Univerzitet u Sarajevu
ELEKTROTEHNIČKI FAKULTET U SARAJEVU

Saša Mrdović

METOD OTKRIVANJA UPADA ZASNOVAN NA ANALIZI MODELA SADRŽAJA PAKETA U RAČUNARSKOJ MREŽI

(Doktorski rad)

Sarajevo, oktobar 2008.

Rad je izrađen na Elektrotehničkom fakultetu u Sarajevu.

Mentor: Akademik red. prof. dr Branislava Peruničić., dipl.el.ing

Rad ima: 156 stranica

Redni broj:

Sažetak

Savremeno društvo se u potpunosti oslanja na umrežene računare. Sigurnost računara i mreža je od presudnog značaja za njihovu praktičnu primjenu. Sistemi za otkrivanje upada su važna komponenta sveobuhvatnog sistema zaštite. Sistemi za otkrivanje upada zasnovani su na dva glavna pristupa: Prvi pristup je da se prikupljaju potpisi napada i na taj način prepoznaju upadi. Drugi pristup je da se modelira normalno ponašanje i otkrivaju odstupanja od normalnog ponašanja. Sistemi zasnovani na prvom pristupu ne uspjevaju otkriti potpuno nove tipove napada. Sistemi zasnovani na drugom pristupu takođe imaju nedostatak: ako oni koji prave napad znaju model normalnog ponašanja, mogu dizajnirati napade koji imaju model sličan normalnom saobraćaju i tako izbjeći otkrivanje. Zbog ove nesavršenosti sistemi za otkrivanje upada se često inoviraju, ali takođe se prave i novi upadi, pa trka može zauvijek trajati.

Tema ove disertacije je sasvim novi metod dizajna sistema za otkrivanje upada zasnovan na otkrivanju abnormalnog ponašanja saobraćaja. Metod je zasnovan na dobro poznatom i široko prihvaćenom principu da sigurnost sistema ne treba da zavisi o tajnosti njegovog dizajna. Ova osobina ostvarena je uvođenjem ključa koji je različit za svaku implementaciju sistema. Zahvaljujući uvođenju ključa napadači ne znaju šta je model normalnog sistema, pa ne mogu znati kako da izbjegnu otkrivanje napada pomoću mimikrije.

Predloženi metod otkriva upade analizom sadržaja mrežnih paketa. Model normalnog ponašanja je napravljen podjelom sadržaja na riječi. Riječi su definisane kao nizovi bajta između znakova za razdvajanje - separatora. Skup separatora je iskorišten kao ključ. Svaki skup separatora generiše različit model normalnog ponašanje. Upadi se otkrivaju na osnovu ovog modela kao ponašanje koje značajno odstupa od normalnog. Poznavanje metoda detekcije

nije dovoljno da se napravi napad koji liči na normalan saobraćaj, pošto model normalnog saobraćaj zavisi od ključa.

Napravljen je sistem za otkrivanje upada zasnovan na ovom principu i testirana je njegova sposobnost da otkrije savremene napade ubačene u normalan saobraćaj. Testiranje je potvrdilo sposobnost novog sistema da otkrije upade koristeći različite skupove separatora kao ključ.

Abstract

The operation of the contemporary society completely relies on networked computers. Security of computers and networks is of utmost importance for their practical application. Intrusion detection systems are an important component of comprehensive protection system. The intrusion detection systems are based on two main approaches: The first approach is to collect the signatures of attacks, and thus recognize an intrusion. The second approach is to model normal behavior, and detect deviation from normal behavior. The systems based on the first approach cannot detect entirely new types of attacks. The systems based on the second approach also have a flaw: if the creators of an attack know the model of the normal behavior, they may design attacks having a model similar to normal traffic, and thus avoid the detection. Due to this imperfection new intrusion detection systems are frequently created, but new intrusions are developed as well, and this race can last forever.

The thesis topic is a new method to design an intrusion detection based on the detection of abnormal traffic behavior. This method is based on the well known and widely accepted principle that the system security should not depend on the secrecy of its design. This feature is achieved by making the system implementation key dependent. Due to the key introduction the creators of attack cannot know what model of the normal system is, since the intrusion detection avoidance is based on the knowledge of the normal behavior.

The proposed method, uncovers intrusions by the analysis of the network packet payloads. The model of normal behavior is built by dividing payload into words. Words are defined as consecutive bytes between delimiters. Set of delimiters is used as a key. Each set of delimiters generates a different model of normal behavior. Intrusions are recognized when traffic substantially differs from normal. The knowledge of detection method only is not enough

to create attacks that look normal, since the model of normal traffic is key-dependent.

The detection system based on this principle was created, and its capability to test typical contemporary attacks inserted in normal traffic was tested. The testing confirmed the new system's capability to detect intrusion using a variety of delimiter sets as keys.

