ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

DIPLOMSKI RAD br. 1889

Određivanje reputacije autonomnih sustava temeljeno na praćenju neželjenog prometa

Domagoj Eklić

Zagreb, 2011.

Sadržaj

1. Uvod

Prve mreže s preklapanjem paketa, *ARPANET* i *NSFNet,* svojevrsne prethodnice Interneta*,* bile su istraživački projekti Američke vlade i akademske zajednice. To su bile male mreže, s ograničenim brojem korisnika, od kojih je svaki radio za dobrobit cjeline. Posljedica toga je da su svi protokoli, od najnižih slojeva, do aplikacijskog sloja, bili dizajnirani s malom ili nikakvom zaštitom. Zbog toga je danas jednostavno sprovesti cijeli niz malicioznih radnji kao što su lažiranje IP adresa ili slanje neželjene elektroničke pošte. Upravo zbog jednostavnosti provedbe i zbog uglavnom ekonomskih motiva, danas na Internetu nalazimo velike količine neželjenog prometa. Neželjeni promet je sav onaj promet koji troši mrežne i računalne resurse na način koji ne koristi vlasnicima tih resursa [19]. Internet Architecture Board (*IAB*) je na radionici o neželjenom prometu kao jedan od glavnih uzroka neželjenog prometa naveo crnu ekonomiju [19]. Primjeri neželjenog prometa su DoS i DDoS napadi, neželjena elektronička pošta, virusi, crvi i sl. Svojim intenzitetom i količnom generiranog neželjenog prometa, posebno se ističu napadi s uskraćivanjem usluge (*engl. Denial of Service, u daljnjem tekstu DoS napadi*). DoS napadi su napadi kod kojih jedan ili više napadača nastoji žrtvi uskratiti pristup do usluge za koju je žrtva legalni korisnik. Žrtva može biti klijent, poslužitelj, mrežna oprema (npr. usmjernik, pristupna točka, mrežna poveznica), pojedinačni korisnik Interneta, kompanija koja posluje koristeći Internet, pružatelj Internet usluga *(engl. Internet Service Provider*) ili bilo koja kombinacija navedenih žrtava. RFC 4732 se bavi problematikom DoS napada [15].

DoS napadi su vrlo jednostavni za provesti. Na Internetu je dostupan velik broj alata koji služe upravo za pokretanje DoS napada, neki poznatiji su: *TFN* [7], *stacheldraht* [8] i *trinoo* [9]. S druge strane protiv DoS napada se je teško obraniti i još uvijek ne postoji adekvatan mehanizam zaštite.

Cilj ovog rada bio je izgraditi reputacijski sustav koji će određivati ocjene reputacije pojedinih autonomnih sustava na temelju praćenja neželjenog prometa. To je ostvareno na način da se najprije mjerila vrsta i količina neželjenog prometa pristigloga iz pojedinih autonomnih sustava. Neželjen promet se većinom mjerio na mrežnom i prijenosnom sloju, a u radu se poseban naglasak pridaje napadima s uskraćivanjem usluge. Neki oblici neželjenog prometa su već mjereni u okviru sličnih radova, neželjena elektronička pošta [28] i pogreške unutar DNS sustava [27], te oni nisu u središtu istraživanja ovog rada. Nakon što je izmjeren neželjen promet pristigao iz pojedinog autonomnog sustava, reputacijski sustav uz pomoć odgovarajuće reputacijske funkcije izračunava ocjenu reputacije.

Rad je podijeljen u sedam poglavlja. U drugom poglavlju su obrađene dvije osnovne klasifikacije DoS napada te su navedeni i opisani primjeri za svaku od klasa napada. Zatim su opisane tehnike detekcije DoS napada temeljene na nadgledanju u jezgri i na rubovima mreže. Naposljetku su navedene neke od metoda zaštite od DoS napada. Opisana je metoda implementacije ulaznog i izlaznog filtriranja. Usmjernici takvim filtriranjem mogu prepoznati dio nelegitimnog prometa te ga odbaciti i na taj način i eventualno spriječiti DoS napad. Slijedi pregled predloženih metoda za identifikaciju stvarnog izvora bilo kojeg paketa poslanog s Interneta (*metoda praćenja do izvorišta*). Također je opisan princip funkcioniranja svake od navedenih tehnika praćenja do izvorišta.

Treće poglavlje opisuje različite scenarije DoS napada te informacije koje su dostupne žrtvi tijekom napada. Nadalje, definiran je pojam neželjenog prometa te su navedene metode detekcije neželjenog prometa.

U četvrtom poglavlju je prikazan način mjerenja neželjenog prometa koji se koristio u okviru ovog rada. Opisani su sustavi za otkrivanje napada općenito, te Snort - besplatni sustav za otkrivanje napada, otvorenog koda. Zatim je objašnjen način pisanja pravila za Snort te Stream5, pretprocesor Snorta koji se koristio u okviru ovog rada. Naveden je i način podešavanja i pokretanja Snort sustava koji se koristio tijekom mjerenja neželjenog prometa u praktičnom dijelu ovog rada.

Peto poglavlje govori o implementaciji reputacijskog sustava. Objašnjenja je podjela i organizacija programskog kôda po modulima te značenje pojedinih klasa i njihovih metoda. Zatim je objašnjen način računanja reputacije i funkcija koja se pri tom koristila. Objašnjenje su i dvije težinske funkcije koje su se koristile unutar osnovne funkcije za izračunavanje reputacije. Slijedi pregled svih datoteka koje se nalaze u praktičnom dijelu ovog rada, te je objašnjena uloga svake od datoteka. Na kraju poglavlja je primjer pokretanja reputacijskog sustava, te objašnjeni izlazni podaci koji su dobiveni.

Šesto poglavlje prikazuje rezultate dobivene mjerenjem u praktičnom dijelu ovog rada. Analizirana je dnevnička datoteka s alarmima, te je prikazan postotni udio pojedinih alarma u ukupnom neželjenom prometu. Objašnjen je i problem mogućih lažnih alarma. Zatim su grafički prikazane reputacije tri najgora autonomna sustava, te njihove promijene u vremenu. Reputacije su prikazane za obje težinske funkcije. U nastavku je prikazano 20 najgorih autonomnih sustava po broj zabilježenih alarma te po srednjim vrijednostima obiju težinskih funkcija. Zatim su dobiveni rezultati uspoređeni sa rezultatima sličnog rada u kojem se reputacija određivala na temelju praćenja DNS sustava [27].

Rad završava zaključkom i popisom literature.

2. Općenito o DoS napadima

2.1. Klasifikacija DoS napada

U ovom poglavlju su objašnjene dvije osnovne klasifikacije DoS napada:

1. Klasifikacija prema izvorištu napada
2. Klasifikacija prema tehnikama napada

DoS napadi se, prema izvorištu napada, klasificiraju kao:

- napadi koji dolaze iz jednog izvora *(engl. single source*)
- napadi koji dolaze iz više izvora (*engl. multi source*)

Najjednostavniji primjer napada koji dolaze iz jednog izvora je *preplavljivanje ICMP paketima* (*engl. ICMP flooding*). To je napad pri kojem napadač šalje žrtvi veliku količinu ICMP paketa. Žrtvina propusnost (*engl. bandwidth*) se na taj način istroši te legitimni paketi nisu u mogućnosti stići do odredišta.

Još jedan primjer napada koji dolazi iz jednog izvora je *Teardrop napad*. Teardrop koristi grešku (*engl. bug*) u fragmentaciji i sakupljanju IP paketa kod nekih operacijskih sustava. Napadač šalje žrtvi preklapajuće fragmente paketa. To može uzrokovati rušenje operacijskog sustava iz razloga što programski kod za ponovno sakupljanje fragmenata ne zna na pravilan način rukovati fragmentima koji se međusobno preklapaju. Operacijski sustavi Windows 3.1x, Windows 95, Windows NT kao i operacijski sustav Linux s verzijama jezgre 2.0.32 i 2.1.63 i ranijim su bili ranjivi na takvu vrstu napada.

Koordinirani napadi iz više izvora se još nazivaju i raspodijeljeni DoS napadi (*engl. Distributed Denial of Service, u daljnjem tekstu DDoS*). DDoS napadi se mogu realizirati korištenjem *Botneta* - mreže tzv. botova. Bot je programski realiziran robot koji se najčešće instalira pomoću crva (*engl.* worm), Trojanskog konja (*engl. Trojan horse*) ili stražnjih vrata (*engl. backdoor*). Botovi su pod nadzorom napadača te napadač može pokrenuti DDoS napad prema odabranoj žrtvi. Intenzitet napada ovisi o količini botova koji sudjeluju u napadu. Svaki od botova je poseban izvor napada.

Drugi primjer ove vrste napada su Smurf napadi (engl. smurf attack) kod kojih napadač šalje veliku količinu IP paketa s lažiranom (engl. spoofed) izvorišnom adresom, postavljenom na adresu žrtve. Odredišna adresa u takvim, lažiranim, paketima je postavljena na difuznu adresu (engl. broadcast address) . Na taj će način sva računala unutar podmreže odgovoriti žrtvi te ju preplaviti s velikom količinom paketa.

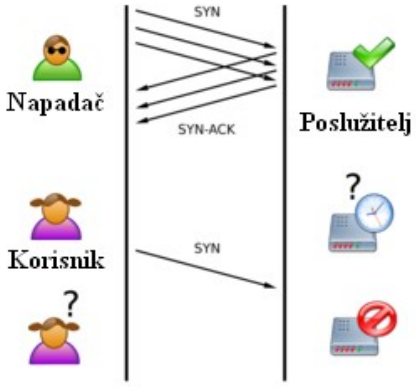
DoS napadi se prema tehnikama izvođenja klasificiraju kao:

- napadi potrošnjom ograničenih resursa
- napadi uništavanjem ili izmjenom konfiguracijskih informacija
- napadi pomoću programskih grešaka (*engl. bug*)
- napadi fizičkim uništavanjem ili izmjenom mrežnih komponenti

Napadi potrošnjom ograničenih resursa su napadi kod kojih napadač nastoji žrtvi iscrpiti jedan ili više resursa. Neki od ograničenih resursa su: raspoloživa memorija, ciklusi procesora, raspoloživi prostor za pohranu podataka, propusnost, broj procesa ili dretvi, simultani broj veza neke aplikacije.

Primjer potrošnje resursa su *reflektirajući napadi*. Oni uključuju slanje paketa s lažiranim adresama do jednog ili više računala koja imaju veliku propusnost (najčešće neka vrsta poslužitelja). Izvorišna adresa u lažiranim paketima je postavljena na adresu žrtve pa će računala s velikom propusnosti odgovoriti žrtvi. Paketi odgovora računala s velikom propusnosti su veći od paketa upita koje šalje napadač. Žrtvina propusnost će na taj način biti potrošena (preplavljena).

Još jedan primjer ove vrste napada je *SYN preplavljivanje*. To je napad na poslužitelj koji koristi TCP. Napad je shematski prikazan na slici 2.1.



Slika 2.1: SYN preplavljivanje

Napadač šalje velik broj SYN paketa poslužitelju, ali ne odgovara na SYN-ACK poruke poslužitelja. Operacijski sustav poslužitelja instancira stanje poluotvorene veze za svaki primljeni SYN paket. Broj poluotvorenih veza je ograničen te ako napadač može dovoljno brzo slati SYN pakete onda će poslužitelj u jednom trenutku dosegnuti maksimalan broj poluotvorenih veza. Legalni korisnici tada neće biti u stanju uspostaviti komunikaciju s poslužiteljem.

Nepravilno podešeno računalo može pogrešno izvršavati svoje zadaće ili ih uopće ne izvršavati. Napadač može izmijeniti ili obrisati konfiguracijske informacije i na taj način spriječiti žrtvu u korištenju neke usluge. Primjerice, ukoliko napadač uspije izmijeniti tablice usmjeravanja na žrtvinim usmjernicima, žrtvina mreža može biti onesposobljena. Ako je napadač u mogućnosti modificirati registry na Windows NT računalu žrtve, određene funkcije mogu biti nedostupne.

Programska greška *(engl. bug)* je greška, propust, zastoji ili kvar u računalnom programu ili sustavu. Posljedica takve greške je nepredvidljivo ponašanje programa koje često može rezultirati da neke usluge ne budu dostupne korisniku. Programske greške mogu biti u samom operacijskom sustavu, u komunikacijskim protokolima ili u računalnim programima. Primjer programske greške je Teardrop napad.

Sve mrežne komponente moraju biti zaštićene od vanjskih fizičkih utjecaja ili se u protivnom može desiti da njihove usluge budu nedostupne. Vanjski utjecaji obuhvaćaju:

- mehaničko oštećenje naprava
- oštećenje nastalo požarom ili poplavom

- krađu uređaja

Fizička zaštita ne ulazi u domenu ovog seminarskog rada te neće biti posebno obrađivana.

2.2. Detekcija DoS napada

Nadgledanjem mreže moguće je otkriti DoS napade koji se koriste tehnikom potrošnje ograničenih resursa. Napade je moguće otkriti u ranoj fazi, dok još nisu ozbiljnije naškodili žrtvi. Napadi potrošnje resursa najčešće ubacuju velike količine prometa u mrežu te na taj način mijenjaju neke unutarnje parametre mreže (npr. kašnjenje paketa, postotak izgubljenih paketa). Nadgledanjem tih parametara pratimo promjene na mreži te možemo identificirati zakrčene veze i ulazne točke kroz koje dolazi dolazi velika količina prometa. Nadgledanjem mreže mjerimo tri parametra:

1. **Kašnjenje paketa** - latencija paketa od kraja do kraja
2. **Omjer izgubljenih paketa** - omjer broja odbačenih paketa iz toka podataka u odnosu na ukupan broj paketa u toku podataka
3. **Propusnost** - propusnost toka podataka unutar mreže

Kašnjenje i omjer izgubljenih paketa su dobra indikacija stanja na mreži. Ako je stanje na mreži normalno, tokovi ne bi trebali imati veliko kašnjenje niti omjer izgubljenih paketa. Napomenimo da je i variranje u kašnjenju dobra indikacija stanja na mreži no ono je zasebno za svaki tok podataka te nije pogodno za nadgledanje.

Navedeni parametri mogu biti mjereni uz pomoć unutarnjih (jezgrenih) usmjernika ili bez njihove pomoći. U daljnjem tekstu opisujemo nadgledanje u jezgri mreže (*engl. core based monitoring*) i nadgledanje na rubovima mreže (*engl. edge based monitoring*) te detekciju DoS napada.

2.2.1. Nadgledanje u jezgri mreže

Problematika nadgledanja u jezgri mreže je opisana u [2]. Kašnjenje paketa se mjeri tako što ulazni usmjernici kopiraju zaglavlja proizvoljno odabranih ulaznih paketa. Kopiranje ovisi o unaprijed podešenom parametru vjerojatnosti kopiranja. Ulazni usmjernik šalje probni paket sa istim zaglavljem kao i paket sa podacima, što znači da će probni paket vjerojatno proći istim putem kao i paket s podacima. Izlazni usmjernici prepoznaju probne pakete i računaju kašnjenje.

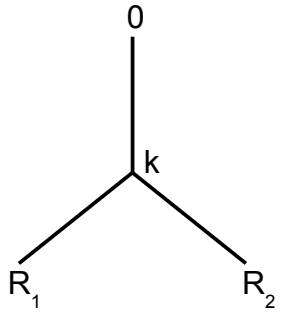
Omjer izgubljenih paketa se mjeri brojanjem odbačenih paketa u jezgrenim usmjernicima. Tada se kontaktiraju ulazni usmjernici da bi se dobio ukupan broj paketa za svaki tok. Na temelju ta dva broja računamo omjer. Ukoliko omjer izgubljenih paketa prekorači neku graničnu vrijednost to je indikacija zakrčene veze.

Za mjerenje propusnosti se koriste izlazni usmjernici. Izlazni usmjernici mogu ustanoviti propusnost za svaki tok.

U ovoj shemi se jezgreni usmjernici koriste jedino za brojanje odbačenih paketa, dok se kašnjenje paketa i propusnost računa bez pomoći jezgrenih usmjernika. Takva shema nadgledanja ipak pretjerano opterećuje jezgrene usmjernike i iz tog razloga nije skalabilna.

2.2.2. Nadgledanje na rubovima mreže

Opisat ćemo dvije sheme nadgledanja temeljenog na rubovima: nadgledanje temeljeno na prugama (*engl. stripe-based*) i raspodijeljeno nadgledanje (*engl. distributed*). Obje sheme mjere kašnjenje i propusnost na isti način kao i nadgledanje u jezgri mreže. Razlikuju se u mjerenju omjera izgubljenih paketa.



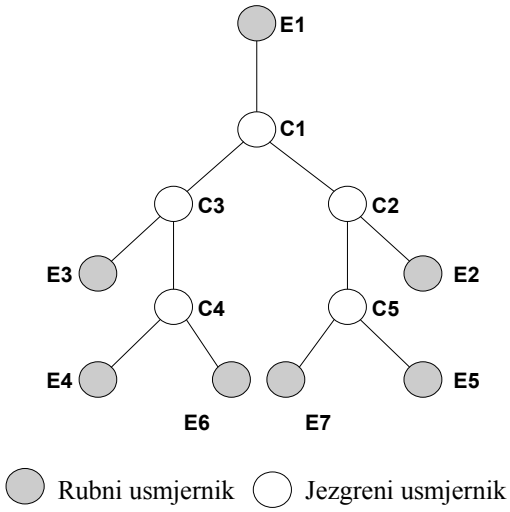
Shema 2.1. Računanje omjera izgubljenih paketa od izvorišta 0 do odredišta R_1 i R_2.

Nadgledanje temeljeno na prugama računa omjer izgubljenih paketa bez oslanjanja na jezgrene usmjernike. Shema koristi slanje nekoliko uzastopnih paketa, nazvanih pruga, bez kašnjenja među njima. Najčešće se jedna pruga sastoji od tri uzastopna paketa. Pruge se šalju od jednog rubnog usmjernika do dva druga rubna usmjernika. Radi jednostavnosti, razmotrimo primjer sa samo jednim jezgrenim i tri rubna usmjernika. Primjer je ilustriran shemom 2.1. Usmjernici formiraju binarno stablo s dva lista.

Stablo sadrži čvorove *0*, *k*, R_1 i R_2. Čvorovi *0*, R_1 i R_2 predstavljaju rubne usmjernike, a čvor *k* je jezgreni usmjernik. Omjer izgubljenih paketa za vezu $k \rightarrow R_1$ se može procijeniti slanjem pruge od korijena *0* do čvorova R_1 i R_2. Prvi paket se šalje od čvora *0* do čvora R_1 dok se zadnja dva šalju do čvora R_2. Ako paket stigne do bilo kojeg primatelja možemo zaključiti da je paket stigao i do čvora grananja *k*. Nadalje, ako R_2 primi oba paketa vjerojatno je da će i R_1 primiti prvi paket. Vjerojatnost da je paket izgubljen se računa na temelju podataka o tome da li su svi paketi poslani do R_1 i R_2 stigli na odredište. Na sličan način je moguće zaključiti i omjer izgubljenih paketa za vezu $k \rightarrow R_2$. Šaljemo komplementarnu prugu u kojoj je prvi paket poslan do R_2, a druga dva do R_1. Omjer izgubljenih paketa zajedničkog puta od $0 \rightarrow k$ se može procijeniti kombiniranjem rezultata iz prva dva koraka. Ova tehnika zaključivanja se može proširiti i na općenita stabla slanjem pruga od korijena do svih uređenih parova listova stabala. Mjerenje omjera izgubljenih paketa na ovakav način je skalabilno [25].

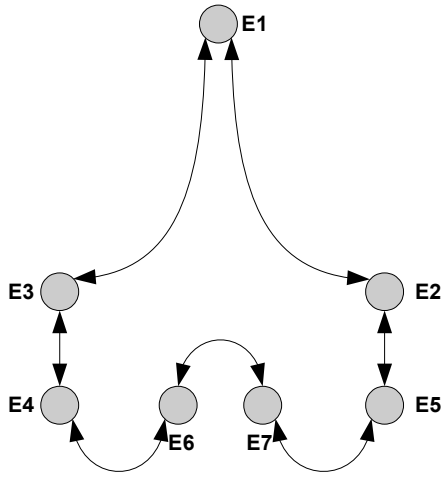
Nadgledanje temeljeno na prugama je usvojeno i za nadgledanja gubitaka unutar mreža s podrškom kvalitete usluge (*engl. Quality of service - QoS*), bez oslanjanja na jezgrene usmjernike [2]. Nadgledanje temeljeno na prugama stvara manje opterećenje (*engl. overhead*) od nadgledanja u jezgri mreže [1].

Raspodijeljeno nadgledanje je predloženo u [1] kako bi se dalje smanjilo opterećenje. Rubni usmjernici domene tvore prekrivajuću mrežu (*engl. overlay network*) povrh fizičke mreže. Shema 2.2 pokazuje razapinjuće stablo topologije domene. Rubni usmjernici su na shemi prikazani sivom bojom te označeni slovom **E** (*engl. edge*) i pripadajućim rednim brojem (**E1 – E7**). Jezgreni umsjernici su prikazani bijelom bojom i označeni slovom **C** (*engl. core*) i pripadnim rednim brojem. (**C1 – C5**).



Shema 2.2. Razapinjuće stablo topologije domene

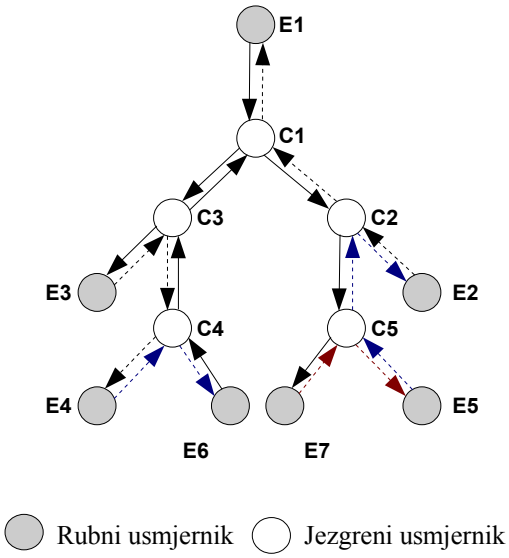
Rubni usmjernici međusobno tvore prekrivajuću mrežu. Prekrivanje se koristi za izgradnju tunela za probne pakete.



