile (bruteforce) in z pomočjo slovarjev (dictionary) napade. V kolikor se v določenem časovni periodi uporabnik prevečkrat zmoti, nas sistem o tem obvesti.

Primer sporočila, ki nam ga pošlje naprednejši sistem OSSEC iz naslova (ossecm@moon.brodul.org):

```
OSSEC HIDS Notification.
2013 Jan 10 03:24:39

Received From: moon->/var/log/auth.log
Rule: 5551 fired (level 10) -> "Multiple failed logins in a small period of time."
Portion of the log(s):

Jan 10 00:24:37 moon sshd[8999]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:32 moon sshd[8997]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:28 moon sshd[8995]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:24 moon sshd[8993]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:20 moon sshd[8991]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:15 moon sshd[8989]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:11 moon sshd[8987]: pam_unix(sshd:auth): authentication failure; logname= uid=
Jan 10 00:24:07 moon sshd[8985]: pam_unix(sshd:auth): authentication failure; logname= uid=



 --END OF NOTIFICATION



OSSEC HIDS Notification.
2013 Jan 10 03:24:41

Received From: moon->/var/log/auth.log
Rule: 5720 fired (level 10) -> "Multiple SSHD authentication failures."
Portion of the log(s):

Jan 10 00:24:39 moon sshd[8999]: Failed password for root from 115.95.166.247 port 35283 s
Jan 10 00:24:35 moon sshd[8997]: Failed password for root from 115.95.166.247 port 34011 s
Jan 10 00:24:30 moon sshd[8995]: Failed password for root from 115.95.166.247 port 60921 s
Jan 10 00:24:26 moon sshd[8993]: Failed password for root from 115.95.166.247 port 59839 s
Jan 10 00:24:22 moon sshd[8991]: Failed password for root from 115.95.166.247 port 58575 s
Jan 10 00:24:17 moon sshd[8989]: Failed password for root from 115.95.166.247 port 57298 s
Jan 10 00:24:13 moon sshd[8987]: Failed password for root from 115.95.166.247 port 56150 s
Jan 10 00:24:09 moon sshd[8985]: Failed password for root from 115.95.166.247 port 54788 s
```

```
--END OF NOTIFICATION
```

IP v primeru je seveda spremenjen.

Zaznavanje 'rootkit' zlonamernih sistemov HIDS sistemi lahko detektirajo tudi takoimenovane 'rootkit' sisteme. Detekcija teh je kompleksen proces zato se v detaile nebi spuščali.

3.5 HIPS

Podobno kot pri NIPS sistemih je mogoce tudi pri alarmih HIDS sistemov narediti ustrezne akcije.

3.5.1 Načini preprečevanja napadov

Spreminjanje lokalnega požarnega zidu Možno je spreminjanje pravil na lokalnem požarnem zidu (samega gostitelja). Podobno kot pri NIPS sistemih.

```
Tue Jan  8 12:08:11 CET 2013 /opt/ossec/active-response/bin/host-deny.sh delete - h1412241
Tue Jan  8 17:15:54 CET 2013 /opt/ossec/active-response/bin/firewall-drop.sh add - 198.15.
Tue Jan  8 17:15:54 CET 2013 /opt/ossec/active-response/bin/host-deny.sh add - 198.15.109.
Tue Jan  8 17:26:27 CET 2013 /opt/ossec/active-response/bin/host-deny.sh delete - 198.15.1
Tue Jan  8 17:26:27 CET 2013 /opt/ossec/active-response/bin/firewall-drop.sh delete - 198.
```

Suspendacija uporabnika za določen čas Ponavadi HIPS sistemi omogočajo izklop avtentikacije za določega uporabnika na samem sistemu. Torej izklopijo uporabnika za časovno obdobje napada, nato pa ga spet vklopijo.

Povrnitev nastavitev Če se zazna sprememba konfiguracije, je mogoče ob spremembi skopirati konfiguracijske datoteke iz oddaljenega sistema.

Odjava napadalca Če je uporabnik neavtorizirano prijavljen, ga lahko odjavimo od seje.