SADRŽAJ

POPIS ILUSTRACIJA

POPIS TABELA

UVOD

Motiv

Savremeni život teško se može zamisliti bez umreženih računara. Broj podataka koje umreženi računari mogu obraditi i brzina sa kojom to rade olakšavaju obavljanje uobičajenih poslova i uvođenje novih, ranije nezamislivih, usluga. Računari u računarskim mrežama mogu da međusobno razmjenjuju podatke i dijele resurse. Zajednički rad znatno povećava brzinu, broj i kvalitet usluga. Razmjena podataka među računarima dovela je do korjenitih društveno ekonomskih promjena, pa se današnja era naziva doba informacija. U industrijskom dobu ekonomija je bila zasnovana na proizvodnji materijalnih dobara, a sada se smatra da je zasnova na obradi i razmjeni informacija. Prema zaključcima ministarske konferencija OECD iz 2008 „Svjetska ekonomija je danas Internet ekonomija“ . Umrežavanje računara promijenilo je ne samo način privređivanja već i način druženja i povezivanja ljudi. Bitan aspekt društvenog života danas čine Web bazirane društvene mreže kao što je Facebook, MySpace ili hi5. Distribucija i razmjena multimedijalnih sadržaja najvećim dijelom odvija se preko umreženih računara, putem Web lokacija kao što je iTunes ili YouTube. Bilo bi zapravo teško pronaći ljudsku aktivnost koja se više ili manje ne oslanja na korištenje umreženih računara. Prema podacima sa kraja 2006 godine, u evropskoj uniji 56% domaćinstava imalo je računare, a 44% pristup internetu od kuće [1] .

Sa ovakvim obimom primjene umreženih računara u ekonomiji i svakodnevnom životu, sigurnost računara i računarskih mreža ima presudan značaj za funkcionisanje ekonomskog i društvenog sistema. Nažalost, računarski sistemi su izloženi opasnostima koje vrebaju iz mnogih izvora. Neke vrste opasnosti nisu nove - takve su prirodne katastrofe. Novu opasnost predstavlja računarski kriminal koji ima brži rast od svih ostalih oblika kriminala [2]. Ranjivost savremenih računarskih sistema ima mnogobrojne

uzroke. Naslijeđena arhitektura računarskih mreža, prije svega Interneta, nema sigurnosne mehanizme i to je jedan je od glavnih uzroka osjetljivosti računarskih sistema na napade. Pojava novih usluga i novih usavršenijih vrsta softvera, nažalost, stvara nove prilike za zloupotrebu. Brz razvoj protokola i tehnologija ponekad ne ostavlja dovoljno vremena za obezbjeđenje adekvatne sigurnosti. Današnja konvergencija davanja različitih usluga u objedinjenim sistemima stvara nove potencijalno ranjive tačke. Povećavanje broja korisnika i geografsko širenje dodatno otežavaju adekvatnu tehničku i administrativnu kontrolu. Jedan indikator razmjera opasnosti je podatak da je svaki četvrti računar u SAD zaražen nekim oblikom zlođudnog programa [3]. Finansijske štete od zlođudnih programa iznose 67,2 milijarde američkih dolara godišnje u SAD [4].

Očigledno je da postoji potreba za unapređenjima u oblasti sigurnosti računarskih sistema. Pronalazak nekog novog rješenja u ovoj oblasti predstavlja izazovan i društveno veoma koristan, doprinos.

Oblast istraživanja u tezi

Obezbjeđenje sigurnosti računarskih sistema za obradu informacija počinje njihovim dobrim dizajnom, nastavlja se korektnom realizacijom i čuva se pravilnim održavanjem. Kod svakog od ovih koraka mogući su propusti koji uzrokuju sigurnosne slabosti i tako omogućavaju napadačima zloupotrebu sistema.

Prvi korak u odbrani od zloupotreba čine mjere koje sprečavaju da se zloupotrebe uopšte dogode. Ovakve mjere, međutim ne mogu uvijek biti potpuno uspješne. Ako one zakažu, važno je otkriti pojavu zloupotrebe, da bi se mogli poduzeti koraci za njeno zaustavljanje i saniranje njenih posljedica. Jedan od načina otkrivanja zloupotreba je primjena sistema za otkrivanje upada.

Sistemi za otkrivanje upada otkrivaju da je korištenje sistema različito od dozvoljenog. Postoje dvije glavne grupe ovih sistema, prema porijeklu informacija koje otkrivaju upad. Prvu grupu čine sistemi za otkrivanje upada korištenjem podataka iz mrežnog saobraćaja, a drugu sistemi za otkrivanje upada analizom rada računara.

Druga podjela ovih sistema počiva na načinu kako se zaključuje da je došlo do upada. Prva grupa prepoznaje poznate zloupotrebe na osnovu baze podataka u kojoj se nalaze događaji karakteristični za ove zloupotrebe. Ovakvi sistemi uglavnom vrlo tačno prepoznaju poznate napade i većina komercijalnih sistema radi na ovom principu. Njihov glavni nedostatak je što ne mogu da prepoznaju napade kojih nema u bazi. Druga grupa sistema prvo pravi model normalnog ponašanja, pa onda otkriva odstupanje od tog modela nazvano anomalija. Pojava anomalije ukazuje na mogući napad.

Predmet istraživanja u ovoj disertaciji su mrežni sistemi za otkrivanje upada zasnovani na otkrivanju anomalija. Oni imaju najveći potencijal za otkrivanje upada i poboljšanje sigurnosti informacionih sistema. Kako mrežni sistemi štite više računara odjednom, ekonomičniji su od drugih sistema zaštite. Dobro dizajnirani sistemi zasnovani na otkrivanju anomalija mogu otkriti nove napade, i tako mogu duže ostati u upotrebi

Cilj istraživanja i originalni doprinos

Cilj istraživanja je da se nađe novi efikasan način modeliranja normalnog saobraćaja u računarskoj mreži. Ovaj model treba pre svega da omogući što tačniju klasifikaciju saobraćaja. Svi zloćudni mrežni paketi trebaju biti prepoznati, a normalni paketi ne bi trebali biti identifikovani kao zloćudni. Klasifikacija treba da bude dovoljno brza da ne bi ometala normalan protok mrežnih paketa.