Shema 2.3. Prekrivajuća mreža rubnih usmjernika

Prekrivajuća mreža je prikazana shemom 2.3. Shema prikazuje rubne usmjernike **E1 – E7** te tunele između njih, prikazane linijama sa strelicama. Svaki rubni usmjernik ima izgrađen tunel do dva susjedna rubna usmjernika. Tako primjerice usmjernik **E1** ima tunele do usmjernika **E2** i **E3**.

Unutarnje veze za svaki put od kraja do kraja u prekrivajućoj mreži su prikazane shemom 2.4. Veze su na shemi prikazane različitim linijama sa strelicama (isprekidana ili puna linija). Tako je primjerice usmjernik **E7** povezan isprekidanom linijom (unutranjim vezama **E7 – C5** i **C5 – E5)** s usmjernikom **E5** koji je njegov susjedni usmjernik u prekirvajućoj mreži.



Shema 2.4. Smjer slanja probnih paketa u unutarnjim vezama

Kod raspodijeljenog nadgledanja se sva tri parametra (kašnjenje, propusnost i gubitak paketa) mjere na svim rubnim usmjernicima. Kašnjenje i propusnost se mjere na isti načina kao i kod nadgledanja temeljenog na prugama. Razlika je u mjerenju korištenom za računanje omjera izgubljenih paketa. Ovo shema se oslanja na činjenicu da nije potrebno znati točnu vrijednost omjera izgubljenih paketa nego je dovoljno znati samo da li je ta vrijednost veća od nekog praga (*engl. treshold*) ili ne. Vezu s velikim gubitkom paketa, odnosno vezu u kojoj je omjer izgubljenih paketa prešao prag, nazivamo zakrčena veza. Cilj raspodijeljenog nadgledanja je uočiti sve zakrčene veze.

Kada kašnjenje paketa postane veliko uključuju se agenti na različitim rubnim usmjernicima da ispitaju gubitak paketa. Agenti su programi kojima je zadaća ispitati stanje veza (zakrčena ili ne). Svaki rubni usmjernik ispituje svoje susjedne usmjernike. Neka je X_ρ booleova varijabla koja predstavlja izlaz probe ρ. X_ρ poprima vrijednost 1 ako je izmjeren omjer izgubljenih paketa veći od neke granične vrijednosti na bilo kojoj vezi duž probanog puta, a u suprotnom poprima vrijednost 0. Na primjer, ako je vrijednost X_ρ za put *E1→E3* 1 to znači da je veza *E1→C1*, *C1→C3*, *C3→E3* ili bilo koja kombinacija navedenih veza zakrčena. Ako je vrijednost varijable X_ρ 0 onda možemo zaključiti da niti jedna od veza nije zakrčena. Na taj način pišemo izraze koji izražavaju stanja unutarnjih veza. Izračunavanje tih izraza i identificiranje zakrčenih veza je detaljno objašnjeno u [1]. Raspodijeljeno nadgledanje zahtjeva manji broj ukupnih proba, $O(n)$, u usporedbi s nadgledanjem temeljenim na prugama, koje zahtjeva $O(n^2)$, gdje je n broj rubnih usmjernika.

Da bismo detektirali DoS napad moramo identificirati skup zakrčenih veza L jednom od metoda nadgledanja navedenih u prethodnom tekstu. Za svaku zakrčenu vezu, $(v_i, v_j) \in L$, stablo je podijeljno u dva podstabla: jedno tvore listovi potomci čvora v_i i drugo listovi potomci čvora v_j. Prvo podstablo ima za listove izlazne usmjernike kroz koje tokovi podataka napuštaju mrežu. Ako puno izlaznih tokova podataka ima istu odredišnu IP adresu, možemo zaključiti da se radi ili o DoS napadu ili promet ide do primjerice popularne stranice. Odluku o tome da li se radi o DoS napadu ili o prometu legitimnih korisnika možemo dati na način da se konzultiramo s odredištem. Ako se radi o DoS napadu možemo ga zaustaviti okidanjem filtra na ulaznim usmjernicima koji tvore listove drugog podstabla (potomci od čvora v_j).

Razmotrimo primjer DoS napada na mreži čija je topologija prikazana shemom 2.2. Neka je žrtvina domena *D* spojena na krajnji usmjernik *E6*. Nadgledanjem smo ustanovili da su veze *C3→C4* i *C4→E6* zakrčene. Za obje zakrčene veze većina tokova podataka izlazi kroz usmjernik *E6*. Utvrđivanjem odredišnog IP prefiksa, na usmjerniku *E6*, određujemo da je velika količina prometa upućena prema domeni *D.* Da bismo mogli zaustaviti napad moramo utvrditi kroz koje ulazne usmjernike tokovi podataka ulaze u domenu. Algoritam za određivanje ulaznih usmjernika je objašnjen u [1]. Nakon što su ustanovljeni ulazni usmjernici potrebno je na njima uključiti filtriranje.

Prednost ovakve detekcije temeljene na nadgledanju je u tome što susjedne žrtvine domene mogu u ranoj fazi primijetiti napad. Konzultiranjem s potencijalnom žrtvom moguće je drastično smanjiti intenzitet napada ili ga, u slučaju rane detekcije, u potpunosti spriječiti.

2.3. Zaštita od DoS napada

U nastavku su opisne neke tehnike zaštite od DoS napada. Implementacija ulaznog i izlaznog filtriranja paketa na usmjernicima suzbija DoS i DDoS napade s lažiranim IP adresama. Tehnike praćenja (*engl. traceback*) pak nastoje utvrditi stvarni izvor paketa te na taj način locirati od kuda dolazi maliciozni promet DoS napada. Dok su ulazno i izlazno filtriranje široko primijenjeni, niti jedna od metoda praćenja nije rasprostranjena na Internetu, odnosno praktično primjenjiva.

2.3.1. Unicast reverse path forwarding (uRPF)

Unicast reverse path forwarding (u daljnjem tekstu uRPF) je tehnika razvijena kako bi se implementiralo ulazo filtriranje (*engl. Ingress filtering*) [6]. Originalna ideja je bila da se blokira promet sa sučelja ako je izvorišna adresa postavljena na privatnu adresu. Ta je ideja kasnije proširena korištenjem znanja usmjernika iz tablice usmjeravanja (*engl. Routing Information Base - RIB*) i tablice prosljeđivanja (*engl. Forwarding Information Base – FIB*) kako bi se uvele daljnje restrikcije na izvorišne adrese. Paketi se prosljeđuju samo ako su došli s najbolje staze usmjernika do izvorišta paketa. Na taj je način osigurano da:

1. Paketi koji dolaze na sučelje dolaze od potencijalno ispravnih računala (*engl. hosts*)
2. Paketi s izvorišnom adresom koja se ne može dohvatiti preko ulaznog sučelja budu odbačeni bez ometanja normalnog rada s obzirom da takvi paketi vjerojatno dolaze od krivo podešenih ili malicioznih izvorišta

U slučajevima simetričnog usmjeravanja i terminalnih mreža sa samo jednom poveznicom to je sigurna pretpostavka i uRPF može biti implementiran bez straha od mogućih problema. Iz tog je razloga izuzetno korisno implementirati uRPF kad god je zajamčeno simetrično usmjeravanje. Korištenje uRPF-a što je moguće bliže izvorištu prometa također zaustavlja lažirani promet prije nego je imao priliku potrošiti dio propusnosti Interneta ili prije nego što je naišao na usmjernik koji nema podešen uRPF.

Ipak često je slučaj da usmjeravanje nije simetrično i ne može se računati da će usmjernik pokazati put do izvorišta paketa preko sučelja kojeg je paket i došao. Tablice usmjeravanja specificiraju najbolji put za prosljeđivanje i samo u slučajevima simetričnih staza je to ujedno i najbolji povratni put. Iz tog je razloga prilikom implementiranja uRPF-a potrebno biti svjestan tog problema kako ne bi spriječili legitimni promet. Bilo koji usmjernik koji koristi podrazumijevanu (*engl. default*) stazu ne može koristiti uRPF na sučelju na koje pokazuje podrazumijevana jer će zbog podrazumijevane staze svi paketi biti proslijeđeni te neće biti postignuto filtriranje.

uRPF se može realizirati na dva načina: striktni način (*engl. strict mode*) i slobodni način (*engl. loose mode*). U striktnom načinu rada svaki dolazni paket se provjerava u tablici prosljeđivanja i ako sučelje s kojeg je paket došao nije ujedno i sučelje za najbolji povratni put onda paket nije prošao provjeru. Paketi koji nisu prošli provjeru se odbacuju. U slobodnom načinu rada se izvorišne adrese dolaznih paketa također provjeravaju u tablici prosljeđivanja i ako izvoršna adresa nije dohvatljiva preko sučelja onda paket nije prošao provjeru.

Također spomenimo da je za prevenciju DoS napada osim spomenutog ulaznog filtriranja od velike važnosti i izlazno filtriranje (*engl. egress filtering*). Izlazno filtriranje kontrolira promet koji izlazi iz mreže. Dva su posebno važna pravila koje je potrebno osigurati na graničnim usmjernicima i sigurnosnim stijenama (*firewall*):

1. Promet koji za izvorišnu IP adresu ima postavljenu adresu koja se ne nalazi unutar adresnog prostora naše mreže treba odbaciti.
2. Promet koji za izvorišnu IP adresu ima postavljenu privatnu adresu također treba odbaciti.

Ostvarivanjem ovih jednostavnih pravila će se u velikoj mjeri suzbiti napade koji uključuju lažiranje izvorišne IP adrese.

2.3.2. Praćenje do izvorišta

Praćenje do izvorišta (*engl. traceback*) je proces identificiranja pravog izvora bilo kojeg paketa poslanog preko Interneta. Većina pristupa praćenju je motivirana DoS i DDoS napadima, ali generalno nije limitirana samo na njih. Zadaća identificiranja pravog izvora paketa je kompliciranа zbog činjenice da se IP adrese lako mogu lažirati. U takvim slučajevima konvencionalne metode određivanja lokacije sustava uz pomoć IP adrese (*traceroute*) ne rade zbog činjenice da je izvorišna IP adresa lažirana.

Napomenimo da identificiranje izvora malicioznih paketa ne znači nužno da smo identificirali i napadača (primjer reflektirajućih napada ili botnetova). Također bitno je reći da tehnike praćenja do izvorišta niti sprječavaju niti zaustavljaju napade, one se koriste isključivo za identifikaciju izvora paketa.

Sheme praćenja do izvorišta možemo svrstati u jednu od četiri kategorija [10]:

1. Pohrana na krajnjim računalima *(engl. End-host storage*)
    - Vjerojatnosno označavanje paketa (*engl. Probabilistic packet marking*)
    - ICMP praćenje (*engl. iTrace*)
2. Bilježenje paketa (*engl. Packet logging*)
    - IP praćenje temeljeno na sažetku (*engl. Hash-based IP traceback*)
3. Specijalizirano usmjeravanje (*engl. Specialized routing*)
    - Prekrivajuća mreža (*engl. Overlay network*)
    - IP praćenje uz pomoć protokola IPSec
4. Zaključivanje na temelju stanja mreže
    - Kontrolirano preplavljivanje (*engl. Controlled flooding*)

Vjerojatnosno označavanje paketa je zasnovano na ideji da usmjernici označavaju pakete koji prolaze kroz njih s njihovom adresom ili dijelom njihove adrese. Paketi koji se označavaju se odabiru nasumično s nekom fiksnom vjerojatnosti da će biti označeni. Žrtva dobiva označene pakete od usmjernika i može rekonstruirati dio puta ili cijeli put do izvora određnih paketa. Označavanje se vrši na način da se dio ili cijela adresa usmjernika zapiše u zaglavlje paketa (ID polje i Reserved Flag polje).

Kod ICMP praćenja svaki usmjernik se podešava na takav način da s određenom vjerojatnošću odabere paket (preporuča se 1 na svakih 20,000 paketa) i stvori ICMP poruku praćenja, tzv. iTrace poruku, upućenu istom odredištu kao i izabrani paket. Za razliku od vjerojatnosnog označavanja kod kojeg se postojeći paketi označavaju, kod ICMP praćenja se generiraju posebni ICMP paketi. iTrace poruka se sastoji od slijedećeg i prijašnjeg skoka i vremenske oznake. Također što je više moguće okteta (*bytes*) izabrane poruke je kopirano u podatkovno polje iTrace poruke. TTL polje se postavlja na 255 i služi da bi se identificirao pravi put paketa.

IP praćenje temeljeno na sažetku je shema u kojoj svaki usmjernik bilježi dijelove svih paketa koji prolaze kroz njega kako bi kasnije bio u mogućnosti odrediti da li je neki paket prošao kroz njega. Takvi usmjernici se, u ovoj shemi, nazivaju agenti za stvaranje podataka (*engl. Data Generation Agents*). Jedinica za upravljanje praćenjem (*engl. Traceback Management Unit*) je centralizirana jedinica koja je u mogućnosti slati upite agentima za stvaranje podataka. Jedinica za upravljanje također komunicira i sa žrtvinim sustavom za otkrivanje napada (*engl. Intrusion Detection System – IDS*) te je u mogućnosti, ukoliko žrtva zatraži, rekonstruirati put dolaska određenog paketa.

Shema prekrivajuće mreže problem rješava uz pomoć usmjernika za praćenje (*engl. Tracking Router, u daljnjem tekstu TR*). TR nadgleda sav promet koji prolazi kroz mrežu. Da bi mogao nadgledati sav promet na mreži, svi paketi moraju proći kroz TR. To se postiže gradnjom tunela od svakog krajnjeg (*edge*) usmjernika do TR. Na taj način sav promet od ulaznih usmjernika putuje putem tunela do TR i od TR nazad putem drugih tunela do izlaznih usmjernika. Dok jezgreni usmjernici prenose promet, svaki je krajnji usmjernik samo jedan skok udaljen od TR. Ovakva arhitektura rezultira zvjezdastom topologijom, gdje je TR u središtu zvijezde. Kako su tuneli sagrađeni preko postojeće topologije mreže, nova zvjezdasta mreža tvori prekrivajuću mrežu pa od tuda i naziv ove sheme.

Kada se primijeti napad, izvor napada može lako biti identificiran zato što je samo jedan skok udaljen od TR.

IP praćenje uz pomoć protokola IPSec je mehanizam temeljen na pretpostavci da je kompletna mrežna topologija poznata sustavu. Shema radi po principu: Ako postoji IPSec sigurnosno udruživanje (*engl. security association*) između proizvoljnog usmjernika i žrtve i maliciozni paketi su primijećeni i autentificirani od strane tog udruživanja onda je napad potekao od nekog uređaja daljeg nego što je usmjernik iz asocijacije. Ako paketi nisu autentificirani od strane udruživanja onda je izvor paketa usmjernik između tog usmjernika i žrtve. Uspostavljanjem takvih sigurnosnih udruživanja moguće je identificirati jedan usmjernik ili grupu usmjernika od kojih potječu paketi.

Kontrolirano preplavljivanje (*engl. Controlled flooding*) je metoda koja je ispravna samo za DoS napade. Oslanja se na činjenicu da su veze na napadnutom putu jako opterećene. Pažljivim mjerenjem dolaznog prometa u napadnuti sustav i ispitivanjem opterećenja veza za koje sumnjamo da su na putu napada, mjerimo omjer odbačenih paketa te zaključujemo preko koje veze dolazi napad. Proces se ponavlja za slijedeći skok sve dok ne identificiramo izvor napada.

3. Detekcija neželjenog prometa

Cilj ovog rada je izgraditi reputacijski sustav koji će ocjenjivati druge autonomne sustave. Ocjena reputacije se određuje na temelju količine neželjenog prometa pristiglog iz pojedinih autonomnih sustava. Manja reputacija označava sustave iz kojih pristiže veća količina neželjenog prometa. Svaki novi sustav dobiva inicijalnu reputaciju koja iznosi nula. Sistemom kažnjavanja i nagrađivanja ocjena reputacije AS-a će se tijekom vremena korigirati, te može poprimiti veće ili manje vrijednosti od inicijalne. Po količini generiranog neželjnog prometa se posebno ističu DoS napadi. Kako bismo mogli vršiti korekcije ocjene reputacije u nastavku najprije slijedi pregled različitih scenarija DoS napada te informacija koje su nam dostupne prilikom svakog od napada. Zatim su u poglavlju 3.2. objašnjeni ostali oblici neželjenog prometa.

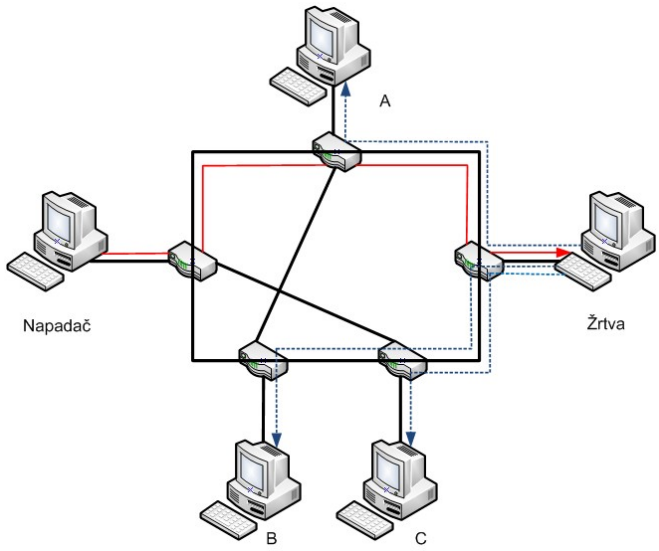
3.1. Scenariji DoS napada

3.1.1.Napadi koji dolaze iz jednog izvorišta s lažiranom izvorišnom adresom

Napadači često lažiraju izvorišne IP adrese kako bi prikrili svoju stvarnu lokaciju. Većina programa za DoS napade postavlja nasumično izvorišnu IP adresu za svaki poslani paket [13]. Ti programi uključuju većinu najpopularnijih DoS napadačkih alata: *Shaft*, *TFN*, *TFN2k*, *trinoo*, sve varijacije *Stacheldrahta*, *mstream* i *Trinity*. Kada lažirani paket stigne do žrtve, žrtva obično šalje odgovor i taj odgovor pristiže na lažiranu IP adresu. Povremeno i mrežni uređaj (npr. usmjernik ili sigurnosna stijena) na putu od napadača do žrtve može poslati svoj vlastiti odgovor putem ICMP poruke [14]. Takve ICMP poruke su ponovno poslane na nasumično lažirane izvorišne adrese.

Iz razloga što napadač bira izvorišne adrese nasumično, žrtvini odgovori će biti podjednako raspodijeljeni po cijelom adresnom prostoru Interneta.

Shema 3.1 prikazuje scenarij napada iz jednog izvorišta u kojem napadač nasumično lažira izvorišne adrese.

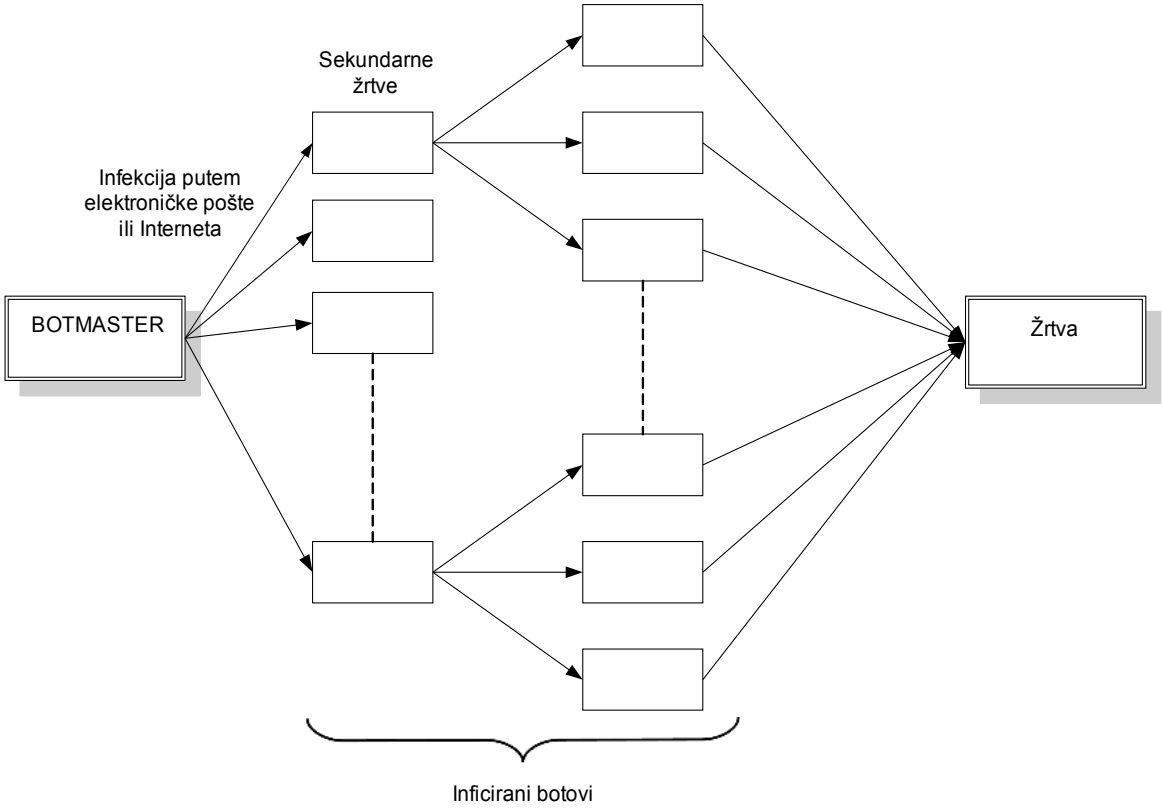


Shema 3.1: Napad iz jednog izvorišta s nasumično lažiranom izvorišnom adresom

Veze između pojedinih usmjernika, i veze između usmjernika i korisnika su prikazane punom debljom linijom. Promet napadača je prikazan punom usmjerenom linijom, a odgovori žrtve su prikazani isprekidanom usmjerenom linijom. Svaki korisnik se nalazi u zasebnom autonomnom sustavu. Napadač je poslao tri paketa s IP adresama nasumično postavljenim na adrese računala **A**, **B** i **C**. Žrtva šalje odgovore do ta tri računala. Žrtvin autonomni sustav nije u stanju locirati autonomni sustav napadača, nego je u stanju locirati jedino susjedni autonomni sustav iz kojeg dolazi promet napadača. Tako u primjeru prikazanom shemom 3.2, žrtvin autonomni sustav detektira da sam promet napada dolazi sa sučelja na kojem se nalazi autonomni sustav **A**, ali nije u stanju odrediti da li se napadač nalazi u autonomnom sustavu **A** ili u nekom daljem autonomnom sustavu.