4 Praktični del

Kot dober primer orodja za izvajanje NIDS ter NIPS se v industriji pogosto uporablja Snort. Naziv "The de facto standard for intrusion detection/prevention"povzame njegovo široko uporabo v industriji.

4.1 Nameščanje programske opreme Snort

Snort sva namestila na Gentoo Linux distribuciji z naslednjim ukazom:

```
$ emerge -av snort
```

Gentoo ponuja ogromno opcij za konfiguracijo Snort-a, najpomembnejše so:

- perfprofilling - Omogoči širok nabor zbiranja statistik o delovanju Snort-a
- paf - Omogoči "Protocol Aware Flushing", sledenje kompletnemu toku paketkov v komunikaciji
- active-response - Omogoči odgovarjanje (ICMP, TCP) na prekinjene povezave
- decoder-preprocessor-rules - Omogoči akcije (alert, drop, pass, etc) za dekoder in preprocesor
- react - Omogoči poslušanje, prekinitev ter preusmeritev HTTP prometa
- inline-init-failopen - Prepreči, da bi promet šel skozi Snort, preden je pripravljen na delovanje

V najinem primeru sva ohranila privzete nastavitve. Nato sva pognala Snort z naslednjim ukazom:

```
$ /etc/init.d/snort start
* Starting snort ...
```

Preverimo, če Snort res deluje:

```
$ ps aux | grep snort
snort    18627  1.5  9.8 621928 178984 ?       Ssl  19:49   0:00 /usr/bin/snort --nolock-
```

4.2 Zmožnosti ter konfiguracija

Snort lahko deluje v treh načinih:

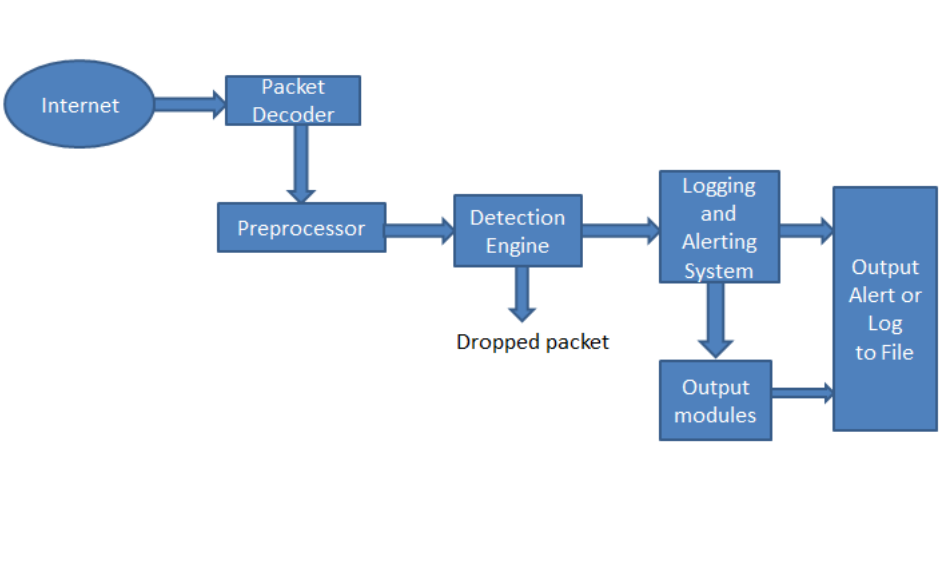
- Šniffer mode- prikazuje pakete (ter njihovo vsebino) v konzoli
- "Packet logger mode- zapisuje pakete (ter njihovo vsebino) v neko bazo
- "NIDS mode- izvaja analize za detekcijo nepravilnosti in sankcije na podlagi ugotovitev

Pri tem je vredno opozoriti, da glede na količino prometa je pomembno premisliti kolikšno količino podatkov Snort zmora premleti ter zmožnosti baze podatkov, kamor se shranjujejo.