Modeliranje mrežnog saobraćaja se razmatra u akademskoj literaturi već duže vrijeme. Međutim, većina radova modele zasniva samo na zaglavljima mrežnih paketa. Ovaj pristup je neadekvatan u savremenim računarskim mrežama jer se napadi sve češće ubacuju u sadržaj paketa. U novijim radovima već se prave modeli sadržaja mrežnih paketa. Medjutim, većina ovih radova analizira samo dio paketa ili radi analizu pojedinačnih bajta sadržaja

Nažalost, napadači mogu da kreiraju napade koji izmiču detekciji i tako prevare većinu savremenih sigurnosnih sistema. Glavna ideja izbjegavanja otkrivanja je da napad treba da veoma liči na normalne događaje, u smislu da se uklapa u model normalnog ponašanja, što se naziva *imitacijski napad*. Ako je poznata metoda otkrivanja upada, uvijek je moguće napraviti napad koji ta metoda neće otkriti jer liči na normalan saobraćaj. Ovaj problem postoji od pojave prvih sistema za otkrivanje upada pa sve do danas. Sve vrijeme se pojavljuju nove metode za otkrivanje upada, a za njima slijede novi napadi koji mogu da ih zaobiđu. Čini se da bi se ova trka mogla nastaviti u nedogled. Ova disertacija ima za cilj da zaustavi ili bar da uspori ovu trku, tako da otežа zaobilaženje otkrivanja napada.

Ključni originalni doprinosi disertacije su:

- Prijedlog metode za modeliranje normalnog mrežnog saobraćaja analizom cjelokupnog sadržaja paketa. Analiziraju se riječi, koje su definisane kao nizovi bajta koji se nalaze izmedju znakova za razdvajanje - separatora;

- Prijedlog potpuno novog dizajna sistema za otkrivanje upada koji je zasnovan na kriptografskom principu da sigurnost sistema ne zavisi od tajnosti dizajna već od tajnosti ključa;

- Prijedlog načina izvedbe sistema za otkrivanje upada koji realizira pomenuti kriptografski pristup;

- Realizacija sistema za otkrivanje upada sa ključevima i njegovo testiranje na stvarnom saobraćaju u koga su ubačeni savremeni napadi.

Struktura disertacije

U prvom poglavlju se uvode osnovni pojmovi i principi sigurnosti informacionih sistema. Kroz objašnjavanje pojma prijetnji sigurnosti i načina realizacije sigurnosti dolazi se do pojma napada i upada i njihove sistematizacije. Sistemi za otkrivanje upada predstavljeni su u drugom poglavlju. Napravljena je njihova klasifikacija, navedeni načini njihovog vrednovanja i prikazana su otvorena pitanja u njihovom dizajnu. Aktuelna problematika sistema za otkrivanje upada sa pregledom radova završava drugo poglavlje. U trećem i četvrtom poglavlju opisano je originalno istraživanje provedeno u radu i dobiveni rezultati. Treće poglavlje prikazuje metodu modeliranja normalnog mrežnog saobraćaja zasnovanu na riječima. Definisane su formule za računanje odstupanja od normalnog ponašanja na osnovu kojih se otkriva pokušaj upada. Na kraju poglavlja prikazana je realizacija metode modeliranja i detekcije, sa rezultatima testiranja provedenim na stvarnom saobraćaju u koga su ubačeni savremeni napadi. Ovi rezultati su upoređeni sa aktuelnim rezultatima drugih istraživača iz ove oblasti. U četvrtom poglavlju opisan je princip otvorenog dizajna sistema čija sigurnost ne zavisi od tajnosti dizajna, već samo od tajnosti ključa, i dat je način njegove primjene u sistemima za otkrivanje upada. Prikazana je konkretna realizacija ove ideje na sistemu iz trećeg poglavlja. Na kraju poglavlja dati su rezultati testiranja sistema i demonstrirana je njegova otpornost na imitacijske napade. Rad završava zaključkom koji rezimira urađeno i iznosi ideje za buduća istraživanja.

1 SIGURNOST INFORMACIONIH SISTEMA

Univerzalna definicija sigurnosti informacionih sistema još ne postoji. Za različite sisteme pojam siguran može imati potpuno različita značenja. Zato svaka organizacija koja ima informacioni sistem ima svoju sigurnosnu politiku. Sigurnosna politika definiše šta je dozvoljena upotreba informacionog sistema, a šta nije. Ono što nije u skladu sa sigurnosnom politikom ugrožava sigurnost informacionog sistema.

Izbor mjera zaštite od ugrožavanja sigurnosti vrši se na osnovu analize rizika. Analiza rizika utvrđuje potrebu za mjerama zaštite i opravdanost njihovog uvođenja.

U ovom poglavlju će biti objašnjene osnovne komponente, prijetnje i načini realizacije sigurnosti. Na kraju će biti navedena sistematizacija upada, kao prijetnje kojom se ova disertacija najviše bavi.

1.1 Osnovne komponente

Osnovne komponente sigurnosti su svojstva informacija koja je potrebno očuvati i procesi kojima se to realizuje.

1.1.1 Povjerljivost, integritet i dostupnost

Sigurnost informacionih sistema zasniva se na ostvarenju *povjerljivosti*, *integriteta* i *dostupnosti* informacija u sistemu.