3.1.2. Napadi koji dolaze iz više izvorišta s lažiranim izvorišnim adresama

Najrasprostranjeniji primjer ove vrste napada je napad korištenjem mreže kompromitiranih računala - *botneta*. Struktura botnet mreže je prikazana na shemi 3.2.



Shema 3.2: Struktura botnet mreže

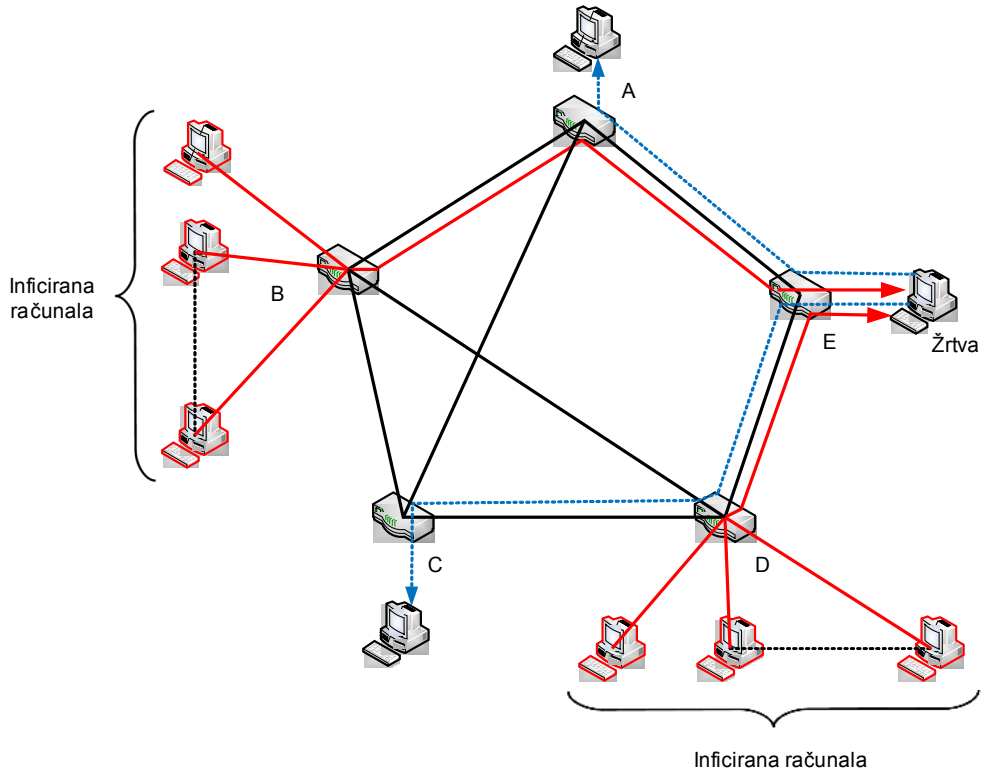
Kompromitirana računala su pod kontrolom jednog operatora, tzv. *botmastera*. Botmaster na početku šalje maliciozni kod da bi izazvao početnu infekciju. Ta početno inficirana računala se nazivaju sekundarne žrtve (*engl. secondary victim*) [16]. Sekundarne žrtve ponovo šalju maliciozan kod do drugih sistema u mreži i ta kompromitirana računala također rade kao botovi. Botmaster daje

naredbe botovima da napadnu žrtvu. Veliki broj botova troši žrtvine ograničene resurse, najčešće propusnost (primjer ICMP i UDP preplavljivanja).

Botmaster inficira sekundarne, primjerice putem elektroničke pošte. Pri tom se koristi metodom pretraživanja ranjivosti (*engl. exploit scanning*).

Scenarij ovakvog DoS napada sličan je napadu iz jednog izvorišta s lažiranom izvorišnom adresom. Razlika je u intenzitetu napada, gdje u slučaju botneta žrtvu istovremeno napada velik broj koordiniranih botova. Botovi također mogu biti postavljeni da lažiraju izvorišne adrese (primjer stacheldracht [8]).

Shema 3.3 prikazuje primjer botnet napada.



Shema 3.3: Botnet DDoS napad

Na shemi je prikazano pet autonomnih sustava označenih slovima **A**, **B**, **C**, **D** i **E**. Inficirana računala (*botovi*) se nalaze unutar autonomnih sustava **B** i **D**. Maliciozni promet od inficiranih računala do žrtve je prikazan punom usmjerenom linijom. Izvorišne adrese u paketima inficiranih računala su lažirane, te su nasumično postavljene na adrese računala koja se nalaze unutar autonomnih sustava A i C. Odgovori žrtve na lažirane pakete napadača su prikazani isprekidanom usmjerenom linijom. Kao i kod napada iz jednog izvorišta s lažiranom IP adresom, ni u slučajevima napada iz više izvorišta, žrtvin autonomni sustav nije u stanju odrediti autonomni sustav napadača.

3.1.3. Napadi bez lažiranja izvorišne adrese

Lažiranje izvorišne adrese je tehnika koja prikriva stvarno izvorište napada. Također pakete s lažiranom izvorišnom adresom je teže filtrirati zato što se čini da svaki lažirani paket dolazi s druge

adrese. Zbog toga je u interesu napadača da lažira izvorišnu adresu kad god je to moguće. Ipak postoje situacije u kojima je to nemoguće. Primjer takve situacije je kada je napadač spojen na usmjernik koji ima implementirano izlazno filtriranje. U tom slučaju će svi paketi s lažiranom izvorišnom adresom koja je izvan adresnog prostora usmjernika biti odbačeni na usmjerniku. Na taj način promet napadača neće izaći iz mreže i napad neće imati efekta. U takvoj situaciji će napadač biti prisiljen koristiti ispravnu adresu ili će eventualno lažirati adresu na neku ispravnu vrijednost unutar njegove podmreže. Žrtva u takvom scenariju može odrediti autonomni sustav unutar kojeg se nalazi napadač jer raspolaže s njegovom IP adresom.

Ukoliko se u komunikaciji koristi TCP te je uspostavljen potpuni spoj tada se sa sigurnošću zna da izvorišna adresa nije lažirana, jer je u procesu trostranog rukovanja došao odgovor s dotične adrese.

Primjer takvog napada prikazan je na shemi 3.4.

Shema 3.4: DoS napad bez lažiranja izvorišne adrese

Usmjernik **A** sa sheme ima implementirano izlazno filtriranje pa napadač koristi izvorišnu adresu koja se nalazi unutar adresnog prostora usmjernika **A**. Dio infrastrukture Interneta između napadača i žrtve je prikazan oblakom. Žrtva sada jednostavno može odrediti autonomni sustav napadača jer ima ispravnu IP adresu.

Očiti problem gore navedenog scenarija je što žrtva ne može biti sigurna da li je u paketu koji je pristigao do nje izvorišna adresa lažirana ili ne, osim u slučaju kada je uspostavljen potpuni TCP spoj. Ipak implementiranjem ulaznog filtriranja možemo postići efekt da se paketi koji ne zadovoljavaju pravila filtriranja odbace na ulaznim usmjernicima, te tako smanjiti broj paketa s lažiranom izvorišnom adresom [6]. Također postoje i različite metode koje se koriste za detekciju lažiranih paketa. One osim metoda baziranih na usmjeravanju (*engl. routing based*), koriste i aktivne i pasivne metode bazirane na računalu (*engl. host based*). Te tehinke nisu potpuna rješenja, ali mogu značajno povećati sposobnost identifikacije lažiranih paketa [17].

3.2. Neželjen promet

Internetom se danas prenose velike količine neželjenog promet (*engl. unwanted traffic*). U najosnovnijem obliku neželjeni promet sačinjavaju paketi koji troše mrežne i računalne resurse na način koji ne koristi vlasnicima tih resursa [19]. Primjeri neželjenog prometa su DoS i DDoS napadi, neželjena elektronička pošta, virusi, crvi i sl.

Internet Architecture Board (*IAB*) je na radionici o neželjenom prometu kao jedan od glavnih zaključaka istaknuo činjenicu da je ogromna količina neželjenog prometa posljedica crne ekonomije [19]. Crna ekonomija se najčešće provodi na IRC (*engl. Internet Relay Chat*) poslužiteljima. IRC poslužitelji omogućavaju *trgovine* za prodavanje informacija o ukradenim kreditnim karticama, bankovnim računima, različite maliciozne programe (*engl. malware*), pristup mrežama botova, administratorski pristup do kompromitiranih računala i web poslužitelja [20]. Postoje *trgovine* za

DDoS napade, trgovine za kreditne kartice, trgovine s PayPal i bankovnim računima, kao i trgovine koje prodaju pristup do kompromitiranih Cisco i Juniper usmjernika [20].

Najveći problemi na Internetu vezani uz neželjni promet su [20]:

- Lažiranje izvorišnih adresa – Rasprostranjenost mreža botova koji mogu pokrenuti različite vrste napada korištenjem stvarnih adresa botova, govori o tome da lažiranje adresa nije više toliko važna tehnika kao što je to nekad bila. Ipak mnogo napada, posebice napadi refleksijom, i dalje koriste lažirane IP adrese.
- Lažiranje BGP staza – Ovo je napad na BGP protokol koji služi za usmjeravanje prometa između administrativnih domena. Napadač ubacuje staze za mreže koje se ne koriste, te kasnije koristi IP adrese iz tih mreža za slanje neželjene elektroničke pošte. Budući da lažirane staze ne utječu na normalan promet, takve staze je teško primijetiti.
- Sva komunikacija putem protokola HTTP – HTTP (*engl. Hypertext Transfer Protocol*) koji je namijenjen za pristup web poslužiteljima se često koristi kao protokol opće namjene u aplikacijama koje imaju malo ili nikakve veze sa webom. Razlog tome je što sigurnosne stijene zatvaraju sve ostale pristupe jer se ne koriste u komunikaciji. To dovodi do problema za nove aplikacije koje koriste pristupe koji se prethodno nisu koristili. Zato su dizajneri aplikacija odgovorili korištenjem HTTP komunikacijskog kanala za kojeg se prilično pouzdano može reći da je otvoren na sigurnosnoj stijeni. Ipak premještanje svog prometa na HTTP ne blokira napade, nego jednostavno premješta ranjivost s jednog mjesta na drugo.
- Svi dolaze od svuda (*engl. Everyone Comes from Everywhere*) – Na Internetu je bilo moguće dobiti neku indikaciju o autentičnosti prometa koji dolazi od nekog izvora na temelju broja skokova na usmjernicima. Svaki paket sadrži TTL (*engl. Time To Live*) polje i paketima koji su putovali istim stazama će biti jednako umanjena TTL polja. Promjene u TTL polju, bez promjena u ruti, su bili indikacija potencijalno malicioznog prometa. Međutim, u posljednje vrijeme korisnici su postali mobilni te promjena u TTL polju može jednostavno značiti da se korisnik pomaknuo. Također, višepristupnost (*engl. multihoming*) znači da dvije ili više vrijednosti TTL polja mogu biti jednako ispravne.
- Kompleksna mrežna autentifikacija – Mrežna autentifikacija koja se koristi danas je suviše kompleksna da bi korisniku bila učinkovita za korištenje.
- Nekorištenje sigurnosnih alata – različiti proizvođači i standardi su proizveli velik broj sigurnosnih alata. Ipak većina tih alata nisu u širokoj primjeni.

Prema istraživanju kompanije Arbor Networks za lažiranje IP adresa te lažiranje BGP staza se smatra da su trenutno dva najveća problema na Internetu vezana uz neželjeni promet[19].

Bitovi	0 -15	16-31
0	Source Port Number	Destination Port Number
32	Length	Checksum
64	Data	

Slika 3.1: Struktura UDP paketa

Najjednostavniji oblici neželjenog prometa mogu se detektirati pregledavanjem polja u zaglavljima paketa. Paketi s neispravnim poljima u zaglavlju bespotrebno troše resurse na mreži. U nastavku slijedi pregled zaglavlja popularnih protokola mrežnog i prijenosnog sloja te polja unutar zaglavlja koja su bitna za detekciju neželjenog prometa.

Slika 3.1 prikazuje strukturu UDP paketa. UDP promet do pristupa 0 nije valjan u normalnim okolnostima, te zbog toga polja *Destination Port Number* i *Source Port Number* ne smiju imati vrijednost 0.

Polje s podacima paketa (*engl. Data*) je obično manje od 4000 okteta budući da je UDP protokol namijenjen za slanje manjih paketa. Kada se pojave veći paketi to može biti znak nepravilne aktivnosti ili pokušaja DoS napada na udaljeno računalo.

Lenght određuje duljinu cijelog paketa (zaglavlja i podatkovnog dijela) u oktetima. Ako polje *lenght* ne odgovara stvarnoj duljini paketa onda se radi o neispravnom paketu, te se smatra neželjenim prometom.

Polje *checksum* služi za provjeru grešaka koje mogu nastati u prijenosu paketa. Provjerava se zaglavlje i podatkovni dio paketa. Ukoliko u paketu postoje greške takav paket se smatra neželjenim prometom. Drugi protokoli mrežnog i prijenosnog sloja također imaju *checksum* polje u zaglavlju, a njegova uloga je jednaka kao i kod UDP protokola, te u nastavku polje *checksum* neće biti ponovo istaknuto.

Slika 3.2 prikazuje strukturu zaglavlja TCP segmenta.

<table>
<tr><th>Bitovi</th><th colspan="10">0 - 15</th><th colspan="2">16 - 31</th></tr>
<tr><td>0</td><td colspan="10">Source port</td><td colspan="2">Destination port</td></tr>
<tr><td>32</td><td colspan="12">Sequence number</td></tr>
<tr><td>64</td><td colspan="12">Acknowledgment number</td></tr>
<tr><td>96</td><td colspan="2">Data offset</td><td colspan="2">Reserved</td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Window Size</td></tr>
<tr><td>128</td><td colspan="10">Checksum</td><td colspan="2">Urgent pointer</td></tr>
<tr><td>160</td><td colspan="12">Options + Padding</td></tr>
</table>

Slika 3.2: Struktura zaglavlja TCP segmenta

Kao i kod protokola UDP, ni kod TCP protokola nije valjan promet do pristupa 0, te polje *Destination Port* ne smije poprimiti vrijednost 0.

Ukoliko je u zaglavlju TCP segmenta postavljena zastavica *SYN* tada paket ne smije sadržavati podatkovno polje veće od 6 okteta. Razlog tome je što paketi s postavljenom SYN zastavicom služe samo za sinkronizaciju slijednih brojeva (*engl. Sequence number*), a ne i za prijenos podataka.

Zastavice *SYN* i *FIN* ne smiju biti istovremeno postavljene. Zastavica *SYN* služi za sinkronizaciju slijednih brojeva te se njome započinje nova TCP veza, dok zastavica *FIN* označava da pošiljatelj nema više podataka za poslati.

TCP je protokol koji implementira spojnu uslugu. Zastavice unutar zaglavlja TCP segmenta, označavaju stanje spoja (*engl. connection status*). RFC dokument koji specificira TCP protokol zahtijeva da neka od zastavica mora biti postavljena [23]. Zbog toga TCP segmenti bez bez

postavljenih zastavica (URG, ACK, SYN, PSH, RST, FIN) nisu valjani. Ovakvi paketi se najčešće koriste za prikupljanje informacija o udaljenom računalu, na taj način se pokušava otkriti koji su otvoreni, a koji zatvoreni TCP pristupi.

SCTP (*engl. Stream Control Transmission Protocol*) je transportni protokol kojem je temeljna jedinica prijenosa blok (*engl. chunks*). Svaka SCTP protokolna podatkovna jedinica (*engl. Protocol data unit - PDU*) se sastoji od zaglavlja, kontrolnih blokova, te podatkovnih blokova. Struktura SCTP PDU je prikazana slikom 3.3.

Bitovi	0 – 7	8 – 15	16 – 23	24 – 31
0	Source port		Destination port	
32	Verification tag			
64	Checksum			
96	Chunk 1 type	Chunk 1 flags	Chunk 1 length	
128	Chunk 1 data			
...	...			
...	Chunk N type	Chunk N flags	Chunk N length	
...	Chunk N data			

Slika 3.3: Struktura SCTP protokolne podatkovne jedinice

SCTP PDU bez bloka nije valjan. Takvi podatkovne jedinice su vjerojatni pokušaj DoS napada na Netfilter modul Linux jezgre, koji ne zna na pravilan način obraditi paket bez bloka.

Slika 3.4 prikazuje strukturu IP paketa.

Bitovi	0 – 3	4 – 7	8 – 13	14 – 15	16 – 18	19 – 31
0	Version	Header Length	Differentiated Services Code Point	Explicit Congestion Notification	Total Length	
32	Identification				Flags	Fragment Offset
64	Time to Live		Protocol		Header Checksum	
96	Source IP Address					
128	Destination IP Address					
160	Options					
160 ili 192+	Data					

Slika 3.4: Struktura IP paketa

Izvorišna i odredišna adresa unutar IP paketa ne smiju biti jednake. Paket s jednakom izvorišnom i odredišnom adresom može biti indikacija DoS Land napada. Također kod DoS Land napada je

najčešće postavljena i SYN zastavica unutar TCP zaglavlja, te je za takvu detekciju potrebno ispitati i IP i TCP zaglavlje.

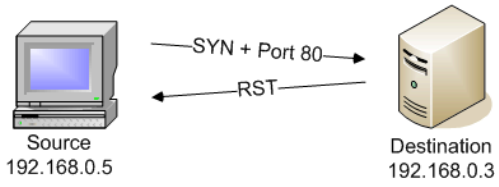
Polje *Protocol* unutar zaglavlja IP paketa služi za identifikaciju protokola koji se nalazi u podatkovnom dijelu IP paketa. Polje *Protocol* ne smije biti postavljeno na rezervirani ili nedodijeljeni protokol.

Polja *Identification*, *Flags* i *Fragment Offset* služe za IP fragmentaciju. Fragmenti se ne smiju međusobno preklapati. Ukoliko se fragmenti preklapaju to može biti pokušaj DoS Teardrop napada koji koristi propust u nekim implementacijama TCP/IP stoga.

Za detekciju nekih slučajeva neželjenog prometa nije dovoljno ispitati samo zaglavlje paketa, već je potrebno ispitati i podatkovni dio paketa. Najpoznatiji alati za DDoS napade, *Trin00*, *TFN* i *Stacheldraht*, osim podrazumijevanih (*engl. default*) protokola i pristupa, koriste i neke podrazumijevane naredbe [21]. Naredbe se uglavnom koriste za komunikaciju botmastera s inficiranim botovima. Takve naredbe moguće je primijetiti jedino ispitivanjem podatkovnog dijela paketa. Primjerice kod DDoS alata *Trin00* botmaster podrazumijevano komunicira s botovima korištenjem UDP protokola na pristupu 27444, a lozinka za botove je "l44adsl". Za detekciju takvog napada potrebno je pretražiti da li podatkovni dio paketa sadrži niz "l44adsl". Ovakim ispitivanjem i zaglavlja i podatkovnog dijela paketa moguće je primijetiti otiske (*engl. fingerprint*) različitih malicioznih aplikacija. Ipak treba napomenuti da je provjera podatkovnog dijela paketa procesno zahtjevna operacija s obzirom da treba pretražiti cijeli podatkovni dio da bi se utvrdilo da li se određeni niz pojavljuje ili ne [22].

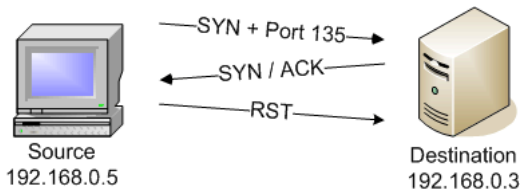
U nekim slučajevima nije dovoljno voditi računa samo o zaglavljima i podatkovnim dijelovima paketa kako bi smo detektirali neželjeni promet, nego je potrebno promatrati i načina na koji se pojedini protokol odvija. Pogledajmo primjer TCP protokola. Svaka nova TCP veza započinje s trostranim rukovanjem (*engl. three-way handshake*) kako bi se izmijenili i potvrdili slijedni brojevi. Tek nakon rukovanja slijedi prijenos podataka. Neke aplikacije, primjerice sigurnosni skeneri, koriste trostrano rukovanje samo kako bi saznali informacije o udaljenom računalu, stvarajući na taj način neželjen promet.

Slika 3.5 prikazuje prikazuje računalo koje uz pomoć sigurnosnog skenera (primjerice nmap) šalje TCP SYN paketa do pristupa 80 odredišnog računala. TCP pristup 80 na odredišnom računalu je zatvoren i iz tog razloga odredišno računalo šalje nazad TCP segment s postavljenom RST zastavicom.