Snort je sestavljen iz naslednjih kompontent, ki skrbijo vsaka za svoj del pri celotnem procesu obdelave paketov:

- Decoder - Dekodira pakete v Snort-u razumljiv format
- Preprocessor - Napredna pravila, ki pripravijo paketke ali/in sprožijo obvestilo
- Detection engine - Jedro Snort-a, ki požene vsa pravila čez razdelan paket
- Thresholding - Obdela vsa obvestila in se odloči, ali jih pošlje glede na število in druge parametre
- Output modules - Načini obveščanja o zaznanih problemih

Komponente so med seboj povezane približno tako:



Slika 6: Snort komponente

Snort zna poročati o zaznanih nepravilnostih na naslednje načine:

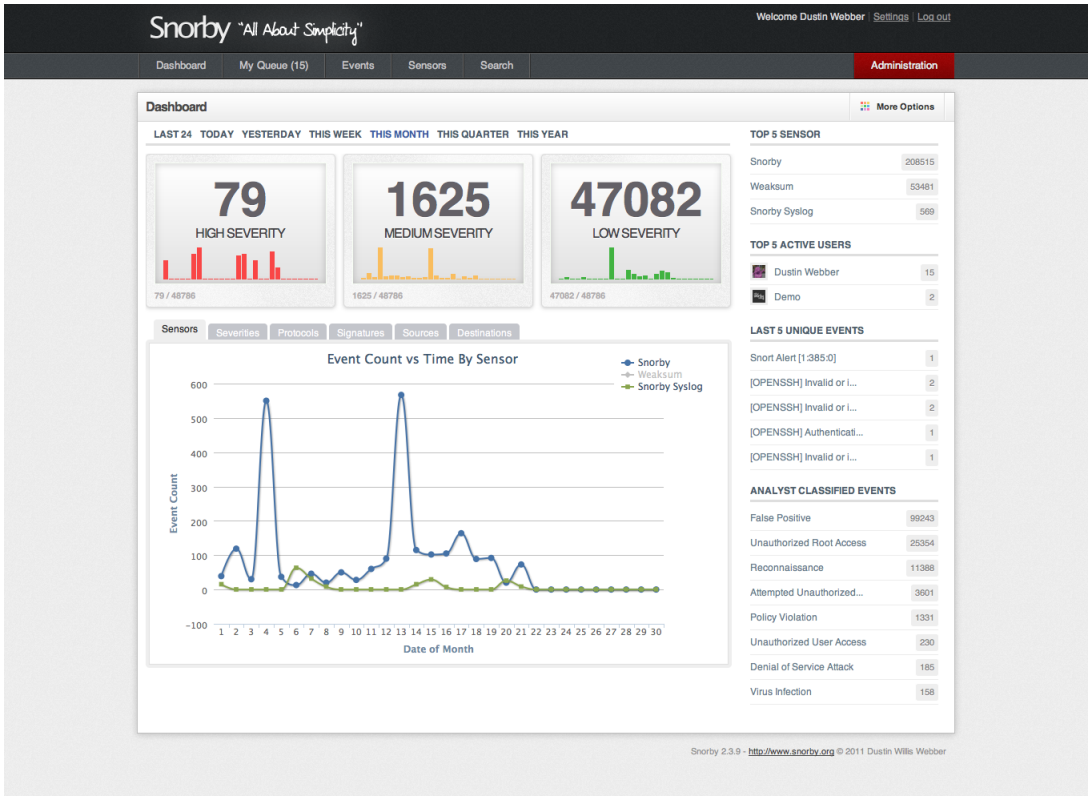
- syslog - Obvestila preda naprej programu Syslog
- full - Pakete prikaže vključno z glavami paketov
- tcpdump - Pakete zapiše v .pcap format
- unixsock - Obvestila pošlje v Unix Socket
- csv - Obvestila zapiše v CSV formatu
- null - Obvestila zavrže

4.3 Spletni vmesnik za Snort

Spletnega vmesnika sama nisva postavljala, saj gre samo za predstavitev podatkov na drug način.