Informacija je povjerljiva ako je dostupna samo onim koji imaju pravo pristupa na nju. Informacija ima integritet ako je ne može promijeniti neovlašteni subjekt ili ovlašteni subjekt na neovlašteni način. Informacija je dostupna ako joj uvijek mogu pristupiti ovlašteni subjekti.

Matematičke osnove povjerljivosti, zasnovane na kontroli toka informacija, postavili su Bell i LaPadula [5][6]. Njihov model zasnovan je na vojnim

sistemima, u kojima je očuvanje povjerljivosti najvažnije i gdje svaka informacija ima definisani nivo sigurnosti. Za svaki subjekt definisan je njegov sigurnosni nivo. Dva jednostavna pravila ostvaruju povjerljivosti. Prvo kaže da subjekt ne može čitati informacije sa višeg sigurnosnog nivoa od vlastitog. Drugo pravilo je da subjekt ne može premjestiti informacije sa višeg na niži nivo. Ova dva pravila predstavljaju prvi matematički model stvarnih sistema sigurnosti. Model nije dovoljno dobar za širu primjenu jer je strogo prilagođen vojnom pristupu, ali je bio osnova više standarda sigurnosti, od kojih je najpoznatiji *Trusted Computer System Evaluation Criteria* Ministarstva odbrane SAD, poznat kao TCSEC ili *Orange Book* [7].

Prvi matematički model koji opisuje pravila za očuvanje integriteta podataka je Biba model [8]. Slično kao kod Bell - LaPadula modela podaci i subjekti imaju svoje nivoe integriteta. Definisana su dva pravila koja osiguravaju očuvanje integriteta. Prvo, subjekt ne smije čitati podatke sa nižeg nivoa integriteta, jer time može dobiti podatke manjeg integriteta i ugroziti integritet svojih podataka. Drugo, subjekt ne smije pisati podatke na viši nivo integriteta iz istog razloga.

Poslovni sistemi postavljaju prioritet na integritet informacija i imaju nešto fleksibilniju strukturu od vojnih. Novi model koji je pogodniji za poslovne sisteme, dali su Clark i Wilson u [9]. Njihov model pored subjekata i objekata ima i treći element, programe. Pomoću ovog tripleta definišu se takozvane ispravne transakcije, te princip razdvajanja zadataka i subjekata, čime se osigurava integritet.

Dok su povjerljivost i integritet podataka detaljno obrađeni i u literaturi i u praksi, za dostupnost još ne postoji formalan model. Taj aspekt sigurnosti se najčešće previđa kod dizajna sigurnosti sistema. Neadekvatno provođenje zaštite informacija od neovlaštenog pristupa i izmjena može podatke učiniti nedostupnim i ovlaštenim subjektima, što predstavlja direktno narušavanje principa dostupnosti. Naime, ako informacije nisu dostupne onima kojima su

potrebne i kada su potrebne, informacioni sistem ne obavlja svoju funkciju, pa je tako njegova sigurnost narušena.

Tri navedena pojma, povjerljivost, integritet i dostupnost, definišu sigurnost informacija. Načini na koje se oni mogu ostvariti biće obrađeni u nastavku.

1.1.2 Utvrđivanje identiteta, ovlaštenja i evidentiranje

Sigurnost informacionih sistema se obično realizuje kroz tri procesa: *utvrđivanje identiteta*, *provjeru ovlaštenja* i *evidentiranje*.

Utvrđivanje identiteta je provjera da je identitet subjekta zaista onaj koji subjekt kaže da jeste. Provjera ovlaštenje je provjera prava pristupa objektima za subjekt utvrđenog identiteta. Evidentiranje je čuvanje podataka o svim akcijama subjekata.

Utvrđivanja identiteta i provjera ovlaštenja omogućavaju pristup informacijama samo onim subjektima koji na to imaju pravo, što garantuje povjerljivost informacija. Pored toga, procedure utvrđivanja identiteta i provjere ovlaštenja dozvoljavaju promjenu informacija samo ovlaštenim subjektima na ovlašteni način, što obezbjeđuje integritet informacija. Ova dva procesa, dakle, ne samo da omogućavaju odgovarajući pristup informacijama ovlaštenim subjektima, već i sprečavaju pristup neovlaštenim što direktno utiče na dostupnost informacija. Evidentiranje akcija subjekata omogućava provjeru da li je došlo da narušavanja principa povjerljivosti, integriteta i dostupnosti informacija.

Za pristup i promjenu informacija od strane ovlaštenog subjekta vezani su i pojmovi *odgovornosti* i *nemogućnosti poricanja*. Odgovornost i nemogućnost poricanja su svojstva sigurnog informacionog sistema i uglavnom se ostvaruju evidentiranjem akcija. Evidentiranje omogućava da svaki subjekat bude odgovoran za svoje akcije, te da ih ne može poreći.

U literaturi je navedeno više načina kako subjekt može da dokaže svoj identitet. Tri osnova načina koja se navode u svim referencama su:

1. Nešto što subjekt zna (lozinka, PIN, ...)
2. Nešto što subjekt ima (iskaznica, kreditna kartica, ...)
3. Neka osobina subjekta (otisak prsta, glas, ...)

Drugi načini utvrđivanja identiteta navedeni u literaturi su lokacija subjekta [10][11] ili potvrda nekog ko poznaje subjekat [12].

Pouzdanost utvrđivanja identiteta povećava istovremena primjena više navedenih načina, čime se postiže višestruko utvrđivanje identiteta. Dobar primjer dvostrukog utvrđivanja identiteta je korištenje bankovne kartice. Da bi se sa karticom mogao podići novac na bankomatu, klijent mora da ima karticu i treba da zna PIN.