Slika 3.5: SYN ping zatvorenog pristupa

Na slici 3.6 izvorišno računalo šalje TCP SYN paket do pristupa 135 na odredišnom računalu. Kako je taj pristup na odredišnom računalu otvoren, odredišno računalo odgovara sa SYN/ACK paketom. Kako je izvorišno računalo jedino htjelo saznati da li je pristup 135 otvoren ili ne, a ne uistinu i uspostaviti TCP vezu, ono šalje nazad do odredišnog računala RST paket.



Slika 3.6: SYN ping otvorenog pristupa

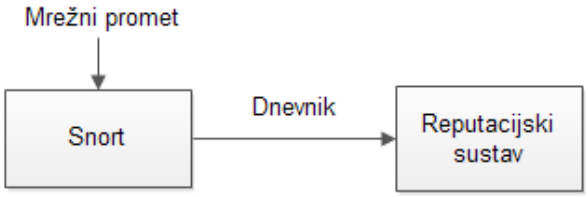
U primjeru sa slike 3.6 izvorišno računalo ne mora poslati posljednji RST paket ili može poslati paket sa postavljenim ACK i RST zastavicama.

Svi gore navedeni scenariji su sa stanovišta mreže neželjen promet iz razloga što veza nije niti uspostavljena ili je uspostavljena pa prekinuta. Drugim riječima mrežom nije prenesena nikakva korisna informacija.

Osim paketa koji započinju novu TCP vezu i prekidaju je prije prijenosa stvarnih podataka, na mreži se mogu pojaviti i TCP segmenti koji pokušavaju uspostaviti nevažeću (*engl. unsolicited*) TCP vezu. To su paketi između dva računala koja prethodno nisu prošla trostrano rukovanje ili je vrijeme veze isteklo (*engl. timeout*) te je potrebno ponovo započeti proces trostranog rukovanja. Takvi paketi nisu valjani i bespotrebno troše resurse mreže te se iz tog razloga također smatraju neželjenim prometom.

4. Mjerenje neželjenog prometa

Komponente sustava za mjerenje neželjenog prometa i njihova međusobna povezanost prikazana je slikom 4.1.



Slika 4.1: Sustav za mjerenje neželjenog prometa

U ovom radu je korišten sustav za otkrivanje napada Snort. Snort sluša mrežni promet na određenom sučelju, te uz pomoć skupa pravila detektira neželjeni promet. Sve informacije o otkrivenom neželjenom prometu Snort zapisuje u dnevnik. Reputacijski sustav, uvidom u dnevnik, ima informacije o količini i vrsti neželjenog prometa koji je pristigao iz pojedinog autonomnog sustava. Na temelju tih informacija reputacijski sustav vrši korekcije ocjena reputacije pojedinih autonomnih sustava.

U nastavku teksta su slijedi opis komponenta sustava za mjerenje neželjenog prometa.

4.1. Sustav otkrivanja napada

Sustav za otkrivanje napada (*engl. Intrusion detection system, u daljenjem tekstu IDS*) je programsko ili sklopovsko rješenje koje nadgleda mrežu ili sustav u cilju otkrivanja napada, odnosno malicioznih aktivnosti. IDS za otkrivanje napada može koristiti tehniku potpisa (*engl. signature*), tehniku temeljenu na nepravilnostima (*engl. anomaly-based*) ili obje tehnike [18].

Napadači, kao i računalni virusi, imaju potpise koji mogu biti otkriveni korištenjem programa. Potpisi su uzorci unutar podatkovnih paketa. Oni mogu biti prisutni u različitim dijelovima paketa u ovisnosti o prirodi samog napada. Potpisi se koriste za otkrivanje jedne ili više vrsta napada. Sustav otkrivanja napada pokušava pronaći podatkovne pakete koji sadrže bilo koje poznate potpise vezane uz napade ili nepravilnosti povezane uz Internet protokole. Na temelju skupa potpisa i pravila, IDS je u mogućnosti naći i zabilježiti bilo kakve sumnjive aktivnosti te izazvati alarm (*engl. alert*). Tehnika otkrivanja napada temeljena na nepravilnostima se obično oslanja na otkrivanje nepravilnosti u zaglavljima paketa.

Kao što je već prethodno spomenuto, IDS može nadgledati mrežu ili sustav. IDS temeljen na računalu (*engl. host-based IDS, u daljnjem tekstu HIDS*) se instalira poput programa. HIDS može gledati u dnevnik (*engl. log*) od drugih aplikacija da bi otkrio bilo kakve napadačke aktivnosti. HIDS je u stanju otkriti samo napad na računalo na kojem je instaliran. Mrežni IDS (*engl. Network IDS, u daljnjem tekstu NIDS*) nadgleda podatke koji putuju po cijeloj mreži te je u stanju otkriti napade na mreži.

Alarm (*engl. alert*) je bilo koja vrsta obavještavanja korisnika o napadačkim aktivnostima. Kada IDS otkrije napad, mora o tome obavijestiti administratora korištenjem alarma. Alarm može biti u obliku prikaza tekstualne poruke u konzoli, zasebnog prozora, slanja poruke elektroničke pošte

administratoru i sl. Alarmi se također zapisuje u dnevnik ili bazu podataka gdje kasnije mogu biti pregledani i ispitani.

Lažni alarmi (*engl. false alarm*) su alarmi koji se pokrenu u slučajevima kada nema napadačkih aktivnosti. Primjerice krivo podešena računala mogu neki puta odaslati poruku koja će zadovoljiti neko od pravila IDS-a te pokrenuti lažni alarm. Da bi se izbjegli takvi lažni alarmi, potrebno je podesiti parametre pojedinih pravila. Ponekad je potrebno u potpunosti isključiti neko pravilo.

Dnevnici se obično zapisuju u datoteke. Poruke u dnevnik mogu biti zapisane u tekstualnom ili binarnom formatu. Pohranjivanje u binarnom formatu je brže iz razloga što je binarni format kompaktniji te nema dodatnog formatiranja teksta.

Senzori (*engl. sensors*) su sustavi na kojima je pokrenut IDS. Na jednoj mreži može biti više senzora koji zatim šalju zabilježene podatke do centraliziranog poslužitelja.

U ovom radu se koristi Snort IDS. Snort je besplatni IDS, otvorenog koda. Može se koristiti u tri glavna načina rada: osluškivanje mreže (*engl. sniffer*), bilježenje paketa (*engl. packet logger*) ili kao sustav za detekciju napada. Kod osluškivanja mreže program čita pakete sa mreže i prikazuje ih na konzoli. Kada radi u načinu bilježenja paketa, program zapisuje pakete u dnevnik na disku. Funkcionalnost sustava za otkrivanje napada je prethodno opisana i Snort se u ovom radu koristi kao IDS.

Snort je mrežni IDS (NIDS). Dnevnik zapisuje u datoteku. Podrazumijevano se dnevnička datoteka nalazi na lokaciji */var/log/snort*, no lokacija dnevničke datoteke može biti promijenjena. Ako su poruke u dnevnik bile zapisivane u binarnom formatu, one se kasnije mogu pregledati korištenjem Snort ili *tcpdump* programa. Također, Snort može pohraniti dnevnik i u MySQL bazu podataka. Nadalje, korištenjem Snort-a, MySQL-a, Apache web poslužitelj i alata ACID (*engl. Analysis Control for Intrusion Database*), moguće je pregledavati pokušaje napada pomoću web sučelja. ACID je alat koji omogućava razne načine analize prikupljenih podataka. Moguće je prikazati frekvencije napada, klasificirati različite napade, pregledati izvore napada i sl. ACID korist PHP skriptni jezik, biblioteke za grafički prikaz (*engl. graphic display library*), te alat za iscrtavanje grafike – PHPLOT. Kombinacija tih alata omogućava generiranje web stranica koje prikazuju, generiraju i iscrtavaju podatke spremljene u MySQL bazi podataka.

Snort je sustav za otkrivanje napada temeljen na pravilima. Pravila se pak baziraju na potpisima napadača. Snort pravila se mogu koristiti da bi se provjerili različiti dijelovi podatkovnih paketa. Snort verzije 1.x može analizirati zaglavlja paketa mrežnog sloja i transportnog sloja. U verzijama 2.x je dodana podrška i za aplikacijski sloj. Snort pravila imaju jednostavnu sintaksu. Većina pravila je napisana u samo jednoj liniji. Pravila su obično pohranjena u konfiguracijsku datoteku *snort.conf*. Može se koristiti i više datoteka koje se kasnije uključe u glavnu konfiguracijsku datoteku.

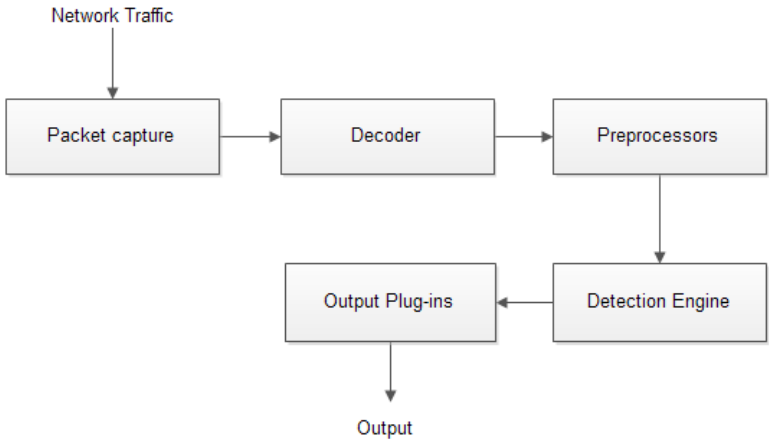
Snort ima i mogućnost programskih dodataka (*engl. plug-in*). Programski dodaci su dijelovi koda koji se kompajliraju i koriste se za modifikaciju izalaza ili ulaza Snorta. Ulazni programski dodaci pripremaju uhvaćene podatkovne pakete prije nego što se nad njima pokrene proces detekcije napada. Izlazni programski dodaci formatiraju izlaz da bi se mogao koristiti u posebne svrhe. Primjerice, izlazni programski dodatak može pretvoriti detektirane podatke u *MIB* format, *SNMP (engl. Simple Network Management Protocol)* protokola. Drugi izlazni programski dodatak se korisiti za zapis Snort izlaza u bazu podataka.

4.2. Arhitektura Snort sustava

Snort je logički podijeljen u više komponenti. Te komponente rade zajedno kako bi otkrile određen napad i generirale izlazne informacije o napadu u odgovarajućem obliku. Snort sustav za otkrivanje napada se sastoji od slijedećih glavnih komponenti:

- Dekoder (*engl. Decoder*)
- Pretprocesori (*engl. Preprocessors*)
- Sustav detekcije (*engl. Detection Engine*)
- Izlazni moduli (*engl. Output Modules*)

Slika 4.2 prikazuje kako su te komponente međusobno povezane.



Slika 4.2: Arhitektura Snort sustava

Dekoder uzima pakete s različitih tipova mrežnih sučelja i priprema ih za pretprocesore ili sustav detekcije. Postavlja pokazivače na podatkovni dio paketa i zaglavlja pojedinih protokola kako bi se kasnije mogli ispitivati.

Pretprocesori se koriste kako bi pripremili pakete prije nego se proslijede do sustava detekcije. Neki pretprocesori odmah pronalaze anomalije u zaglavljima paketa i generiraju alarm. Pretprocesori u Snortu defragmentiraju pakete, dekodiraju HTTP URI te sakupljaju TCP tokove.

Sustav detekcije je najvažnija komponenta u Snort sustavu. Odgovoran je za pronalaženje bilo kakvih nepravilnosti u paketima. Za tu svrhu koristi pravila. Pravila se čitaju u podatkovne strukture te se svi paketi ispituju s pravilima. Ako paket zadovoljava neko od pravila, poduzimaju se odgovarajuće radnje, u suprotnom se paket odbacuje. Ako je promet na mreži prevelik može se desiti da sustav detekcije ne stigne obraditi sve pakete. U tom slučaju neki paketi će biti odbačeni te nećemo dobiti pravi odziv u realnom vremenu. Opterećenje sustava detekcije ovisi o broju pravila, procesnoj snazi računala te opterećenju mreže.

Izlazni moduli mogu raditi različite operacije u ovisnosti o tome na koji način želimo pohraniti dnevnike i alarme. U ovisnosti o konfiguraciji izlazni moduli mogu:

- pohraniti dnevnike i alarme u */var/log/snort/alert* datoteku ili neku drugu datoteku

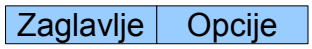
- slati SNMP poruke
- slati poruke do syslog programa
- bilježiti podatke u bazu podataka poput MySQL-a
- generirati poruke u eXML formatu
- modificirati konfiguracije usmjernika ili sigurnosnih stijena
- slati Server Message Block (SMB) poruke

Konfiguracije Snort sustava se pohranjuju u datoteku te se čitaju prilikom pokretanja Snorta. Podrazumijevano se konfiguracijska datoteka nalazi na lokaciji */etc/snort/* te nosi naziv *snort.conf*, iako se može koristiti i bilo koje drugo ime. Moguće je imati više konfiguracijskih datoteka te pokrenuti više instanci Snorta na različitim mrežnim sučeljima, s različitim postavkama specificiranim u konfiguracijskim datotekama. Svaka konfiguracijska datoteka je podijeljena u šest osnovnih sekcija:

1. Definicija varijabli. Varijable se koriste u pravilima, ali i u drugim slučajevima kao što je određivanje lokacije datoteka s pravilima.
2. Konfiguracijski parametri. Ti parametri određuju različite konfiguracijske opcije. Neki od njih se mogu navesti prilikom pokretanja Snorta iz komandne linije.
3. Konfiguracija pretprocesora. Pretprocesori se koriste da bi obavili različite akcije nad paketom prije nego paket obradi sustav detekcije.
4. Konfiguracija izlaznih modula. Izlazni moduli kontroliraju kako će Snort podaci biti pohranjeni u dnevničke datoteke.
5. Definiranje novih tipova akcija. Ako unaprijed definirani tipovi akcija nisu dovoljni, korisnik može definirati i vlastite tipove akcija.
6. Konfiguracija pravila i uključivanje datoteka s pravilima. Iako se pravila mogu dodavati i izravno u Snort konfiguracijsku datoteku, konvencija je da se koriste odvojene datoteke za pohranu pravila. Te datoteke s pravilima se kasnije uključuju u glavnu konfiguracijsku datoteku korištenjem ključne riječi *include*.

4.3. Snort pravila

Sva Snort pravila se sastoje od dva logička dijela: zaglavlja i opcija. Struktura pravila prikazana je slikom 4.3.



Slika 4.3: Struktura Snort pravila

Zaglavlje sadrži informacije o akciji koju pravilo poduzima. Također sadrži i kriterije za uspoređivanje pravila s podatkovnim paketima. U opcijama se obično nalazi poruka alarma i informacije o dijelu paketa koji će se koristiti za stvaranje poruke alarma. Opcije također sadrže i dodatne kriterije za uspoređivanje pravila s podatkovnim paketima. Jedno pravilo može otkriti jednu ili više vrsta napada.

Struktura zaglavlja Snort pravila prikazana je slikom 4.4.

Slika 4.4: Stuktura zaglavlja Snort pravila

Polje *akcija* određuje tip akcije koja će se poduzeti kada se pojavi paket koji zadovoljava sve kriterije pravila. Tipične akcije su stvaranje alarma, zapisivanje poruke u dnevnik *(engl. log)* ili pozivanje drugog pravila.

Polje *protokol* se koristi da bi se pravilo primijenilo samo na paketima određenog protokola. Ovo je prvi kriterij koji se pojavljuje u pravilima. Trenutno su podržana četiri protokola: TCP, UDP, IP i ICMP [24].

Polja *adresa1* i *adresa2* definiraju izvorišnu i odredišnu IP adresu. Može se koristiti adresa jednog računala, više računala ili adresa mreže. Izvorišna i odredišna adresa se određuju na temelju polja *smjer*. Polje *smjer* može sadržavati dva operatora smjera: "->" i "<>". Ako je postavljen smjer "->" to znači da polje *adresa1* određuje izvorišnu, a *adresa2* odredišnu adresu. Bidirekcionalni operator "<>" označuje da se pravilo primijenjuje nad paketima koji putuju i u jednom i u drugom smjeru. Operator "<-" ne postoji kako bi se pravila uvijek čitala konzistentno.

Kod protokola UDP i TCP polja *pristup1* i *pristup2* određuju izvorišni i odredišni pristup. U slučaju protokola IP i ICMP pristup se ne uzima u obzir.

Opcije Snort pravila slijede nakon zaglavlja i nalaze se unutar para oblih zagrada. Može biti jedna ili više opcija koje se međusobno odvajaju s znakom točka-zarez. Ako se koristi više opcija, one su međusobno povezane operatorom logičko I (*engl. AND*). Akcija u zaglavlju pravila se pokreće samo kada su zadovoljeni svi uvjeti iz opcija. Sve opcije su definirane s ključnim riječima (*engl. keyword*). Neke opcije sadrže i argumente. Općenito, opcija može imati dva dijela: ključnu riječ i argument. Argumenti se odvajaju od ključne riječi znakom dvotočka. Primjeri nekih ključnih riječi su *msg* (specificira poruku alarma), *content* (pretražuje podatkovni dio paketa za određenim sadržajem) i *ttl* (provjerava TTL polje IP paketa). U dodatku B na kraju rada se nalazi pregled svih ključnih riječi i njihovih značenja.

Pogledajmo primjer Snort pravila koje provjerava da li su u zaglavlju TCP paketa istovremeno postavljene zastavice SYN i FIN:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; stateless;
flags:SF,12; reference:arachnids,198; classtype:attempted-recon; sid:624;
rev:3;)
```

Dio pravila prije oblih zagrada je zaglavlje. Zaglavlje sadrži slijedeće dijelove:

- Akcija je alarm (*engl. alert*).
- Pravilo se primjenjuje samo nad paketima TCP protokola.
- *$EXTERNAL_NET* je varijabla definirana unutar Snort konfiguracijske datoteke i ona u ovom slučaju određuje izvorišnu adresu.
- *any* je ključna riječ koja govori da izvorišni pristup može biti postavljen na bilo koju vrijednost.
- "->" određuje smjer.
- $HOME_NET je varijabla definirana unutar Snort konfiguracijske datoteke i ona u ovom slučaju određuje odredišnu adresu.

- odredišni pristup također može biti postavljen na bilo koju vrijednost zbog ključne riječi *any*

Dio pravila koji se nalazi unutar zagrada su opcije. Pravilo sadrži slijedeće opcije:

- *msg* - poruka alarma
- Ključna riječ *stateless* označava da se pravilo primjenjuje ne uzimajući u obzir stanje TCP veze
- Ključna riječ *flags* se koristi za provjeru zastavica unutar TCP zaglavlja. U ovom slučaju provjerava se da li su postavljene zastavice SYN i FIN, te se pritom ignoriraju rezervirani bit 1 i rezervirani bit 2.
- *reference* daje referencu na informacije o napadu. Sve reference se pohranjuju unutar konfiguracijske datoteke *reference.config*.
- *classtype* klasificira pravilo. Informacije o klasifikaciji Snort pravila se nalaze unutar datoteke *classification.config*.
- *sid* (Snort ID) je jedinstveni identifikator pravila.
- Ključna riječ *rev* označava reviziju pravila

Sva pravila koja su korištena u okviru ovog rada su priložena u datoteci *unwanted_traffic.rules*. Datoteka sadrži pravila koja su distribuirana sa Snort sustavom kao i nekoliko dodatnih pravila koja su napisana za potrebe ovog rada. Ta dodatna pravila detektiraju neželjen promet koji započinje TCP sjednicu te je prekida prije prijenosa neke korisne informacije (primjer sigurnosnih skenera - pogledati poglavlje 3.2). Pravila prate stanja trostranog rukovanja kod uspostave TCP sjednice. Pretprocesor Stream5, opisan u poglavlju 4.4, omogućava praćenje stanja TCP sjednice. Prvo pravilo detektira početak trostranog rukovanja, odnosno TCP segment s postavljenom SYN zastavicom:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (flags:S,12; flowbits:set,
SYN; flowbits: noalert; msg: "Detektirano: SYN"; sid:1000001;)
```

Opcija *flowbits* se pojavljuje dva puta u pravilu. *flowbits* u kombinaciji s ključnom riječi *set* postavlja vezu u stanje *SYN* (`flowbits:set, SYN;`). Drugo pojavljivanje opcije *flowbits* u kombinaciji s ključnom riječi *noalert* označava da navedeno pravilo neće prouzročiti alarm, već će samo postaviti vezu u stanje *SYN*. Potpuno objašnjenje opcije *flowbits* i svih njezinih ključnih riječi nalazi se u poglavlju 4.4. Drugo pravilo provjerava da li se nakon *SYN* segmenta, pojavio *SYN-ACK* segment:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (flags:SA,12;
flowbits:isset, SYN; flowbits:set,SYN-ACK; flowbits:unset, SYN;
flowbits: noalert; msg:"Detektirano: SYN->SYN-ACK"; sid:1000002;)
```

Ovo pravilo provjerava da li je veza u stanju *SYN* opcijom *flowbits* i ključnom riječi *isset(`flowbits:isset, SYN;`)*. Zatim postavlja vezu u novo stanje *SYN-ACK*. Također izlazi iz starog stanja opcijom *flowbits* i ključnom riječi *unset* (`flowbits:unset, SYN;`). Niti ovo pravilo neće prouzročiti alarm iz razloga što se radi o normalnom nastavku TCP trostranog rukovanja.

Slijedeće pravilo pokreće alarm ukoliko je veza u *SYN* stanju te se detektira segment s postavljenom *RST* zastavicom:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (flags:R,12;
flowbits:isset, SYN; flowbits:unset, SYN; msg:"Detektirano: SYN->RST";
sid:1000003;)
```

Na jednak način slijedeća tri pravila ispituju stanja TCP veze do završetka trostranog rukovanja:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (flags:A,12;
flowbits:isset, SYN-ACK; flowbits:set, ACK; flowbits:unset, SYN-ACK;
flowbits: noalert; msg:"Detektirano: SYN->SYN-ACK->ACK"; sid:1000004;)


alert tcp $EXTERNAL_NET any -> $HOME_NET any (flags:R,12;
flowbits:isset, SYN-ACK; flowbits:unset, SYN-ACK; msg:"Detektirano: SYN-
>SYN-ACK->RST"; sid:1000005;)


alert tcp $HOME_NET any -> $EXTERNAL_NET any (flags:F,12;
flowbits:isset, ACK; flowbits:unset, ACK; msg:"Detektirano: SYN->SYN-
ACK->ACK->FIN"; sid:1000006;)
```

Slijedeće pravilo detektira segmente koji pokušavaju uspostaviti nevažeću TCP vezu (vezu koja nije prošla trostrano rukovanje):

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (flags:A,12;
flowbits:isnotset, SYN; flowbits:isnotset, SYN-ACK; flowbits:isnotset,
ACK; msg:"Detektirano: Unsolicited connection"; sid:1000007;)
```

Ovo pravilo detektira segmente s postavljenom *ACK* zastavicom te provjerava da li je za zadan segment veza prošla trostrano rukovanje ili je u procesu trostranog rukovanja. Ukoliko provjera ne uspije, pokreće alarm.