Zato sva raje preverila uporabnost Snorby spletnega vmesnika napisanega v ruby kar preko demo strani na http://demo.snorby.org/ z uporabniškim imenom demo@snorby.org in geslom snorby.

Prva stran vmesnika jasno prikazuje agragiranje podatke v smiselnih grafih:



Slika 7: Snorby dashboard

Snorby vsebuje tipične spletne bonitete kot so berljiva poročila obvestil, klasifikacija ter filtriranje obvestil v vmesniku, itd.

Uporaba spletnega vmesnika je predvsem zanimiva za hiter pregled nad stanjem omrežja.

4.4 Primeri zaznavanja anomalij v omrežju

4.4.1 Zaznavanje Port Scanninga

Port Scanning je širok pojem, zato bomo v tem primeru lovili primer, ko napadalec uporabi orodje NMap in ga uporabi samo dotični tarči ter pregleda celotni nabor vrat.

V konfiguracijsko datoteko /etc/snort/snort.conf vpišemo naslednji preprocessor:

```
preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }
```

Snort zazna dogodek in javi:

```
Jan 03 18:03:56 ananas snort[1815]: [122:1:1] PSNG_TCP_PORTSCAN [Classification: Attempted
```

4.4.2 Zaznavanje nepravilnosti v HTTP prometu

Privzeto je Snort HTTP preprocessor vkljopljen in obvešča o vseh dogodkih, ki se po RFC HTTP standardu ne bi smele zgoditi v praksi.

Na primer naslednje gre za primer HTTP requesta, ki ne vsebuje glav ČONTENT-LENGTH"ali "TRANSFER-ENCODING". Uporabno orodje za nadzor nad HTTP prometom in pravilnim delovanjem samega protokola.

```
Jan 03 18:37:47 ananas snort[11129]: [120:3:1] (http_inspect) NO CONTENT-LENGTH OR TRANSFE
```

4.4.3 Zaznavanje ARP Spoofinga

V konfiguracijsko datoteko /etc/snort/snort.conf vpišemo naslednji preprocessor:

```
preprocessor arpspoof
preprocessor arpspoof_detect_host: 10.2.1.3 00:aa:bb:cc:dd:ee
```

Ko ponovno zaženemo Snort in izvedemo "ARP Spoofing"napad na povezovalnem omrežnem nivoju, javi naslednjo obvestilo:

```
Jan 04 12:33:23 ananas snort[11129]: [112:4:1] (spp_arpspoof) Attempted ARP cache overwrit
```

Tako lahko uspešno ulovimo vse nepridiprave, ki se igrajo z varnostjo našega omrežja.

5 Rezultati

Na podlagi podanih informacij, eksperimentov ter gradiva, ki sva ga predelala, lahko vidimo, da postavitev sistemov za preprečevanje napadov predstavlja izziv od varovanjem zasebnosti do pisanjem pravil za preprečevanje napadov.

V praksi podjetja kupijo pakete pravil kar od podjetij kot je Sourcefire (na http://www.snort.org/snort-rules/) in tako "outsourcajo"problem podjetjem, ki se ukvarjajo izključno s takšnimi problematikami.

Kljub temu je vredno poudariti, da je Snort samo orodje in šele pravilna konfiguracija lahko pripelje do pozitivnih učinkov, pa še to ne pri vseh primerih.

Literatura

[1] Snort Manual, http://manual.snort.org/, 20.12.2012

[2] Slika 6, http://seclists.org/snort/2012/q3/att-894/image.png, 06.01.2012

[3] Slika 7, https://forrst-production.s3.amazonaws.com/posts/snaps/109759/original.png, 06.01.2012

[4] Cisco ASA 5000 serije, http://www.cisco.com/en/US/products/ps6120/prod_models_comparison.html, 27.12.2012

[5] AIDE, http://aide.sourceforge.net/, 27.12.2012

[6] SourceFire, http://en.wikipedia.org/wiki/Sourcefire,_Inc, 27.12.2012

[7] Snort, http://www.snort.org/, 27.12.2012

[8] OISF, https://www.openinfosecfoundation.org/, 27.12.2012