Ovlaštenje daje pravo pristupa nekom subjektu utvrđenog identiteta na neki utvrđeni objekat. Realizacija ovlaštenja se rješava kontrolom pristupa. Postoje dvije tradicionalne metode realizacije kontrole pristupa: diskreciona i obavezna. Obje metode definisane su u TCSEC standardu [7]. *Diskreciona kontrola pristupa* (DAC - *Discretionary Access Control*) omogućava vlasniku objekta da definiše prava pristupa objektu za druge subjekte. *Obavezna kontrola pristupa* (MAC - *Mandatory Access Control*) zasnovana je na ranije pomenutom Bell - LaPadula modelu [5]. Ova metoda ne daje vlasniku nikakva posebna prava u regulisanju prava pristupa jer su ova prava već postavljena u sistemu. Kao praktična alternativa tradicionalnim metodama pojavila se *kontrola pristupa zasnovana na ulozi subjekta* (RBAC – *Role Based Access Control*) [13]. Kod ove metode prava pristupa objektima regulišu se preko uloge, odnosno funkcije u organizaciji. Subjekat koji obavlja određenu funkciju dobiva prava pristupa određenim objektima i resursima organizacije. Ova metoda je znatno

fleksibilnija i lakša za praktičnu realizaciju od diskrecione i obavezne kontrole pristupa.

Najjednostavniji mehanizam realizacije kontrole pristupa je *matrica kontrole pristupa*, koja je opisana prvi put u [14], te preciznije definisana u [15] i [16]. Kolone te matrice predstavljaju objekte, a redovi predstavljaju subjekte. U presjeku reda i kolone su prava pristupa koja subjekt iz tog reda ima objektu iz te kolone. Glavni nedostatak ovih matrica je što je broj njihovih elemenata jednak umnošku broja objekata i subjekata, pa postoji veliki broj praznih polja. Jedan način uštede na memoriji je da se matrica čuva po kolonama, odnosno objektima, i da se čuvaju samo neprazni elementi. Na ovaj način se dobiva takozvana *lista za kontrolu pristupa* (ACL – *Access Control List*) u kojoj se za svaki objekat nalaze svi subjekti koji mu imaju pravo pristupa sa vrstom prava. Drugi način je čuvanje matrice na istom principu, ali po kolonama. Na ovaj način za svaki subjekat dobije se lista objekata kojima subjekt ime neko pravo pristup i vrsta prava pristupa. Ova lista se obično naziva *lista sposobnosti* (*capabilities*) [17]. Iako su ACL ponekad manje sigurne, mnogo više se koriste jer su lakše za realizaciju.

Nakon definisanja osnovnih elemenata informacione sigurnosti te osnovnih procedura za njihovu realizaciju, u nastavku će biti razmotreni pojmovi vezani za ugrožavanje sigurnosti informacija.

1.2 Prijetnje sigurnosti

Obezbjeđenje sigurnosti informacionih sistema zahtjeva određene resurse, što obavezno povlači troškove, pa se uvijek postavlja i pitanje ekonomske opravdanosti Ulaganje u sigurnost ne bi smjelo biti veće od vrijednosti onoga što se štiti, zapravo moralo bi biti dovoljno malo, da bi bilo opravdano. Opravdana visina ulaganja u neku mjeru sigurnosti bi trebala biti izračunata na osnovu rizika da dođe do narušavanje sigurnosti koje ta mjera spriječava i na osnovu štete koja bi tako nastala. Generalno istraživanje rizika je predmet

jedne posebne oblasti ekonomije. Trenutno postoje i standardi koji regulišu upravljanje sigurnosti informacija. ISO/IEC 27001 standard [18] zahtijeva da prvi korak u uspostavljanju sistema sigurnosti bude razmatranje rizika. Po ovom standardu, rizik je prvo neophodno identificirati, zatim ga analizirati i procijeniti, te odrediti na koji način će se sistem nositi sa rizikom. Standard definiše mnogo detalja iz ove oblasti, ali ono što je bitno je da se prilikom definisanja koraka za identifikaciju rizika definiše i okvir za utvrđivanje prijetnji sigurnosti informacija.

Prijetnja je uzrok nekog incidenta koji narušava sigurnost, odnosno povjerljivost, integritet ili dostupnost informacija. Prijetnja može da se ostvari ako postoji neka *slabost* u sistemu upravljanja sigurnošću informacija. *Rizik* je vjerovatnoća da će se prijetnja ostvariti. Eliminacijom bilo prijetnje bilo slabosti može se postići potpuna sigurnost informacija.

Nažalost niti slabosti niti prijetnje se ne mogu u potpunosti ukloniti. Slabosti, koje se u oblasti sigurnosti obično nazivaju i *sigurnosnim propustima*, nastaju ili kao posljedica odstupanja realizacije od specifikacije, ili kao posljedica odstupanja specifikacije od originalnih zahtjeva na sigurnost. *Zahtjevi na sigurnost* su povjerljivost, integritet i dostupnost informacija. Procedure utvrđivanja identiteta, ovlaštenja i evidentiranja predstavljaju *specifikaciju* za realizaciju principa sigurnosti. *Realizacija* je konkretni sistem za zaštitu sigurnosti informacija [19]. Postoji brojna literatura koja teoretski razmatra zašto praktične realizacije ne mogu u potpunosti odgovarati specifikacijama, koje opet ne mogu u potpunosti obuhvatiti zahtjeve. Razlozi koji se navode mogu biti ljudski faktor [20], ekonomski faktor [21], kao i princip da ne postoji testiranje koje bi garantovalo da neki softver nema grešaka [22]. Dobar primjer nemogućnosti eliminacije sigurnosnih propusta je preko 22 000 propusta u bazi sigurnosne kompanije Symantec [23]; oko 28000 sigurnosnih propusta u CVE bazi [24]; te preko 36 000 propusta po CERT (*Computer*

Emergency Response Team) statistikama [25]. Može se navesti i najnoviji podatak o 17 novih propusta dnevno u CVE bazi [24].