4.4. Stream5 pretprocesor

Stream5 pretprocesor pruža dvije osnovne funkcionalnosti:

1. Rekonstrukcija TCP tokova (*engl. TPC stream reassembly*)
2. Ispitivanje temeljeno na stanju veze *(engl. Stateful inspection)*

Stream5 ima mogućnost pratiti sjednice *(engl. session)* za TCP i UDP. TCP sjednice se raspoznaju putem klasične TCP veze. UDP sjednice su posljedica niza UDP paketa između dvije krajnje točke s istim skupom pristupa. ICMP poruke se prate kako bi se registrirale poruke nedohvatljiv (*engl. unreachable*) i nedostupan (*engl. unavailable*), koje prekidaju TCP i UDP sjednice.

Funkcionalnost Stream5 pretprocesora ovisi o odredištu, odnosno o operacijskom sustavu računala na kojem je pokrenut. Metode i politike (*engl. policies*) rukovanja različitim TCP anomalijama (npr. preklapajući TCP segmenti, podaci unutar segmenta s postavljenom SYN zastavicom) su ovisne o ciljanom operativnom sustavu.

Sva podešavanja Stream5 pretprocesora se vrše putem konfiguracijske datoteke Snorta, u sekciji opcije pretprocesora. Opcije su podijeljene na globalne opcije te TCP, UDP i ICMP specifične opcije. Postavke Stream5 pretprocesora koje su nužne za izvođenje praktičnog dijela ovog diplomskog rada su opisane u poglavlju 4.5.

Stream5 također uvodi nove ključne riječi *flow* i *flowbits* koje se mogu koristiti prilikom pisanja pravila za protokole UDP i TCP. *flow* omogućava pravilima da se primijene u ovisnosti o smjeru mrežnog prometa. Sintaksa ključne riječi *flow* je:

```
flow:[(established|stateless)] [,(to_client|to_server|from_client|from_server)]
[,(no_stream|only_stream)];
```

Objašnjenje značenja pojedinih opcija je navedeno u tablici 4.1.

Opcija	Značenje
to client	Primijeni na odgovoru poslužitelja od A do B
to server	Primijeni na zahtjevu klijenta od A do B
from client	Primijeni na zahtjevu klijenta od A do B
from server	Primijeni na odgovoru poslužitelja od A do B
established	Primijeni samo na uspostavljenim TCP vezama
stateless	Primijeni bez obzira na stanje
no stream	Ne primijeni na rekonstruiranim paketima iz toka (*engl. rebuilt stream packets*)
only stream	Primijeni samo na rekonstruiranim paketima iz toka

Tablica 4.1: flow opcije

flowbits omogućava pravilima praćenje stanja sjednice prijenosnog protokola, najčešće TCP sjednice. Postoji osam ključnih riječi povezanih s *flowbits* opcijom. Većina zahtijeva korisnički definirano ime za stanje koje se provjerava. Ime može biti bilo koji alfanumerički niz uključujući točke, crtice i i podvlaku (*engl. underscore*).

Opcija	Značenje
set	Postavlja stanje za određeni tok i izlazi iz svih preostalih stanja u grupi kada je GROUP_NAME naveden
unset	Izlazi iz određenog stanja za trenutni tok
toggle	Postavlja određeno stanje ako stanje nije postavljeno i izlazi iz svi preostalih stanja u grupi kada je GROUP_NAME naveden, u protivnom izlazi iz stanja ako je stanje postavljeno
isset	Provjerava da li je određeno stanje postavljeno
isnotset	Provjerava da li određeno stanje nije postavljeno
noalert	Uzrokuje da pravilo ne generira alarm bez obzira na ostatak opcija u pravilu

Tablica 4.2: flowbits opcije

Ključne riječi *set* i *toggle* primaju opcionalni argument koji specificira grupu u koju će stanja pripadati. Kada grupa nije navedena, stanja će pripadati podrazumijevanoj grupi. Sva stanja u pojedinoj grupi su međusobno isključiva, s iznimkom podrazumijevane grupe. Pojedino stanje ne može pripadati u više od jedne grupe. *flowbits* ima slijedeći format:

```
flowbits: [set|unset|toggle|isset|reset|noalert][,<STATE_NAME>][,<GROUP_NAME>];
```

Objašnjenja pojedinih opcija su navedena u tablici 4.2.

4.5. Pokretanje Snort sustava

Snort pravila se nalaze u direktoriju */etc/snort/rules* ako nije drugačije definirano tijekom provođenja programa. Pravila su po kategorijama grupirana unutar datoteka, te tako postoje primjerice datoteke *ftp.rules, icmp.rules, mysql.rules, voip.rules.* Cilj ovog diplomskog rada je otkriti samo neželjen promet te nam iz tog razloga sva pravila koja dolaze sa Snort sustavom nisu od interesa i potrebno ih je isključiti kako bismo kasnije imali dnevničke datoteke koje sadrže alarme vezane samo uz neželjen promet. Kako bi se isključila podrazumijevana Snort pravila potrebno je otvoriti Snort konfiguracijsku datoteku te pronaći rubriku "*#7: Customize your rule set*". Unutar te rubrike, potrebno je zakomentirati sva uključivanja (*engl. include*) vanjskih datoteka. Znak za komentar je "#" i potrebno ga je dodati na početku svake linije, prije ključne riječi *include*. Zatim je potrebno uključiti pravila za detekciju neželjenog prometa koja su priložena uz ovaj diplomski rad unutar datoteke *unwanted_traffic.rules*. Prethodno je potrebno datoteku kopirati u direktorij s ostalim pravilima(*/etc/snort/rules*) te u konfiguracijsku datoteku, u istu rubriku s pravilima, dodati slijedeću liniju:

```
include $RULE_PATH/unwanted_traffic.rules
```

Nakon što su pravila podešena, potrebno je još podesiti i pretprocesore prije pokretanja Snorta. U ovom diplomskom radu se koristi pretprocesor *Stream5*. Potrebno je unutar konfiguracijske datotke Snorta u rubrici "*#5: Configure preprocessors*" provjeriti da je uključen pretprocesor *Stream5* i njegove opcije "*track_tcp yes*" i "*detect_anomalies*". Zatim je još potrebno kreirati direktorij unutar kojeg će Snort pohranjivati svoje dnevničke datoteke.

Nakon što smo obavili sve prethodno navedene radnje, Snort pokrećemo iz komandne linije na slijedeći način:

```
snort -c put_do_konf._datoteke -l put_do_log_foldera -b -i sučelje
```

Značenja pojedinih opcija su:

- *-c* navodi put do konfiguracijske datoteke Snorta
- *-l* navodi put do direktorija unutar kojeg će se pohranjivati dnevnici
- *-b* specificira da će se paketi pohranjivati u binarnom formatu
- *-i* navodi mrežno sučelje na kojem će Snort slušati

Primjer pokretanja Snorta na mrežnom sučelju *eth0*, s konfiguracijskom datotekom */etc/snort/snort.conf*, te direktorijem */var/log/snort* za pohranu dnevnika:

```
snort -c /etc/snort/snort.conf -l /var/log/snort -b -i eth0
```

Ovako pokrenuti Snort će unutar dnevničkog direktorija stvoriti datoteku *alert* unutar koje će biti pohranjeni alarmi. Primjer jednog alarma iz *alert* datoteke je:

```
[**] [1:527:10] DELETED BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
05/13-18:10:51.852041 0.0.0.0:68 -> 255.255.255.255:67
UDP TTL:64 TOS:0x0 ID:28079 IpLen:20 DgmLen:328
Len: 300
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0016][Xref =>
http://www.securityfocus.com/bid/2666]
```

Alarm se sastoji od redom:

- GID - ID generatora; prvi broj iz niza `[1:527:10]`. On govori korisniku koja komponenta Snorta je generirala alarm (npr. dekoder, pretprocesor). U ovom slučaju GID je postavljen na vrijednost 1 te nam govori da je alarm došao od sustava detekcije. Lista svih GID-ova se nalazi u izvornom kodu Snorta u datoteci *etc/generators*.
- SID - Snort ID; drugi broj iz niza `[1:527:10]`. Lista pretprocesorskih SID-a se nalazi u datoteci *etc/gen-msg.map* u izvornom kodu. SID-ovi pravila su zapisani u samim pravilima uz pomoć ključne riječi *sid*.
- ID revizije; treći broj iz niza `[1:527:10]`. Taj broj se primarno koristi prilikom pisanja pravila. Svaka nova revizija treba inkrementirati broj pomoću opcije *rev*.
- Poruka alarma "*DELETED BAD-TRAFFIC same SRC/DST*". Poruke alarma definiramo unutar pravila ključnom riječi *msg*.
- Klasifikacija alarma (ukoliko smo je prilikom pisanja pravila specificirali pomoću ključne riječi *classtype*). Alarm iz primjera spada u kategoriju "*Potentially Bad Traffic*".
- Prioritet alarma. Niži brojevi prioriteta označavaju veći prioritet alarma. Prioritet iz primjera je 2.
- Datum i vrijeme generiranja alarma. U primjeru je alarm generiran 13.05. u 18:10:51.
- Izvorišna IP adresa i izvorišni pristup. U primjeru je izvorišna IP adresa 0.0.0.0. i pristup 68.
- Odredišna IP adresa i odredišni pristup. U primjeru je odredišna IP adresa 255.255.255.255 i pristup 67.
- Protokol. U alarmu iz primjera je UDP.
- TTL (*engl. Time To Live*) polje unutar zaglavlja IP paketa. U primjeru je vrijednost TTL polja 64.
- TOS (*engl. Type of Service*) polje unutar zaglavlja IP paketa. U primjeru je vrijednost TOS polja 0x0 heksadecimalno.
- ID polje IP paketa. U primjeru je vrijednost ID polja 28079.
- Veličina zaglavlja IP paketa u oktetima. 20 okteta u primjeru.
- Ukupna veličina IP paketa u oktetitma. 328 okteta u primjeru.
- Veličina podatkovnog dijela paketa. U primjeru 300 okteta (ukupna veličina IP paketa umanjena za 20 okteta IP zaglavlja i 8 okteta UDP zaglavlja).
- Reference na informacije o potencijalnom uzroku alarma.

4.6. Kažnjavanje neželjenog prometa

Nakon što je pokrenut, Snort sustav bilježi sve alarme unutar dnevničke datoteke. Svaki alarm nastaje nakon detekcije nekog oblika neželjenog prometa. Međutim nisu svi oblici neželjenog prometa jednako štetni. Iz tog razloga se pojavila potreba za odgovarajućim sustavom kažnjavanja neželjenog prometa.

Dnevničke datoteke su u okviru ovog rada generirane uz pomoć skupa pravila iz datoteke *unwanted_traffic.rules*. Tablica 4.3 prikazuje sve alarme iz dnevničih datoteka kao i odgovarajuće ocjene kažnjavanja neželjenog prometa. Ocjene kažnjavanja su u intervalu [1-5]. Alarm *1:1000007 Unsolicited connection* ima najveću mogućnost lažnog uzbunjivanja (*engl. false alarm*). Ukoliko Snort senzor iz bilo kojeg razloga ne primijeti jedan od paketa trostranog rukovanja, svi paketi te veze će izazivati lažni alarm *1:1000007*. Upravo zbog velike mogućnosti izazivanja lažnog alarma, alarm *1:1000007* ima najmanju ocjenu kažnjavanja; *1*. Krivo postavljena vrijednosti unutar zaglavlja pojedinih paketa, nepoštivanje određenih protokola i ostali oblici neželjenog prometa koji najčešće nisu posljedica napada ili malicioznih radnji, već krivo podešenih programa i mrežne opreme, se kažnjavaju ocjenom *2*. Sigurnosna skeniranja se kažnjavaju ocjenom *3*. Pokušaji zaobilaženja sustava za otkrivanje napada i mogući pokušaji napada se kažnjavaju ocjenom *4*. Najviše se kažnjavaju pokušaji iskorištavanja poznatih ranjivosti *(engl. vulnerability)* kao i pokušaji DoS napada. Kazna u tom slučaju ima vrijednost *5*.

Prilikom računanja reputacije autonomnih sustava, broj pojavljivanja određenog alarma se množi s pripadajućom ocjenom alarma (računanje reputacije AS-a je objašnjeno u poglavlju 5.2.). Tako se primjerice jedan zabilježen pokušaj DoS napada kažnjava jednako kao i pet zabilježenih pokušaja uspostave nevažeće TCP veze (*engl. Unsolicited connection*).

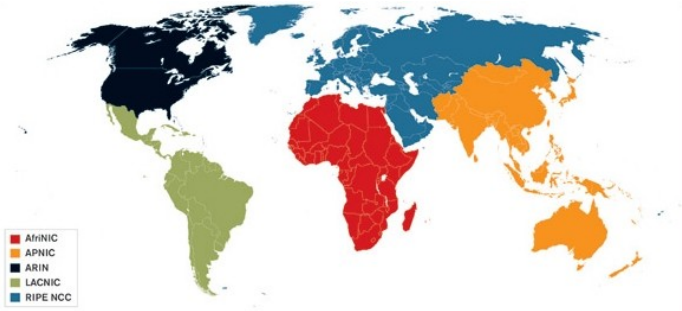
Ocjene kažnjavanja pojedinih alarma mogu se po potrebi mijenjati u konfiguracijskoj datoteci reputacijskog sustava. Napomenimo da je u konfiguracijsku datoteku moguće dodavati i ocjene kažnjavanja za nove alarme. Na taj je način reputacijski sustav moguće proširiti novim pravilima za detekciju neželjenog prometa (dodavanjem u datoteku *unwanted_traffic.rules*) kao i odgovarajućim kaznama, bez mijenjanja izvornog koda samog sustava. Konfiguracijska datoteka reputacijskog sustava je objašnjenja u poglavlju 5.1.

GID	SID	Opis alarma	Kazna
1	527	DELETED BAD-TRAFFIC same SRC/DST	2
1	624	SCAN SYN FIN	3
1	11263	DOS Apache mod_ssl non-SSL connection to SSL port denial of service attempt	5
1	15259	DOS DNS root query traffic amplification attempt	5
1	1000003	Detektirano: SYN->RST	3
1	1000005	Detektirano: SYN->SYN-ACK->RST	3
1	1000007	Unsolicited connection	1
119	14	NON-RFC DEFINED CHAR	4
119	15	OVERSIZE REQUEST-URI DIRECTORY	4
119	19	(http_inspect) LONG HEADER	2
120	3	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	2
123	8	(spp_frag3) Fragmentation overlap	2
123	12	(spp_frag3) Excessive fragment overlap	4
124	2	(smtp) Attempted data header buffer overflow	5
124	3	(smtp) Attempted response buffer overflow	5
124	7	(smtp) Attempted header name buffer overflow	5
128	4	(spp_ssh) Protocol mismatch	2
129	3	Data sent on stream not accepting data	2
129	4	TCP Timestamp is outside of PAWS window	2
129	5	Bad segment, adjusted size <= 0	2
129	7	Limit on number of overlapping TCP packets reached	4
129	8	Data sent on stream after TCP Reset sent	2
129	12	Consecutive TCP small segments exceeding threshold	2
129	14	TCP Timestamp is missing	2
129	15	Reset outside window	2
129	19	TCP window closed before receiving data	2
137	1	(ssp_ssl) Invalid Client HELLO after Server HELLO Detected	2

Tablica 4.3: Pregled svih alarma i odgovarajućih kazni

5. Reputacijski sustav

Zadaća reputacijskog sustava izgrađenog u sklopu ovog rada je određivanje reputacije autonomnih sustava (u daljnjem tekstu AS). AS je administrativna cjelina odnosno jedna ili više mreža koje se nalaze pod jedinstvenim administrativnim upravljanjem. AS je osnovni element arhitekture Interneta i identificira se po globalno jedinstvenom broju, broju autonomnog sustava (*engl. Autonomus System Number, ASN*). ASN je 16 ili 32 bitni broj koji se ne može lako izmijeniti te stoga predstavlja stabilni identitet AS-a. *IANA* (*engl. Internet Assigned Number Authority*) je organizacija čija je jedna od zadaća koordinacija brojeva autonomnih sustava. *IANA* alocira IP adrese i brojeve autonomnih sustava iz skupa nealociranih brojeva i predaje ih regionalnim Internet registrima (*engl. Regional Internet Registry RIR*) ili direktnom nekom autonomnom sustavu. Postoji pet regionalnih registara, a njihova nadležnost je prikazana na slici 5.1.



Slika 5.1: Reginalni Internet registri

RIPE NCC pokriva Grenland, Europu, Bliski istok i centralnu Aziju, *ARIN* je vezan za područje sjeverne Amerike, *LACNIC* pokriva Latinsku Ameriku i neke Karipske otoke, *AfriNIC* pokriva Afriku, a *APNIC* Azijsko pacifičku regiju. Informacije o autonomnim sustavima se nalaze u bazama podataka na poslužiteljima regionalnih registara. Detaljne informacija o određenom AS-u možemo dobiti samo ispitivanjem poslužitelja koji je vezan uz regiju u kojoj se nalazi AS. Informacije se od poslužitelja dobivaju putem *Whois* protokola [26]. *Whois* protokol je protokol aplikacijskog sloja koji služi za slanje upita i odgovora na upite. Upiti primjerice mogu biti vezani uz informacije o imenima domena, blokovima IP adresa ili o autonomnim sustavima. Protokol isporučuje informacije u korisniku čitljivom formatu.

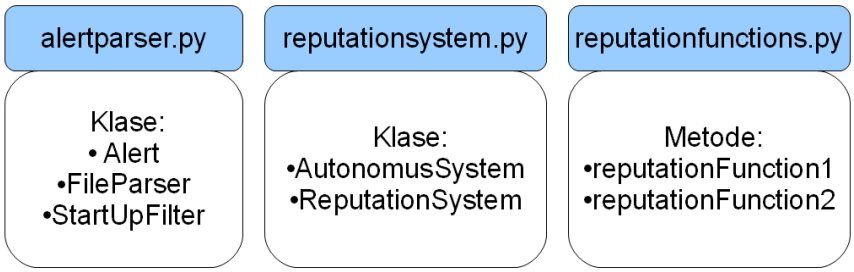
Rad reputacijskog sustava implementiranog u okviru ovog rada može se pojednostavljeno prikazati u četiri koraka:

1. *Čitanje alarma iz dnevničke datoteke* - reputacijski sustav najprije iz dnevničke datoteke pročita informacije o alarmu. Unutar dnevničkih datoteka su zabilježene samo IP adrese koje su uzrokovale određeni alarm, ali ne i broj autonomnog sustava pod čijim se upravljanjem nalazi adresa.
2. *Dobavljanje ASN-a za IP adresu iz alarma* - korištenjem *Whois* protokola dobavlja brojeve autonomnih sustava za IP adrese koje su uzrokovale alarm.

3. *Kažnjavanje AS-a koji je prouzročio alarm* - različite vrste alarma se različito kažnjavaju kako je to objašnjeno u poglavlju 4.6. . Ocjene kazne su zapisane u konfiguracijskoj datoteci reputacijskog sustava. (Više o konfiguracijskoj datoteci u poglavlju 5.1.)
4. *Računanje reputacije AS-a pomoću reputacijske funkcije* - uz pomoću odgovarajuće reputacijske funkcije, reputacijski sustav izračunava konačnu ocjenu reputacije za pojedini AS. Više o reputacijskim funkcijama i izračunu reputacije u poglavlju 5.2. .

5.1. Organizacija programskog kôda

Reputacijski sustav je napisan u programskom jeziku Python. Slika 5.2 prikazuje organizaciju kôda po modulima.



Slika 5.2: Moduli reputacijskog sustava

Kôd se sastoji od tri Python modula: *alertparser.py*, *reputationsystem.py* i *reputationfunctions.py*.

U *alertparser* modulu se nalaze klase: *Alert*, *FileParser* i *StartUpFilter*. Objekt tipa *FileParser* parsira datoteku s alarmima i vraća objekte tipa *Alert*. *FileParser* u tu svrhu implementira metodu *getNextAlert*. Metoda *getNextAlert* vraća slijedeći alarm, redom kojim su alarmi zapisani u dnevničkoj datoteci. Alarmi se dohvaćaju i obrađuju jedan po jedan, a ne svi od jednom, kako bi program bio manje memorijski zahtjevan (na taj način nisu svi alarmi iz dnevničke datoteke istovremeno pohranjeni u memoriji). Početno filtriranje *Unsolicited connection* alarma (period dok Snort ne registrira sve već uspostavljene TCP veze) se postiže klasom *StartUpFilter*. *StartUpFilter* zapravo samo specijalizira metodu *getNextAlert* klase *FileParser* na način da izbaci početna pojavljivanja *Unsolicited connection* alarma. Iz tog razloga klasa *StartUpFilter* nasljeđuje klasu *FileParser*.

Modul *reputationsystem* sadrži klase: *AutonomusSystem* i *ReputationSystem*. Svaki objekt tipa *AutonomusSystem* ima pohranjen broj autonomnog sustava (atribut *asn*). Alarmi pojedinog autonomnog sustava su pohranjeni u hash tablici *alerts*. Ta hash tablica ima za ključeve jedinstvene identifikatore alarma (kombinacija GID:SID; GeneratorID:SnortID), a za vrijednosti broj pojavljivanja tog alarma. *AutonomusSystem* sadrži još i trenutnu reputaciju (atribut *currentrep*) te listu prethodno izračunatih reputacija (atribut *replist*). *AutonomusSystem* ima i metodu *addNewAlert* koja dodaje novi alarm. Objekt tipa *ReputationSystem* se incijalizira s postavkama iz konfiguracijske datoteke. Konfiguracijska datoteka je podrazumijevano *unwanted_traffic.config*, ali se može i drugačije zvati, a također može biti i više konfiguracijskih datoteka kako bi se reputacijski sustav mogao pokretati s više različitih postavki. Reputacijski sustav računa reputaciju za svaki autonomni sustav u pojedinom vremenskom period. Pritom najprije skuplja alarme koji su se desili u zadanom vremenskom periodu, a onda izračunava reputaciju na temelju prikupljenih alarma. Prilikom izračunavanja reputacije koristi reputacijsku funkciju. Sve reputacijske funkcije se nalaze

u modulu *reputationfunctions*, a odabir reputacijske funkcije se vrši kroz konfiguracijsku datoteku. *ReputationSystem* objekt prilikom inicijalizacije sprema referencu na reputacijsku funkciju. Na ovaj način je moguće proširivanje sustava s novim reputacijskim funkcijama. Dovoljno je samo kôd nove reputacijske funkcije pohraniti u *reputationfunctions* modulu te u konfiguracijskoj datoteci navesti ime nove funkcije. Sve reputacijske funkcije rade s objektima tipa *AutonomusSystem,* odnosno izračunavaju reputaciju na temelju alarma koji su zabilježeni za dotični AS.

Kako bi mogao identificirati iz kojeg AS-a dolazi određeni alarm *ReputationSystem* sadrži i dvije metode koje za određenu IP adresu vraćaju broj AS-a unutar kojeg se nalazi adresa. To su metode *getASN* i *getASNFromFile*. Metoda *getASN* koristi *Whois* protokol i Linux naredbu *whois*. Ukoliko dnevničke datoteke sadrže veliki broj alarma, dobavljanje brojeva AS-a putem *Whois* protokola može biti dosta vremenski zahtjevno. Kako bi se lakše moglo eksperimentirati s parametrima reputacijskog sustava, klasa *ReputationSystem* sadrži i metodu *getASNFromFile.* Ta metoda dobavlja brojeve AS-a iz tekstualne datoteke. Datoteka se prethodno mora kreirati korištenjem *Whois* protokola. Datoteka je formata "IP_adresa ASN", te ima ime jednako kao i dnevnička datoteka sa sufiksom "*-ip-asn*". Primjerice za dnevničku datoteku "*alert_00"* odgovarajuća datoteka za dobavljanje AS brojeva je "*alert_00-ip-asn*". Reputacijski sustav provjerava da li postoji takva datoteka, te ukoliko postoji dobavlja broj AS-a iz nje pomoću metode *getASNFromFile.* Ukoliko datoteka ne postoji, zove metodu *getASN* i dobavlja broj AS-a pomoću W*hois* protokola. Uz ovaj rad je priložene i datoteke "*alert-ip-asn*" u kojima se nalaze brojevi AS-a za IP adrese iz dnevničke datoteke "*alert".*

Konfiguracijska datoteka (*unwanted_traffic.config*) ima strukturu sličnu Microsoft Windows INI datotekama. Datoteka se sastoji od sekcija čija su imena navedena unutar uglatih zagrada(npr. *[Section]*) te parova oblika *ime:vrijednost*. Datoteka trenutno sadrži dvije sekcije: *General* i *Penalty.* Sekcija *General* sadrži općenite postavke reputacijskog sustava kao što su vrijeme izračunavanja reputacije ili lokacija dnevničkih datoteka. Značenje svih postavki konfiguracijske datoteke je ukratko objašnjeno unutar same datoteke pomoću komentara. Komentari započinju znakom '*#'.* Sekcija *Penalty* definira kažnjavanje za pojedine alarme kako je to navedeno tablicom 4.3. Vrijednosti kažnjavanja se po potrebi mogu mijenjati. Također sustav se lako može nadograditi dodavanjem novih Snort pravila te definiranjem kažnjavanja za nove alarme.

Ispis 5.1 prikazuje primjer dijela konfiguracijske datoteke. Reputacijski sustav pokrenut pomoću konfiguracijske datoteke iz primjera će određivati reputaciju na temelju alarma zapisanih u datoteci *alert_01* koja se nalazi unutar direktorija *alerts.* Početno filtriranje je postavljeno na 30 minuta, a reputacija se računa u periodima od 8 sati. Prilikom računanja reputacije koristit će se osnovna težinska funkcija normalizirana brojem pogrešaka (*reputationFunction2*) i faktor opadanja α u vrijednost od 0.5. Više o reputacijskim funkcijama i faktoru opadanja u poglavlju 5.2. U sekciji Penalty konfiguracijske datoteke definirana su kažnjavanja pojedinih alarma. Tako se primjerice alarm *1:527* kažnjava ocjenom *2*, a alarm *1:11263* ocjenom *5*.

```
#################
# GENERAL
#################
[General]
#Path to the alert file
AlertFile: alerts//alert_01
#Startup time in minutes
StartUp: 30
#Reputation period in hours
ReputationPeriod: 8
# Rn+1 = alpha * Rn + (1 - alpha) Rw
# alpha <0,1>
alpha: 0.5
#Reputation function
ReputationFunction: reputationFunction2
#################
# PENALTY
#        Format:
# GID:SID         penalty
#################
[Penalty]
Penalty: 1:527    2
         1:624    3
         1:11263  5
         1:15259  5
         1:1000003         3
```

Ispis 5.1: Primjer konfiguracijske datoteke reputacijskog sustava

5.2. Reputacijska funkcija

Prilikom računanja reputacije, korištena je sljedeća rekurzivna funkcija:

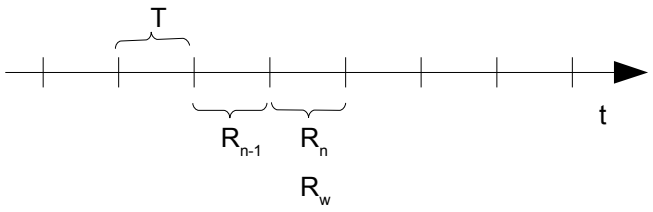
$$R_{n+1} = \alpha R_n + (1-\alpha) R_w$$

1. R_{n+1} je nova vrijednost reputacije pojedinog AS-a u periodu $n+1$
2. R_n je reputacija pojedinog AS-a u prethodnom periodu n
3. α je faktor opadanja (*engl. decay factor*)
4. R_w rezultat težinske funkcije za period T

Nova vrijednost reputacije se računa svaki put nakon isteka vremenskog perioda *T*, izraženog u satima. Pri tom je početna reputacija, odnosno reputacija u nultom periodu, nula. Vrijednost vremenskog perioda *T* moguće je podesiti u konfiguracijskoj datoteci (više o konfiguracijskoj datoteci u poglavlju 5.1.).

Faktor opadanja α je realan broj definiran u rasponu $\alpha = \langle 0,1 \rangle$ i predstavlja utjecaj reputacije prethodnog perioda *n – 1* na vrijednost reputacije u novom periodu *n.* Kako nova reputacija R_w utječe upravo obrnuto, razlikom *1 – α*, faktor opadanja zapravo određuje postotni udio R_n i nove vrijednosti R_w u konačnoj reputaciji. Vrijednost faktora opadanja je također moguće podesiti u konfiguracijskoj datoteci.

Slika 5.3 prikazuje periode računanja reputacije.



Slika 5.3: Vremenska podjela reputacije

Vrijeme unutar kojeg se mjeri reputacija se podijeli na vremenske periode *T*. Zatim se izračunava reputacija za svaki od vremenskih perioda. Reputacija R_n u vremenskom periodu *n* je kombinacija reputacije R_{n-1} iz prethodnog vremenskog perioda *n-1* i težinske funkcije R_w. Udio R_{n-1} i R_w u izračunu trenutne reputacije R_n se regulira faktorom opadanja α.

Reputacijski sustav podržava dvije funkcije R_w koje je moguće birati u konfiguracijskoj datoteci:

1. $R_{w1} = \sum r_i \beta_i$ osnovna težinska funkcija

2. $R_{w2} = \frac{\sum r_i \beta_i}{\sum r_i}$ osnovna težinska funkcija normalizirana brojem pogrešaka

Parametar r_i predstavlja broj pojavljivanja određenog alarma u danom periodu računanja. Parametar β_i predstavlja ocjenu kazne za pripadajući alarm, na način kako je to definirano tablicom 4.3. Osnovna težinska funkcija R_{w1} je suma umnožaka broja pojavljivanja alarma i njegove vrijednosti kazne. Ocjenu kazne β_i je moguće podešavati u konfiguracijskoj datoteci.

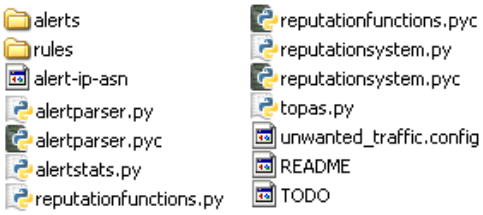
Funkcija R_{w2} normalizira funkciju R_{w1} brojem pojavljivanja alarma. Normalizacijom apsolutne vrijednosti reputacije dobivamo relativnu vrijednost s obzirom na broj pojavljivanja alarma. Funkcija R_{w2} zapravo teži prema srednjoj vrijednosti kazne najviše zastupljenog alarama. Primjerice ukoliko su za neki AS uglavnom zabilježeni alarmi čija je ocjena kazne 3, tada će funkcija težiti k 3.

Reputacijske funkcije se nalaze unutar Python modula *reputationfunctions.py.* Trenutno su raspoložive dvije reputacijske funkcije, imena *ReputationFunction1* i *ReputationFunction2,* a odabiru se postavljanjem *ReputationFunction* atributa unutar konfiguracijske datoteke. Funkcija *ReputationFunction1* koristi osnovnu težinsku funkciju dok *ReputationFunction2* koristi osnovnu težinsku funkciju normaliziranu brojem pogrešaka. Sustav je u jednostavno nadograditi s novim reputacijskim funkcijama, bez izmjena postojećeg izvornog koda. Potrebno je samo implementaciju reputacijske funkcije dodati na kraj datoteke *reputationfunctions.py* te unutar konfiguracijske datoteke postaviti za vrijednost atributa *Reputation function* ime nove funkcije. Prilikom

implementacije novih reputacijskih funkcija je bitno znati da funkcija kao ulazni parametar prima objekt tipa *AutonomusSystem*. Funkcija putem hash tablice *alerts* objekta *AutonomusSystem* ima uvid u alarme vezane uz dotični AS te na željeni način može izračunati novu reputaciju. Na posljetku, funkcija treba ažurirati atribute *currentrep* (trenutna reputacija) i *replist* (lista mijenjanja reputacije kroz vremenske periode *T*) objekta *AutonomusSystem.*

5.3. Pokretanje reputacijskog sustava

Slika 5.4 prikazuje listu svih direktorija i datoteka isporučenih na mediju priloženom uz ovaj rad.



Slika 5.4: Datotečno stablo reputacijskog sustava

Unutar direktorija *alerts* se nalazi se datoteka *alert*. To je dnevnička datoteka sustava za otkrivanje napada Snort, nastala mjerenjem neželjenog prometa za potrebe ovog rada. Više o dnevničkoj datoteci u poglavlju 6.1. . U direktoriju *rules* se nalazi datoteka *unwanted_traffic.rules* koja sadrži sva Snort pravila koja su se koristila za detekciju neželjenog prometa. Datoteka *alert-ip-asn* sadrži brojeve autonomnih sustava za IP adrese alarma iz datoteka *alert*. Datoteke *alertparser.py*, *reputationsystem.py* i *reputationfunctions.py* su moduli reputacijskog sustava i njihova je uloga već objašnjenja u poglavlju 5.1. unwanted_traffic.config je konfiguracijska datoteka reputacijskog sustava, također objašnjenja u poglavlju 5.1. Moduli *alertstats.py* i *topas.py* služe za generiranje statistike. *alertstats.py* generira statistiku alarma koji se nalaze unutar dnevničke datoteke. Modul se iz komandne linije Linux operacijskog sustava pokreće na slijedeći način:

```
python alertstats.py alert_file startup_time
```

pri tom je *alert_file* put do dnevniče datoteke s alarmima, a *startup_time* vrijeme početnog filtriranja u minutama. Nakon što je pokrenut *alertstats* modul ispisuje redom: vrijeme početnog filtriranja, datum i vrijeme prvog alarma iz datoteke, datum i vrijeme posljednjeg alarma iz datoteke, sve alarme koji su zabilježeni u dnevničkoj datoteci i njihov postotni udio u svim alarmima, te broj svih zabilježenih alarma.

Modul *topas.py* ima slijedeću sintaksu:

```
python alertstats.py broja_najgorih_ASova [-r]
```

Prvi argument je broj najgorih AS-ova i on je obavezan. Ukoliko se navede opcija *-r,* tada će lista AS-ova biti sortirana prema srednjoj vrijednosti reputacije (aritmetička sredina svih vrijednosti iz reputacijske liste), u suprotnom lista se sortira s obzirom na broj zabilježenih alarma. Modul u svojem ispisu prikazuje broj autonomnog sustava te broj alarma ili srednju reputaciju koja je vezan uz njega.

U okviru rada se još nalazi tekstualna datoteka *README* koja sadrži upute za pokretanje sustava, tekstualna datoteka *TODO* s listom zadaća koje bi trebalo realizirati u budućnosti, te tri Python kompajlirane datoteke (ekstenzija *pyc*).

Reputacijski sustav pokrećemo u dva jednostavna koraka. Najprije pomoću željenog tekst editora postavimo željene postavke u konfiguracijskoj datoteci *unwanted_traffic.config*, a zatim pokrenemo modul *reputationsystem*. Modul ispisuje sve zabilježene autonomne sustava i njihove liste reputacije. Lista reputacije sadrži vrijednosti reputacije koje su izračunate po periodima T. Kako broj autonomnih sustava i njihovih lista reputacije može biti prilično dugačak, neki put ga može biti korisno pohraniti u datoteku:

```
python reputationsystem.py > reputation_lists.txt
```

6. Rezultati mjerenja

U okviru ovog rada je provedeno mjerenje neželjenog prometa pomoću sustava za otkrivanje napada Snort, na način kako je opisano u poglavlju 4.5. Mjerenje je provedeno u razdoblju od 13.05.2011 do 19.05.2011. na poslužitelju Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave, koji se nalazi unutar autonomnog sustava CARNet. Kao rezultat mjerenja generirana je dnevnička datoteka *alert* unutar koje su pohranjeni svi alarmi zabilježeni za vrijeme mjerenja.

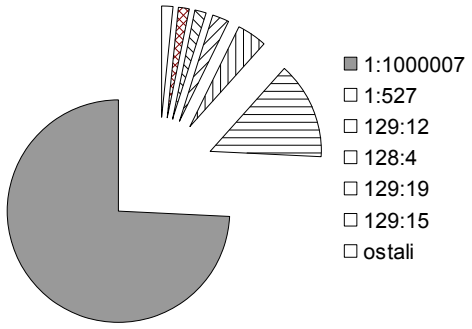
6.1. Analiza alarma

Osnovna statistika dnevničke datoteke *alert* prikazana je tablicom 6.1.

Veličina datoteke	115.630KB
Startup filter	0 min
Ukupno alarma	404969
Vrijeme prvog alarma	13.05.2011. 18:10:51.641788
Vrijeme zadnje alarma	19.05.2011. 11:54:58.562249

Tablica 6.1: Osnovna statistika dnevničke datoteke alert

Bez uključenog početnog filtriranja u datoteci je zabilježeno 404969 alarma. Grafički prikaz zastupljenosti pojedinog alarma dan je slikom 6.1. Imena alarma su formata *GID*:*SID* (*GeneratorID*:*SnortID*) , a značenje pojedinog alarma može se dobiti uvidom u tablicu 4.3.



Slika 6.1: Zastupljenost alarma u datoteci *alert* bez početnog filtriranja

U datoteci se nalazi ukupno 27 različitih vrsta alarma, a neki od njih se pojavljuju u malom postotku. Tablica 6.2 prikazuje sve alarme iz datoteke *alert*, broj njihovih pojavljivanja te postotni udio u ukupnom broju alarma. I tablica 6.2 i slika 6.1 upućuju na veliko pojavljivanje alarma *1:1000007* od čak 74.24%. Radi se o alarmu *Unsolicited connection*, a veliki udio u ukupnoj količini alarma upućuje na potrebu početnog filtriranja. Naime, na početku pokretanja Snorta pravila nemaju informacije o trenutno uspostavljenim TCP vezama, te podižu lažne alarme za sve

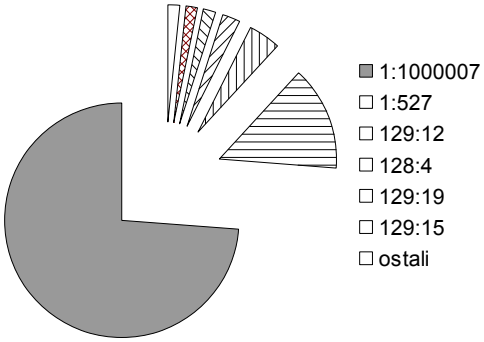
TCP veze u tijeku. Kako bismo uklonili te lažne alarme, uključujemo početno filtriranje.

GID	SID	Broj pojavljivanja	%
1	527	57191	14.122
1	624	57	0.014
1	11263	3	0.001
1	15259	5	0.001
1	1000003	409	0.101
1	1000005	196	0.048
1	1000007	300665	74.244
119	14	18	0.004
119	15	10	0.002
119	19	1295	0.320
120	3	122	0.030
123	8	20	0.005
123	12	130	0.032
124	2	1	0.000
124	3	2580	0.637
124	7	157	0.039
128	4	9739	2.405
129	3	49	0.012
129	4	347	0.086
129	5	571	0.141
129	7	445	0.110
129	8	5	0.001
129	12	17658	4.360
129	14	45	0.011
129	15	6351	1.568
129	19	6776	1.673
137	1	124	0.031

Tablica 6.2: Tablični prikaz zastupljenosti alarma u datoteci alert bez početnog filtriranja

Početnim filtriranjem u trajanju od 30 minuta, broj alarma *1:1000007* smanjio se s 300665 na 293471. Time smo uklonili čak 7194 pogrešna alarma te postotni udio alarma *1:1000007* sada iznosi 73.78%. Slika 6.2 prikazuje zastupljenost alarma nakon početnog filtriranja u trajanju od 30 minuta. Veliki udio alarma *1:1000007* i nakon filtriranja ukazuje na činjenicu da Snort pravila nisu uhvatila sve pakete trostranog rukovanja za neke TCP veze. Uzrok tome može biti opterećenost Snort sustava, te u slučaju kada Snort ne stigne obraditi sve pakete dolazi do odbacivanja paketa. U prilog tome ide i činjenica da se kod detekcije alarma *1:1000007* koristi ispitivanje sa stanjima koje je procesno zahtjevnije. Drugi uzrok može biti činjenica da se Snort senzor ne nalazi na mjestu kojim su prošli svi paketi trostranog rukovanja. Nadalje, u slučaju kada Snort ne registrira cijeli proces trostranog rukovanja za neku TCP vezu, svaki paket te veze će uzrokovati novi alarm. Kod

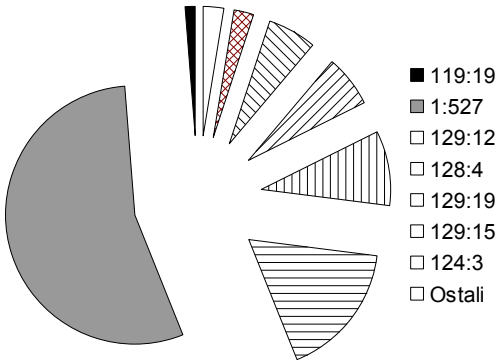
veza u kojima se izmjenjuje puno podataka to može rezultirati jako velikim brojem generiranih alarma.



Slika 6.2: Zastupljenost alarma u datoteci *alert* s početnim filtriranjem od 30 min

Upravo zbog velike nepouzdanosti alarma *1:1000007,* analiza reputacije autonomnih sustava se provodila na 2 načina: uzimajući u obzir alarm *1:1000007* i zanemarujući ga.

Slika 6.3 prikazuje strukturu alarma iz datoteke alert, zanemarujući pri tom alarm *1:1000007.* Ukupan broj zabilježenih alarma sada iznosi 104304.