Prijetnja je mogućnost narušavanja sigurnosti. Akcije kojima se prijetnje ostvaruju i od kojih se treba štititi nazivaju se napadi. Prijetnje se mogu podijeliti u četiri kategorije [26]:

- Otkrivanje – neovlašten pristup informacijama
- Prevara – prihvatanje pogrešnih podataka
- Smetnja – prekidanje ili sprečavanje normalnog rada
- Uzurpacija – neovlaštena kontrola nekog dijela sistema

Prijetnje dolaze iz različitih izvora, imaju različite pobude, nepredvidive su, a nekad za njih nema posebnog razlog. Zato ih je teško eliminisati u potpunosti. Prirodna katastrofa je primjer prijetnje koja ugrožava sigurnost, prije svega dostupnost, a ne može se niti predvidjeti niti eliminisati. Ljudski postupci su također nepredvidiva, ali česta, prijetnja sigurnosti. Ljudi ugrožavaju sigurnost nekad namjerno, ali jos češće iz neznanja. Teorija upravljanja rizikom bavi se postupcima koje treba provesti poslije realizacije neke prijetnje.

Pošto je nemoguće u potpunosti ukloniti rizik da će prijetnja iskoristiti slabost, razvijena je metodologija upravljanja rizikom. Upravljanje rizikom sastoji se od tri koraka [27]:

- Analiza rizika
- Proračun rizika
- Tretman rizika

Analiza rizika je sistematična identifikacija izvora rizika i procjena moguće štete. Analiza rizika daje osnovu za proračun rizika i njegov tretman. *Proračun rizika* je proces poređenja procijenjenoga rizika sa zadanim *kriterijem rizika*. Kriterij je skalar koji određuje važnost rizika u odnosu na postavljene prioritete. Prioriteti mogu biti, između ostalog, finansijskog, pravnog ili društvenog karaktera. *Tretman rizika* je izbor i provedba mjera za promjenu rizika. Ove mjere mogu biti: izbjegavanje, optimizacija, te prenos i prihvatanje rizika. *Izbjegavanje rizika* je odluka da se ne uključi u neku akcija ili da se povuče iz neke situacije, ako se procjeni da su skopčane sa određenim rizikom. *Optimizacija rizika* je postupak minimiziranje negativnih i maksimizacije pozitivnih posljedica. *Prenos rizika* je prenos tereta rizika na druga lica putem osiguranja ili sličnog ugovora. *Prihvatanje rizika* je odluka da se ne poduzimaju nikakve mjere smanjenja rizika. Ova odluka se donosi na osnovu proračuna posljedica rizika i proračuna odnosa troškova tretmana rizika i troškova realizacije rizika. [28]

Optimizacija je jedini postupak koji se bavi smanjenjem posljedica rizika putem provedbe različitih mjera. Ove mjere su zapravo realizacija sigurnosti o kojoj će biti riječi u nastavku.

1.3 Realizacija sigurnosti

Sigurnost informacija postiže se realizacijom *kontrola*. Kontrole se obično dijele na administrativne, logičke i fizičke. *Administrativne* kontrole su primarno politike i procedure uspostavljene da bi se definisalo dozvoljeno ponašanje i načini sprovođenja politike. *Tehničke*, ili *logičke*, kontrole su uređaji, procesi, protokoli i druge mjere za zaštitu informacija. *Fizičke* kontrole su uređaji i sredstva za fizičku kontrolu pristupa i zaštitu dostupnosti informacija.

Drugi način podjele kontrola sigurnosti je na *preventivne*, *detektivne* i *korektivne*. Preventivne kontrole sprečavaju narušavanje sigurnosti. Kada ove kontrole ne uspiju spriječiti neko narušavanje sigurnosti, detektivne kontrole otkrivaju da

je došlo do narušavanja sigurnosti. Korektivne kontrole koriguju posljedice narušavanje sigurnosti, tako da prekinu događaj koji je narušio sigurnost, ili tako da povrate sistem na stanje koje je bilo prije događaja ili, što je najbolje, tako da omoguće da sistem nastavi da sigurno funkcioniše i tokom ovakvog događaja.

Praktična realizacija sigurnosti kreće od utvrđivanja sigurnosne politike sistema. *Sigurnosna politika* sistema je ustvari iskaz o tome šta je dozvoljeno, a šta nije [10]. Sigurnosna politika sistema uzima u obzir sve relevantne aspekte povjerljivosti, integriteta i dostupnosti, tako što formalno pobrojava sva željena svojstva sistema. Ove formalne navode koriste dizajneri i izvođači sistema da dokažu da sistem ima tražena svojstva, čime se potvrđuje da je sistem siguran. Sigurnosna politika sistema definiše *dozvoljena stanja sistema*. Stanje sistema se definiše kroz ponašanja, akcije i ovlaštenja utvrđujući ovlaštene korisnike i dozvoljenu upotrebu. Često korišteni primjeri su vojna i državna politika sigurnosti koje imaju naglasak na zaštiti povjerljivosti; te komercijalna i poslovna, sa naglaskom na zaštitu integriteta.