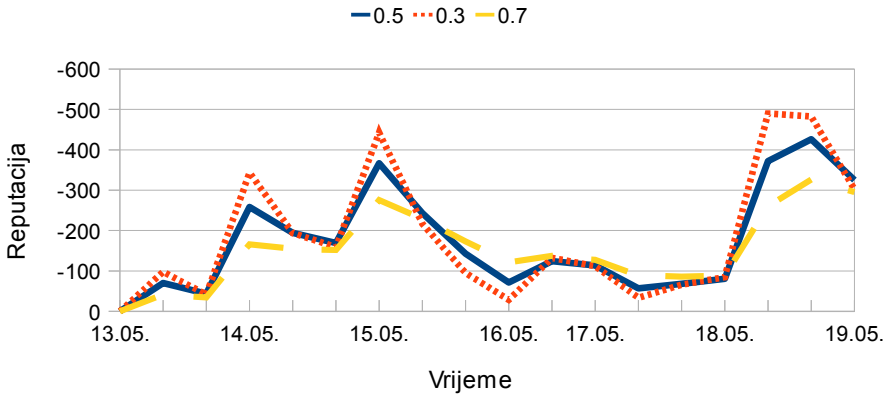


Slika 6.3: Zastupljenost alarma u datoteci *alert* bez alarma *1:1000007*

Najviše zastupljen alarm sad je *1:527* koji se pojavljuje kada su izvorišna i odredišna adresa u paketu jednake. Slijede *129:12* koji znači da je broj uzastopnih ponavljanja malih TCP segmenta premašio graničnu vrijednost, te *128:4* koji se javlja kada se verzija SSH protokola kod klijenta i poslužitelja ne podudaraju.

6.2. Analiza reputacije

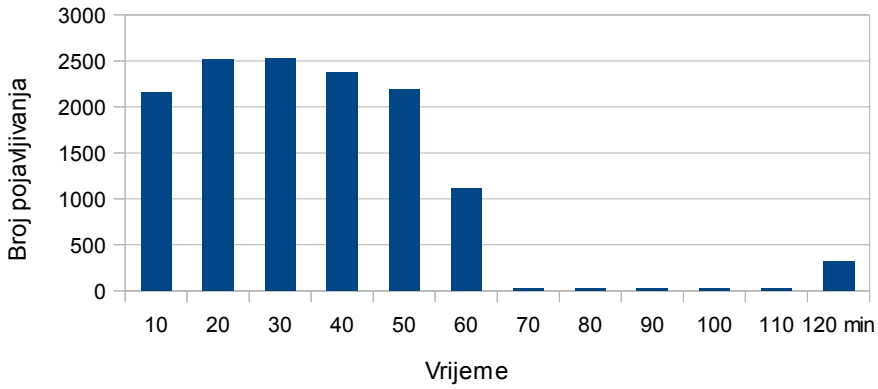
Grafikon 6.1 prikazuje utjecaj faktora opadanja na izračun reputacije. Na grafikonu je prikazana reputacija autonomnog sustava *AS21502* (*NUMERICABLE*) izračunata pomoću osnovne težinske funkcije. Period računanja reputacije je iznosi 8 sati, a početno filtriranje je bilo postavljeno na 30 minuta.



Grafikon 6.1: Utjecaj faktora opadanja na reputaciju

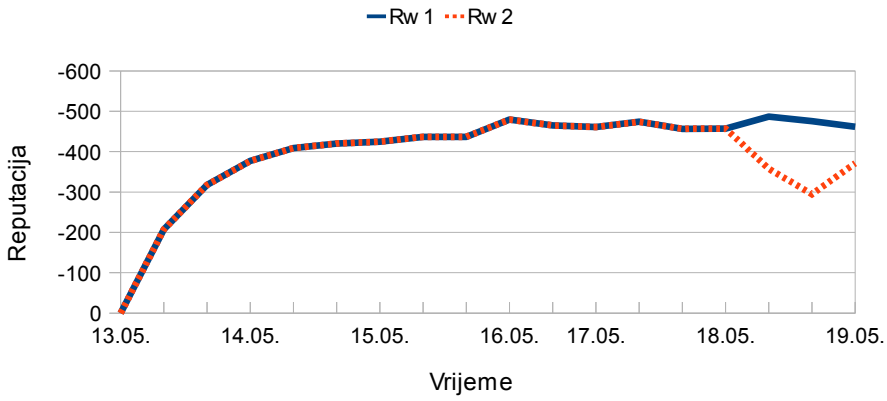
Grafikon prikazuje reputacije izračunate za faktore opadanja 0.3, 0.5 i 0.7. Ukoliko je faktor opadanja postavljen na 0.3, tada će u izračunu nove vrijednosti reputacije, stara vrijednost reputacije utjecati s 30% , a reputacija izračunata u trenutnom vremenskom periodu sa 70%. Takav graf reputacije ima veće oscilacije vrijednosti reputacije. Takvo ponašanje je posebno loše u periodima kada je za neki AS pristigla mala količina neželjenog prometa. Razlog tome je da ne želimo dobiti nagli rast reputacije nego postići da autonomni sustav tijekom dužeg vremenskog perioda *zasluži* dobru reputaciju. Na primjeru iz grafikona 6.1 se takav nagli rast reputacije najbolje vidi u razdoblju od 15.05 do 16.05. kada je reputacija narasla po apsolutnoj vrijednosti za približno 400. Ukoliko faktor opadanja postavimo na 0.7 tada će stara reputacija imati čak 70% udjela u novoj reputaciji. To uzrokuje puno sporije mijenjanje vrijednosti reputacije. Takvo ponašanje je pak loše u periodima kada je za neki AS pristigla velika količina neželjenog prometa iz razloga što želimo da reputacija, u takvim slučajevima, brzo opada. Za postizanje željenog mijenjanja vrijednosti reputacije najbolji se pokazao faktor opadanja 0.5 te je stoga korišten u daljnjim mjerenjima. Faktor opadanja 0.5 je generirao vrijednosti reputacije koje se nalaze između vrijednosti reputacija generiranih faktorima 0.3 i 0.7.

Grafikon 6.2 prikazuje broj pojavljivanja alarma *1:1000007* za datoteku *alert* u prva dva sata mjerenja. Broj pojavljivanja je mjeren u intervalima od 10 minuta. Iz grafikona je vidljivo da se broj pojavljivanja alarma *1:1000007* drastično smanjio nakon prvih 60 minuta mjerenja. To se desilo zato što su nakon 60 minuta istekle sve TCP veze koje su bile u tijeku kada je Snort pokrenut. Iz tog razloga je dobro trajanje početnog filtriranja postaviti na 60 minuta.



Grafikon 6.2: Broj pojavljivanja alarma 1: 1:1000007

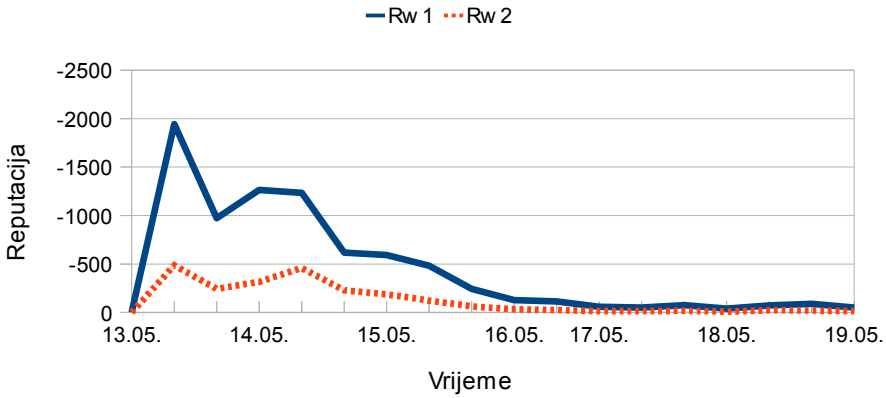
Uzimajući u obzir i alarm *1:1000007,* u datoteci *alert* su zabilježeni alarmi koji su pristigli iz 347 različtih autonomnih sustava. Ukoliko izostavimo *CARNet*, autonomni sustav unutar kojeg su provedena mjerenja, tada je najviše neželjenog prometa pristiglo iz autonomnog sustava *AS18705.* Radi se o Kanadskoj telekomunikacijskoj kompaniji *Research In Motion Limited (RIM)*, koja je poznata kao proizvođač *BlackBerry* smartphonea. Grafikon 6.3 prikazuje mijenjanje reputacije autonomnog sustava *AS18705* u ovisnosti o vremenu.



Grafikon 6.3: Reputacija autonomnog sustava AS18705

R_{w1} je osnovna težinska funkcija, R_{w2} osnovna težinska funkcija normalizirana brojem grešaka (pogledati poglavlje 5.2.). Početno filtriranje je postavljeno na 30 minuta, a faktor opadanja α je iznosio 0.5. Reputacija se mjerila u periodima od 8 sati. Nakon početnog razdoblja mjerenja, u trajanju od približno jednog dana, vrijednost reputacije se stabilizirala te se uglavnom kretala između *-400* i *-500*.

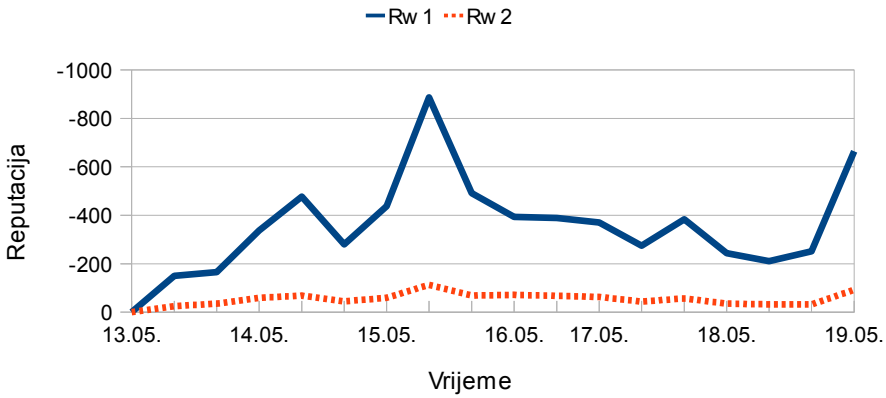
Drugi po redu autonomni sustav po količini pristigloga neželjenog prometa je *AS31012.* Riječ je o kompaniji *B.net* iz Hrvatske koja pruža Internet i telefonske usluge te usluge kablovske televizije. Reputacija sustava *AS31012* prikazana je grafikonom 6.4.



Grafikon 6.4: Reputacija autonomnog sustava AS31012

Početno filtriranje je iznosilo 30 minuta, faktor opadanja α je imao vrijednost 0.5, a reputacija se mjerila u periodima od 8 sati.

Treći autonomni sustav po količini izmjerenog neželjenog prometa je *AS5391*. To je *T-HT*, davatelj telekomunikacijskih usluga u Hrvatskoj. Njegova reputacija prikazana je grafikonom 6.5.

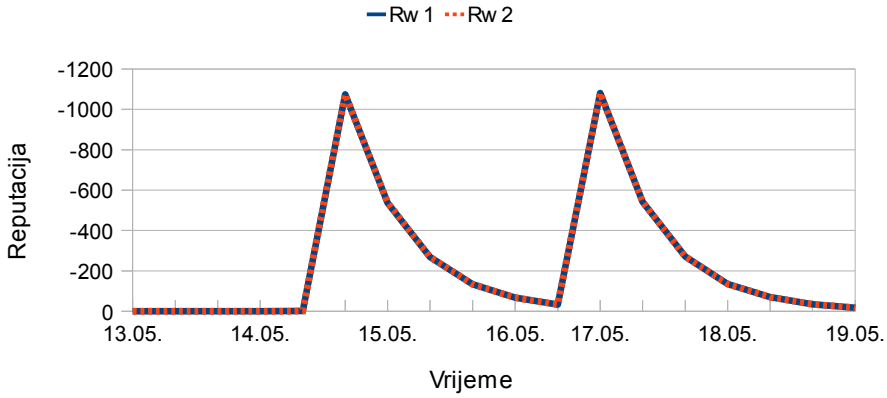


Grafikon 6.5: Reputacija autonomnog sustava AS5391

Početno filtriranje je iznosilo 30 minuta, faktor opadanja α je imao vrijednost 0.5, a reputacija se mjerila u periodima od 8 sati.

Sva tri prethodno navedena mjerenja reputacije su uzimala u obzir alarm *1:1000007*. Ne uzimajući u obzir alarm *1:1000007*, neželjen promet zabilježen u datoteci *alert* je pristigao iz. 339 različitih autonomnih sustava. To je 8 AS-a manje, odnosno za 8 AS-a je zabilježen isključivo alarm *1:1000007*.

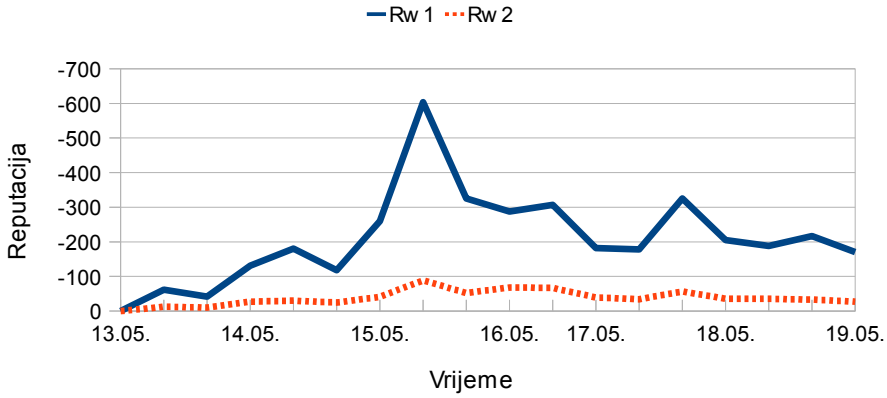
Uz zanemarivanje alarma *1:1000007,* najviše neželjenog prometa je pristiglo iz autonomnog sustava *AS15169*. Reputacija sustava *AS15169,* uz faktor opadanja 0.5 i period mjerenja od 8 sati, prikazana je grafikonom 6.6.



Grafikon 6.6: Reputacija autonomnog sustava AS15169, uz zanemarivanje alarma 1:1000007

Radi se o dobro poznatom autonomnom sustavu *Google Inc*. Grafovi reputacije za obje težinske funkcije su identični. To ukazuje na činjenicu da je unutar perioda računanja reputacije bila zabilježena isključivo jedna vrsta alarma. Uvidom u dnevničku datoteku doznajemo da se radi o alarmu *129:15 Reset outside window*. Također, dobro je uočiti da je reputacija u dva navrata, u kratkom vremenskom periodu poprimila izrazito veliku vrijednost. Pretpostavka je da se radi o periodima kada je bio pojačan promet između autonomnih sustava *CARNet* i *Google Inc*. Kako bi se izbjegle velike oscilacije u reputaciji, osnovnu težinsku funkciju bilo bi dobro normalizirati s ukupnom količinom prometa pristiglog iz nekog AS-a. Mjerenja ukupne količine prometa nisu provedena u okviru ovog rada te to ostaje kao jedna od zadaća za buduća mjerenja reputacije temeljana na količini neželjenog prometa.

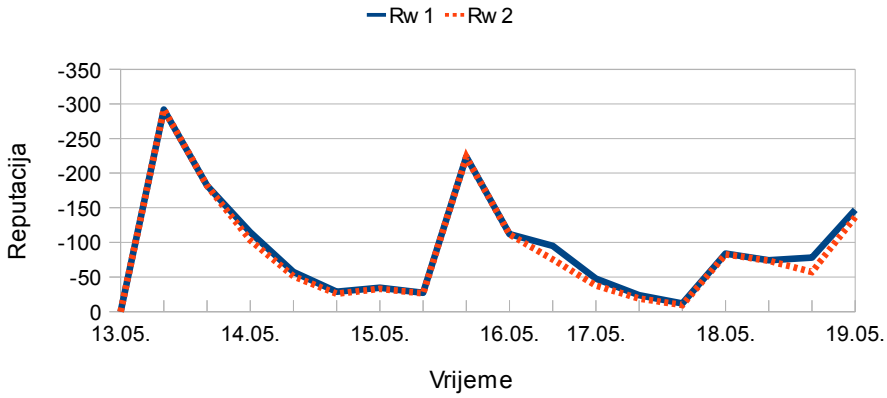
Uz zanemarivanje alarma *1:1000007*, drugi po redu najgori AS je *AS5391* (T-HT), koji je bez zanemarivanja bio treći. Grafikon 6.7 prikazuje njegovu reputaciju uz faktor opadanja 0.5 i period mjerenja reputacije od 8 sati.



Grafikon 6.7: Reputacija autonomnog sustava AS5391 uz zanemarivanje alarma 1:1000007

Grafikon 6.5 i grafikon 6.7 su prilično slične forme uz nešto manje apsolutne vrijednosti na grafikonu 6.7. Radi se naravno o razlici u broju pojavljivanja alarma *1:1000007*.

Treći po redu AS po količini pristiglog neželjenog prometa je *AS174*. Njegova reputacija prikazana je grafikonom 6.8.



Grafikon 6.8: Reputacija autonomnog sustava AS174

Reputacija je mjerenja uz faktor opadanja 0.5 u periodima od 8 sati. *AS174* registriran je na kompaniju *Cogent Communications*. Riječ je o internacionalnom pružatelju Internetskih usluga sa sjedištem u Washingtonu.

6.3. Najgori AS-ovi

Nekoliko je mogućih kriterija za određivanje najgorih AS-ova. Najjednostavniji način je dakako po količini zabilježenih alarma koji se vežu uz pojedini AS. Kako bismo dobili uvid u *n* najgorih autonomnih sustava sortiranih po broju zabilježenih alarma, dovoljno je pozvati skriptu *topas.py* i kao argument joj predati broj *n*. Tablica 6.3 donosi pregled 20 najgorih AS-ova po broju zabilježenih alarma.

U oba slučaja *AS2108* je prvi na listi. To je broj autonomnog sustava *CARNet*, unutar kojeg je provedeno mjerenje. Velik broj zabilježenih alarma vezanih uz *CARNet* ne čudi. Za očekivati je da će Snort senzor registrirati najveću količinu prometa upravo iz AS-a unutar kojeg se nalazi, a proporcionalno tome i najveću količinu neželjenog prometa. Također Snort pravila detektiraju mnoge jednostavne anomalije, koje će primjerice izlaznim filtriranjem ili sigurnosnom stijenom biti odbačene, pa se neće niti pojaviti izvan granica AS-a.

Na listi koja uzima u obzir i alarm *1:1000007* nakon *CARNet* slijede *AS18705 (RIM)*, *AS31012(B.net)* te *AS5391(T-HT)*. Grafovi reputacije tih autonomnih sustava prikazani su u prethodnom poglavlju 6.2. Na petom mjestu je *AS21502, NUMERICABLE*, kablovski mrežni operater iz Francuske. Zatim slijede redom *AS12810 (VIPnet d.o.o.)*, *AS15169 (Google Inc.)*, *AS2200 (FR-RENATER; Francuska nacionalna mreža za telekomunikacije i tehnologiju)*, *AS174 (Congent Communication)* i *AS47988 (A-STEDNABANKA; Štedna banka malog poduzetništva d.d.)*. Podatke o preostalim autonomnim sustavima moguće je dobiti putem *Whois* protokola.

Lista koja ne uzima u obzira alarm *1:1000007* osim već prethodno spomenutih autnomnih sustava *AS2108 (CARNet)*, *AS15169 (Google Inc.)*, *AS5391 (T-HT)*, *AS174 (Congent Communications)* sadrži redom i sustave *AS2914 (NTT America)*, *AS766 (RedIRIS, Španjolska)*, *AS31012 (B.net)*, *AS47988 (A-STEDNABANKA)*, *AS9146 (BIHNET)* te *AS12810 (VIPnet d.o.o.)*.

#	S alarmom *1:1000007*	Bez alarma *1:1000007*
1.	AS2108	AS2108
2.	AS18705	AS15169
3.	AS31012	AS5391
4.	AS5391	AS174
5.	AS21502	AS2914
6.	AS12810	AS766
7.	AS15169	AS31012
8.	AS2200	AS47988
9.	AS174	AS9146
10.	AS47988	AS12810
11.	AS35549	AS26627
12.	AS2914	AS34594
13.	AS766	AS13046
14.	AS9146	AS13238
15.	AS26627	AS20875
16.	AS34594	AS6849
17.	AS13046	AS4134
18.	AS13238	AS13092
19.	AS20875	AS29485
20.	AS6849	AS3462

Tablica 6.3: 20 najgorih AS-ova prema broju zabilježenih alarma

Reputacija je vremenski promjenjiva i istekom svakog perioda T računanja reputacije, ona poprima novu vrijednost. Kako bismo mogli uspoređivati reputacije različitih AS-ova mjerene kroz duži vremenski period, možemo se poslužiti sa njihovom aritmetičkom sredinom:

$$\bar{R} = \frac{R_1 + \ldots + R_n}{n}$$

Modul *topas.py* kojem kao argument predamo broj n i opciju *-r,* ispisat će prvih n najgorih autonomnih sustava sortiranih prema aritmetičkoj sredini njihove reputacije.

Tablica 6.5 prikazuje 20 najgorih AS-ova prema srednjoj vrijednosti reputacije za osnovnu težinsku funkcija R_{w1} i osnovnu težinsku funkciju normaliziranu brojem grešaka R_{w2}, sa i bez utjecaja alarma *1:1000007.* Ponovo pri vrhu svih rubrika susrećemo već poznate autonomne sustave: *AS2108 (CARNet), AS31012 (B.net), AS18705 (RIM), AS5391 (T-HT), AS15169 (Google Inc.) AS21502 (NUMERICABLE), AS174 (Congent Communication), AS47988 (A-STEDNABANKA), AS2914 (NTT America).*

U [27] je mjerena reputacija autonomnih sustava temeljena na praćenju DNS sustava. Pri tom je korištena jednaka rekurzivna funkcija koja je definirana u poglavlju 5.2. te je korišteno pet

različitih težinskih funkcija među kojima su i osnovna težinska funkcija R_{w1} i osnovna težinska funkcija normalizirana brojem grešaka $R_{w2.}$ Mjerene su različite pogreške unutar DNS protokola, a pogreške su se kažnjavale ocjenama iz intervala [0 - 60]. Kao rezultat mjerenja priložena je tablica 20 najgorih autonomnih sustava u pet mjernih točaka. Tablica sadrži šest istih AS-a koji su se našli i u tablici 6.5, 20 najgorih AS-a ovog rada. Autonomni sustavi koji su se našli u najgorih 20 AS i po mjerenju reputacije na temelju praćenju DNS sustava, i po mjerenju reputacija na temelju neželjenog prometa su:

1. *AS5391 T-HT*
2. *AS15169 Google Inc.*
3. *AS174 Congent Communications*
4. *AS2914 NTT America*
5. *AS6849 UKRTELNET*
6. *AS17408 PT Telekomunikasi Indonesia*

Napomenimo još da su tri AS-a koji su se našli na listi 20 najgorih po reputaciji temeljenoj na neželjenom promet, bili mjerne točke prilikom mjerenja reputacija na temelju praćenja DNS-a. To su: *AS35549 (Metronet), AS47988 (A-STEDNABANKA)* i *AS2108 (CARNet).*Loša ocjena reputacije lokalnih autonomnih sustava, posljedica je intenzivne komunikacije među njima.