Sigurnosna politika sistema provodi se pomoću sigurnosnih *mehanizama*. Sigurnosni mehanizam je metod, sredstvo ili procedura koji provodi neki dio sigurnosne politike [10]. Sigurnosni mehanizmi mogu biti *tehnički* i *netehnički*. Sredstva su primjer tehničkih, dok su procedure primjer netehničkih sigurnosnih mehanizama. Sigurnosnih mehanizama ima mnogo, a mogu se grupisati u mehanizme za utvrđivanje identiteta; mehanizme za kontrolu pristupa; i mehanizme za očuvanje dostupnosti. Jedan od osnovnih principa dizajna u računarskim naukama je odvajanje mehanizama od politike, u smislu da mehanizmi ne bi trebali diktirati ili ograničavati politiku [29]. Politika sistema, naime, ne smije da zavisi i ne smije da se pravi polazeći od nekog konkretnog skupa mehanizama jer politika definiše šta je dozvoljeno i šta nije, a ne kako će se to sprovesti u djelo. Ovaj princip je originalno postuliran za računarske nauke, a može se direktno primjeniti i na sigurnost informacija.

Dizajn i realizacija sigurnosnih mehanizama zasnovani su na osnovnim *principima dizajna sigurnosti*. Prema tim principima sigurnosni mehanizmi ograničavaju prava subjekata, a treba da budu jednostavni. Jednostavnost znači da mehanizmi treba da budu lako razumljivi, što smanjuje mogućnost grešaka i odstupanja od politike. Osam osnovnih principa su [17]:

1. Jednostavnost – sigurnosni mehanizam treba biti što je moguće jednostavniji.

2. Restriktivnost – ako subjektu nije eksplicitno dan pristup objektu, pristup mu mora biti onemogućen.

3. Obaveza provjeravanja – svaki pristup objektima mora biti provjeren da bi se potvrdilo da je dozvoljen.

4. Otvorenost dizajna – sigurnost mehanizma ne treba da zavisi od tajnosti njegovog dizajna ili načina realizacije.

5. Razdvajanje privilegija – pristup sistemu se ne treba dati samo na osnovu ispunjenja jednog uslova, ako je izvodljivo.

6. Minimizacija privilegija – subjekt treba imati samo one privilegije koje su mu potrebne da obavi svoj zadatak.

7. Minimizacija broja zajedničkih mehanizama – mehanizmi koji se koriste za pristup resursima ne trebaju biti u principu dijeljeni među subjektima.

8. Psihološka prihvatljivost – sigurnosni mehanizam ne treba da otežavaju pristup resursu.

Ovim je završen pregleda osnovnih komponenti sigurnosti, kao i prijetnji sigurnosti i načina realizacije sigurnosti. Preostalo je još da se u ovom

poglavlju obrade i sistematiziraju upadi kao glavne prijetnje sigurnosti kojima se rad bavi.

1.4 Sistematizacija upada

Prije same sistematizacije potrebno je definisati pojam upada i objasniti razliku između napada i upada. Upadi su oni napadi koje preventivni sigurnosni mehanizmi nisu zaustavili. U literaturi ova razlika često nije jasna. Recimo po jednoj definiciji upad je bilo koji skup akcija koji pokušava ugroziti povjerljivost, integritet ili dostupnost [30]. Druga definicija kaže da je pokušaj zaobilaska sigurnosnih mehanizama upad [31].

Upade prave spoljni napadači; ovlašteni korisnici sistema koji pokušavaju dobiti privilegije veće od onih za koje su ovlašteni; te ovlašteni korisnici koji zloupotrebljavaju svoje privilegije. Ponekad se kaže da su upadi napadi izvana, a zloupotrebe napadi iznutra. U ovom radu upadi se definišu kao podskup napada koji narušava sigurnosnu politiku sistema, a nije spriječen preventivnim mehanizmima.

Svaki upad je rezultat napada, pa je sistematizacija napada i upada zajednička. Postoji brojna literatura koja se bavi problematikom sistematizacije upada – napada [32][33][34][35]. Najčešće korištena podjela napada je data u [36]:

- Izviđanje – testiranje potencijalne mete radi prikupljanje informacija. Ovi napadi su česti i uglavnom ne prouzrokuju štetu, osim ako se sa njima otkrije slabost koja se kasnije iskoristi.

- Onemogućavanje pružanja usluge (*DoS*) – napad koji ima za cilj da poremeti normalan rad, tako da napadnuti računar prestane da radi ili da se blokira mrežni saobraćaj.

- Pristup sa daljine (*R2L*) – napad u koma neovlašteni korisnik zaobilazi normalan proces provjere identiteta i izvršava komandu na napadnutom računaru.

- Podizanje privilegija (*U2R*) – napadi kojima ovlašteni korisnik sistema zaobilazi normalan proces provjere identiteta da bi dobio privilegije drugog korisnika, najčešće onog sa najvećim privilegijama.