Naknadnim, dužim mjerenjem neželjenog prometa u razdoblju od 13.05.2011. do 25.06.2011. (datoteke *alert.1305828962, alert.1306456547, alert.1307100756, alert.1309001129*) dobivena je tablica 6.4. Tablica 6.4 prikazuje 10 najgorih AS-a prema srednjoj vrijednosti reputacije.

#	R_{w1}	R_{w2}
1.	AS5391 T-HT	AS18705 RIM
2.	AS18705 RIM	AS43037 SEZNAM-CZ
3.	AS31012 B.net	AS15169 Google Inc.
4.	AS12810 VIPnet d.o.o.	AS21502 NUMERICABLE
5.	AS13046 Iskon	AS12810 VIPnet d.o.o.
6.	AS43037 SEZNAM-CZ	AS31012 B.net
7.	AS15169 Google Inc.	AS5391 T-HT
8.	AS21502 NUMERICABLE	AS13046 Iskon
9.	AS174 Cogent Communication	AS174 Cogent Communication
10.	AS2200 FR-RENATER	AS2200 FR-RENATER

Tablica 6.4: Lista 10 najgorih AS-a za razoblje od 13.05.2011. do 25.06.2011.

Iz tablice je izostavljen *CARNet*, AS unutar kojeg je provedeno mjerene. Faktor opadanja prilikom mjerenja je bio postavljen na 0.5, a početno filtriranje na 60 minuta.

#	S alarmom *1:1000007*		Bez alarma *1:1000007*	
	R_{w1}	R_{w2}	R_{w1}	R_{w2}
1.	AS2108	AS2108	AS2108	AS2108
2.	AS31012	AS18705	AS15169	AS15169
3.	AS18705	AS15169	AS5391	AS174
4.	AS5391	AS21502	AS174	AS5391
5.	AS15169	AS31012	AS47988	AS2914
6.	AS12810	AS2200	AS766	AS766
7.	AS21502	AS12810	AS2914	AS3707
8.	AS47988	AS5391	AS31012	AS15429
9.	AS2200	AS174	AS3707	AS47988
10.	AS174	AS2914	AS15429	AS27925
11.	AS766	AS766	AS9146	AS26627
12.	AS2914	AS47988	AS17672	AS109
13.	AS35549	AS15429	AS27925	AS6849
14.	AS15429	AS3707	AS6849	AS9146
15.	AS3707	AS35549	AS12810	AS9466
16.	AS9146	AS27925	AS26627	AS36375
17.	AS17672	AS26627	AS34594	AS12302
18.	AS27925	AS109	AS109	AS31012
19.	AS6849	AS6849	AS9466	AS243
20.	AS26627	AS9146	AS13046	AS17974

Tablica 6.5: 20 najgorih AS-ova prema srednjoj vrijednosti reputacije

7. Zaključak

Zadaća ovog diplomskog rada bila je izgraditi reputacijski sustav koji će ocjenjivati autonomne sustave na temelju promatranja neželjenog prometa. Kako bi se to postiglo rad je trebao detektirati neželjen promet, otkriti izvore iz kojih dolazi, analizirati modele kažnjavanja neželjenog prometa, modele izračunavanja reputacije autonomnih sustava, te na izmjerenim podacima pokazati ponašanje razvijenog sustava.

Kao rezultat ovog rada nastala je datoteka *unwanted_traffic.rules* koja sadrži Snort pravila za detekciju neželjenog prometa. Unutar datoteke se nalaze pravila od kojih je dio distribuiran sa Snort sustavom, a dio razvijen za potrebe mjerenja ovog rada. Problemi su se pojavili jedino sa pravilom *1:1000007 Unsolicited connection.* To je pravilo uzrokovalo čak 74% zabilježenih alarma te je tako visok udio u ukupnim alarmima odmah povukao sumnju na lažna uzbunjivanja. Zatim je primjećen problem u početnoj fazi mjerenja kada je Snort tek pokrenut, te pravilo još nema uvid u postojeće TCP veze, što uzrokuje lažne alarme. Taj je problem riješen uvođenjem dodatnog početnog filtera, koji je implementiran zasebnom klasom. Međutim ni početno filtriranje nije značajno smanjilo udio alarma *1:1000007.* Pretpostavka je da su dva uzroka tom problemu: Snort pravila ne vide sve pakete trostranog rukovanja, te Snort nije u stanju procesirati sve pakete pristigle na njegov senzor pa dolazi do odbacivanja nekih paketa. U budućim mjerenjima problem se može pokušati riješiti postavljanjem Snort senzora na više lokacija, te eventualno pokretanje Snort sustava na računalu jače procesne snage. Ipak, rezultati mjerenja najgorih autonomnih sustava uzimanjem u obzir alarma *1:1000007*, i njegovim zanemarivanjem, nisu pokazala velika odstupanja, pa su se tako isti autonomni sustavi pojavljivali na obje liste. Datoteka *unwanted_traffic.rules* je baza za detekciju neželjenog prometa te se jednostavnim dodavanjem na kraj datoteke može nadograditi s novim pravilima. Također, u budućim mjerenjima nije isključena mogućnost kombiniranja ovakvog mjerenja temeljenog na sustavu za otkrivanje napada, s nekim drugim metodama detekcije neželjenog prometa, a sve u svrhu dobivanja preciznijih rezultata.

Reputacijski sustav razvijen u ovom radu je u potpunosti podesiv. Moguće je kao ulaz zadavati željene datoteke s alarmima, postaviti proizvoljno vrijeme početnog filtriranja, zadavati period računanja reputacije, birati reputacijsku funkciju te njezin faktor opadanja, kao i mijenjati ocjenu kazne za pojedine alarme. Reputacijski sustav je fleksibilan i u pogledu budućih proširivanja. Jednostavno se mogu dodavati nove reputacijske funkcije, te kazne za nove alarme, ukoliko se proširi način detekcije neželjenog prometa.

Kao dio ovog rada provedena su i mjerenja neželjenog prometa u trajanju od šest dana, na poslužitelju ZEMRIS-a, koji se nalazi unutar autonomnog sustava *CARNet*. Kako bi se dobio kvalitetniji uvid u reputaciju, buduća mjerenja, idealno bi bilo provesti na više mjernih točaka unutar više različitih autonomnih sustava. Također rezultati mjerenja su pokazali da bi reputacijsku funkciju dobro bilo normalizirati s ukupnom količinom prometa, pa bi u budućim mjerenjima idealno bilo mjeriti i ukupnu količinu prometa, a ne samo neželjen promet. Konačni rezultati mjerenja reputacije provedenog u ovom radu daju okvirni uvid u sliku iz kojih autonomnih sustava je pristiglo najviše neželjenog prometa u *CARNet.* U prilog tome ide i činjenica da se rezultati djelomično podudaraju s rezultatima mjerenja reputacije na temelju praćenja DNS sustava [27]. Tako su se šest autonomnih sustava našli na listama 20 najgorih sustava u oba rada. S obzirom da su se reputacije određivale na temelju različitih kriterija, riječ je o dostatno velikoj podudarnosti. Bitno je naglasiti i činjenicu da su se u vrhu lista najgorih sustava često nalazili lokalni pružatelji

Internetskih usluga kao primjerice *T-HT*, *B.net* i *Metronet*. Uzrok tome je očekivano velika komunikacija između lokalnih AS-ova.

Kako bi u budućnosti dobili što bolju sliku o kvaliteti pojedinih autonomnih sustava i prometu koji iz njih pristiže, idealno bi bilo reputaciju mjeriti na temelju više različitih kriterija, različitim metodama, te završnu ocjenu reputacije dati uzimajući u obzir sve mjerene komponente.

8. Literatura

1. A. Habib, M. Khan, B. Bhargava: *Edge-to-edge measurement-based distributed network monitoring*, Technical report, CSD-TR-02-019, Purdue University, Sept. 2002.
2. A. Habib, S. Fahmy, S. R. Avasarala, V. Prabhakar, B. Bhargava: *On detecting service violations and bandwidth theft in QoS network domains*, Journal of Computer Communications, 2003.
3. Y. Breibart, C.Y. Chan, M. Garofalakis, R. Rastogi, A. Silberschatz: *Efficiently monitoring bandwidth and latency in IP networks*, IEEE INFOCOM, Apr. 2001.
4. M. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxon, S. Shenker: *Controlling high bandwidth aggregates in the network*, ACM Computer Communication Review, July 2002.
5. M. Dilman, D. Raz: *Efficient reactive monitoring*, IEEE INFOCOM, Apr. 2001.
6. P. Ferguson: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,* The Internet Engineering Task Force, RFC 2827

   URL: http://tools.ietf.org/html/rfc2827 (11/12/2008)
7. The "Tribe Flood Network" distributed denial of service attack tool
   URL: http://staff.washington.edu/dittrich/misc/tfn.analysis (11/12/2008)
8. The "stacheldraht" distributed denial of service attack tool
   URL: http://staff.washington.edu/dittrich/misc/stacheldrNaht.analysis (11/12/2008)
9. The DoS Project's "trinoo" distributed denial of service attack tool

   URL: http://staff.washington.edu/dittrich/misc/trinoo.analysis (11/12/2008)
10. Andrey Belenky, Nirwan Ansari: On IP Traceback, IEEE Communications Magazine, July 2003.
11. Denail of Service Attack

    URL: http://www.cert.org/tech_tips/denial_of_service.html#3 (14/01/2009)
12. A. Habib, M. Hefeeda, B. Bhargava: *Detecting Service Violations and DoS Attacks*, Internet Society Symposium on Network and Distributed System Security, 2003.
13. Geoffrey M. Voelker, Stefan Savage: *Inferring Internet Denial-of-Service Activity*, USENIX Security Symposium, 2001.
14. Jon Postel, Editor: *Internet Control Message Protocol*, RFC 792, September 1981.
15. M. Handley: *Internet Denial-of-Service Considerations,* The Internet Engineering Task Force, November 2006, RFC 4732
    URL: http://tools.ietf.org/html/rfc4732 (14/01/2009)
16. S. Singh, M. Gyanchandani*: Analysis of Botnet Behavior Using Queuing Theory*, International Journal of Computer Science & Communication, July-December 2010.
17. Steven J. Templeton, Karl E. Levitt: *Detecting Spoofed Packets*, Department of Computer Science U.C. Davis, 2003.
18. R. Rehman: *Intrusion Detection System with Snort*, Prentic Hall 2003.

19. Elwyn Davies: Unwanted traffic. IETF Journal.

URL: http://isoc.org/wp/ietfjournal/?p=172 (25/02/2011)

20. L. Andersson, E. Davies, L. Zhang: *Report from the IAB workshop on Unwanted Traffic,* August 2007, RFC 4948

URL: http://www.ietf.org/rfc/rfc4948.txt (28/02/2011)

21. Paul J. Criscuol: *Distributed Denail of Service*, Computer Incident Advisory Capability, 2000.

22. R. Rehman: *Intrusion Detection Systems with Snort*, Prentice Hall, 2003.

23. M. Rey: *Transmission control protocol,* The Internet Engineering Task Force

URL: http://tools.ietf.org/html/rfc793 (07/03/2011)

24. M. Roesch, C. Green: *Snort users manual*, The Snort project, 2010.

25. C. Ji, A. Elwalid: *Measurement-based network monitoring and inference: scalability and missing information*, IEEE Journal on Selected Areas in Communications 20, 2002.

26. L. Daigle: *WHOIS Protocol Specification,* IETF, 2004., RFC 3912
URL: http://www.ietf.org/rfc/rfc3912.txt (29/08/2011)

27. M. Stublić: *Određivanje reputacije autonomnih sustava temeljeno na praćenju sustava DNS,* ZEMRIS, Fakultet elektrotehnike i računarstva, diplomski rad br. 1879, 2010.

28. F. Tomislav: *Određivanje reputacije autonomnih sustava zasnovano na praćenju elektroničke pošte*, ZEMRIS, Fakultet elektrotehnike i računarstva, diplomski rad br. 1857, 2010.

9. Dodaci

Dodatak A: IDS pravila za detekciju neželjenog prometa

Ideja ovog rada je detektirati neželjen promet na mreži (promet DoS napada i beskoristan promet), odrediti autonomne sustave iz kojih dolazi takav promet, te na odgovarajući način korigirati ocjene reputacije tih autonomnih sustava.

Neželjen promet detektira IDS uz pomoć niza pravila. Svaki paket koji zadovoljava uvjete nekog od pravila se bilježi u dnevniku (*engl. log*).

U nastavku slijedi pregled pravila. Sva pravila su prikazana u tabličnom obliku. Svaka tablica se satoji od četiri reda. *Poruka* je tekst koji se zabilježi u alarmu (*engl. alert*) kada se detektira paket koji se podudara s pravilom. *SID* je identifikacijski broj pravila. *Opis* daje informacije o samom pravilu te slučajevima kada će se pravilo primijeniti. Polje *Jednostavnost napada* govori o tome koliko je jednostavno pokrenuti napad koji je obuhvaćen ovim pravilom.

Poruka	DELETED BAD-TRAFFIC syn to multicast address
SID	1431
Opis	Pod normalnim okolnostima paketi s postavljenom SYN zastavicom ne smiju biti poslani na adresu difuzije u grupi. Ako je napadač lažirao adresu difuzije u grupi, kod napada SYN preplavljivanjem, ovaj promet će biti detektiran. Pravilo je indikator neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	DELETED BAD-TRAFFIC data in TCP SYN packet
SID	526
Opis	Pravilo obuhvaća SYN pakete koji sadrže podatke veće od onog što je normalno očekivano. Pod normalnim okolnostima TCP SYN paketi se razmjenjuju između računala da bi se sinkronizirali TCP slijedni brojevi (*sequence numbers*). SYN paket koji ima datagram veći od 6 okteta može biti indikacija DoS napda ili pokušaja da se izbjegne IDS. Pravilo je indikator neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	SCAN SYN FIN
SID	624
Opis	Detektira pakete sa SYN i FIN postavljenim zastavicama. Većina implementacija TCP/IP stoga će odgovoriti s ACK SYN paketom te na taj način pokazati da je pristup (*port*) otvoren, dok će kod zatvorenog pristupa odgovoriti s ACK RST. Napadač može koristiti ovu metodu za prikupljnje informacija o udaljenom računalu, te na taj način može pokušati otkriti koji su otvoreni i zatvoreni pristupi.
Jednostavnost napada	Srednja. Da bi se pokrenuo ovakav napad, napadač treba alat za slanje paketa sa postavljenim SYN i FIN zastvicama ili treba biti u stanju sam kreirati takve pakete. Prvi slučaj je jednostavan dok drugi zahtjeva malo naprednije vještine.

Poruka	DELETED DOS Land attack
SID	269
Opis	Ovo pravilo detektira pokušaj DoS Land napada. Kod takvog napada, napadač šalje žrtvi lažirani TCP SYN paket s identičnom izvorišnom i odredišnom adresom i s identičnim izvorišnim i odredišnim pristupom. Žrtvino računalo kontinuirano odgovara na paket samom sebi. U nekim slučajevima će se žrtvino računalo srušiti, a u nekim će biti privremeno onemogučeno.
Jednostavnost napada	Jednostavan. Koristeći alate kao što su *nmap* ili *hping* je jednostavno kreirati takav paket.

Poruka	DELETED BAD-TRAFFIC same SRC/DST
SID	527
Opis	Pravilo obuhvaća pakete u kojima su jednaka izvorišna i odredišna IP adresa. U normalnim okolnostima takav promet ne bi trebao postojati na mreži. Ovakvi paketi mogu biti indikator DoS Land napada. Pravilo je pokazatelj neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	SCAN NULL
SID	623

Opis	Detektira TCP segmente bez bez kontrolnih bitova (URG, ACK, SYN, PSH, RST, FIN). Također, slijedni broj (*sequence number*) i broj potvrde (*acknowledgement number*) su postavljeni na 0. Otvoreni pristupi obično neće odgovoriti na ovakvu vrstu paketa dok će zatvoreni pristupi obično odgovoriti s ACK RST. Konkretan odgovor varira u ovisnosti o operacijskom sustavu. Napadač može koristiti ovu metodu za prikupljanje informacija o udaljenom računalu, te na taj način može pokušati otkriti koji su otvoreni i zatvoreni pristupi.
Jednostavnost napada	Srednja. Da bi se pokrenuo ovakav napad, napadač treba alat za slanje paketa bez postavljenih kontrolnih bitova ili treba biti u stanju sam kreirati takve pakete. Prvi slučaj je jednostavan dok drugi zahtjeva malo naprednije vještine.

Poruka	DELETED MISC Large UDP Packet
SID	521
Opis	Polje s podacima unutar UDP paketa je obično manje od 4000 okteta budući da je UDP protokol namijenjen za slanje manjih paketa. Kada se pojave veći paketi to može biti znak nepravilne aktivnosti ili pokušaja DoS napada na udaljeno računalo.
Jednostavnost napada	Jednostavan

Poruka	DOS UDP echo+chargen bomb
SID	271
Opis	Detektira promet između UDP echo pristupa na računalu u štičenoj mreži i UDP generatora znakova (*character generator, UDP chargen service*). Zbog nespojne prirode UDP protokola, jedan paket UDP generatora znakova do echo pristupa rezultira u velikoj količini prometa između ta dva uređaja. Napadač će pronaći računalo koje pruža uslugu UDP generatora znakova te generirati promet između njega i UDP echo usluge.
Jednostavnost napada	Jednostavan

Poruka	DELETED BAD-TRAFFIC udp port 0 traffic

SID	525
Opis	Detektira UDP promet do pristupa 0 koji nije valjan u normalnim okolnostima. Moguće je da se radi o pukušaju utvrđivanja postojanja računala ili grupe računala na nekoj određenoj adresi ili adresnom opsegu. Također je moguće da se radi o DoS napadu na Checkpoint sigurnosnu stijenu. Pravilo je pokazatelj neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	DELETED BAD-TRAFFIC tcp port 0 traffic
SID	524
Opis	Detektira TCP promet do pristupa 0 koji nije valjan u normalnim okolnostima. Moguće je da se radi o pukušaju utvrđivanja postojanja računala ili grupe računala na nekoj određenoj adresi ili adresnom opsegu. Pravilo je pokazatelj neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	DELETED DOS Teardrop attack
SID	270
Opis	Detektira pokušaje DoS Teardrop napada. Teardrop napad koristi propuste u nekim implementacijama TCP/IP stoga. Napadač šalje preklapajuće fragmente paketa. Paket se ne može valjano sastaviti i to uzrokuje privremenu nedostupnost ili rušenje sustava.
Jednostavnost napada	Jednostavan. Postoji puno gotovih alata koji zahtijevaju jako malo znanja od napadača.

Poruka	ICMP PING undefined code
SID	365

Opis	Pravilo detektira slučajeve u kojima vanjski korisnik šalje ICMP ping pakete do unutarnjeg poslužitelja. To može biti indikacija ispitivanja (*scan*) mreže ili uzrok DoS napada ping preplavljivanjem.
Jednostavnost napada	Jednostavan. Alati za ispitivanje mreže i DoS napade ICMP preplavljivanjem su dostupni na Internetu.

Poruka	DOS BGP spoofed connection reset attempt
SID	2523
Opis	Pravilo obuhvaća pokušaje da se iskoristi poznata ranjivost u TCP protokolu koji se koristi kod BGP protokola. Kako se BGP protokol oslanja na TCP protokol, napadač može pokušati pogoditi TCP slijedni broj unutar očekivanog opsega i na taj način poništiti TCP vezu.
Jednostavnost napada	Jednostavan. Metode iskorištavanja su dostupne na Internetu.

Poruka	DELETED BAD-TRAFFIC Unassigned/Reserved IP protocol
SID	1627
Opis	Detektira pakete koji u zaglavlju imaju nedodijeljen ili rezerviran protokol. Pravilo je pokazatelj neovlaštenog korištenja mreže, kompromitiranih sustava ili krivo podešenih mrežnih uređaja.
Jednostavnost napada	Jednostavan

Poruka	DNS SPOOF query response PTR with TTL of 1 min. and no authority
SID	253

Opis	Napadač može prisluškivati DNS upite i pokušati lažirati odgovor na takav upit prije nego što odgovori DNS poslužitelj. Takav lažirani odgovor nije tipičan iz razloga što ne sadrži autoritativni DNS poslužitelj u zapisu. Legitimni DNS odgovor će vjerojatno sadržati ime autoritativnog DNS poslužitelja. Lažirani odgovori imaju postavljeno TTL polje na 1 minutu. Sumnja se da se TTL polje postavlja na kratko vrijeme kako bi se brzo eliminirali dokazi lažiranog paketa. Šteta kod ovakovog lažiranja može varirati od beznačajne (u slučajevima kad je u lažiranom odgovoru IP adresa koja se ne korisit) do teške (u slučajevima kad je u lažiranom odgovoru IP adresa neprijateljskog računala).
Jednostavnost napada	Srednja. Napadač mora biti u stanju prisluškivati DNS upite i lažirati odgovore prije od pravog DNS poslužitelja.