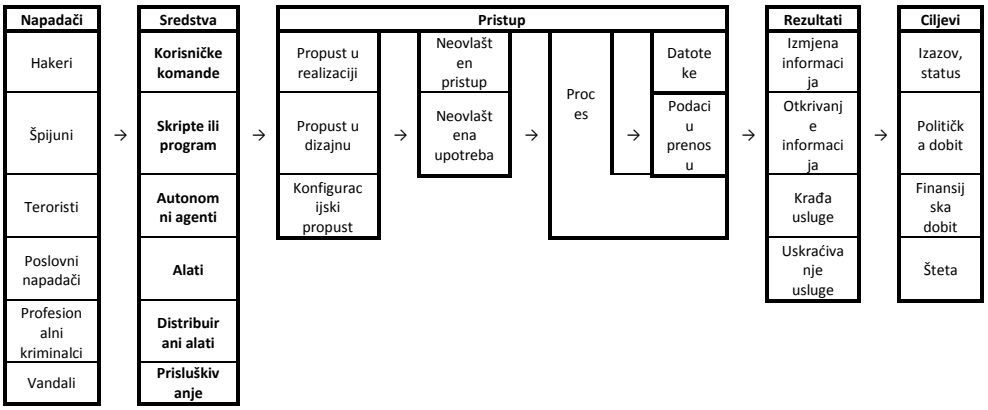
Slijed događaja prilikom napada može se razdvojiti u tri faze [37]:

1. Vrijeme prije napada kada napadač vrši izviđanje sistema u potrazi za slabostima koje bi mogao iskoristiti.

2. Izvođenje napada na pronađenu slabost.

3. Radnje koje slijede uspješan napad.

Na kraju ovog poglavlja data je sistematizacija i povezivanje svih faktora u sekvenci ugrožavanja sigurnosti koju koristi CERT. Napadači imaju ciljeve i ostvaruju ih koristeći sredstva koji im omogućavaju pristup kojim postižu rezultate koji ostvaruju cilj.



Slika 1. CERT sistematizacija [38]

2 SISTEMI ZA OTKRIVANJE UPADA

Kako je ranije rečeno u poglavlju o prijetnjama sigurnosti, ne može se očekivati potpuna eliminacija slabosti i prijetnji. Izvjesni rizik po sigurnost sistema uvek postoji, ali ga je potrebno umanjiti. Rizik se umanjuje sa četiri pomenute mjere. Tri od ovih mjera, izbjegavanje, prenos i prihvatanje rizika, ne podrazumjevaju nikakvu provedbu sigurnosnih kontrola. Obično one nisu dovoljne da ostvare sigurnost, već treba provesti i optimizaciju rizika. Optimizacija rizika je poduzimanje koraka za minimizaciju rizika putem provedbe sigurnosnih kontrola. Idealna situacija bi bila kada bi preventivne kontrole bile dovoljne da se spriječi ugrožavanje sigurnosti. Međutim [39]:

1. Gotovo svaki postojeći sistem ima sigurnosne propuste koji ga čine podložnim upadima i drugim formama zloupotrebe. Pronalaženje i otklanjanje svih ovih nedostataka nije izvodljivo iz tehničkih ili ekonomskih razloga;

2. Postojeći sistem sa znanim nedostacima nije lako zamijeniti sa sigurnijim sistemom – uglavnom zato sto postojeći sistem ima neke privlačne osobine koje sigurniji sistem nema ili za zamjenu nema dovoljno sredstava;

3. U opštem slučaju razvoj apsolutno sigurnog sistema je iznimno težak, ako ne i nemoguć;

4. Čak i najsigurniji sistem je podložan zloupotrebama od strane ovlaštenih korisnika putem zloupotrebe privilegija.

Zbog ovih činjenica potrebne su detektivne kontrole, odnosno sistemi za otkrivanje upada.

2.1 Uvod

Anderson [40] je 1980. prvi formalno iznio ideju otkrivanja zloupotreba pomoću analize podataka o radu sistema. Osnovna ideja njegove studije je da je moguće napraviti karakteristiku upotrebe računarskog sistema posmatranjem *parametara*. Parametri su podaci o radu sistema. Za parametre je moguće utvrditi „normalne" opsege vrijednosti koji čine *karakteristiku*. Odstupanja od karakteristike ukazuju na potencijalnu zloupotrebu sistema. Studija ukazuje i na moguće poteškoće prilikom analize podataka o radu sistema uzrokovane velikom količinom podataka.

Prvu formalnu specifikaciju modela sistema za otkrivanje upada dala je Denning [39]. Metoda otkrivanja upada zasnovana je na otkrivanju neuobičajenih događaja. Predložena je automatizacija procesa. Osnove za model dolaze iz rada autora na stvarnom ekspertnom sistemu za otkrivanje upada u realnom vremenu IDES [41].

Na ovim idejama nastao je niz sistema za otkrivanje upada. Ovi sistemi su vremenom evoluirali i prilagođavali se razvoju računarskih sistema i razvoju novih napada, u skladu sa poznatom izrekom da sigurnost nije stanje nego proces. Njihovi pravci razvoja, klasifikacija, načini vrednovanja i aktuelni problemi biće razmatrani u ostatku ovog poglavlja. Prije toga biće navedena četiri potrebna cilja upotrebe sistema za otkrivanje upada [10]:

1. Otkrivanje poznatih i nepoznatih upada; izazvanih van ili unutar sistema

2. Pravovremeno otkrivanje upada u vremenu bliskom realnom;

3. Prikazivanje rezultata analize u jednostavnom lako razumljivom formatu koji omogućavaju čovjeku da utvrdi da li je zaista došlo do upada ili ne.

4. Tačnost, što znači da se normalni događaji ne proglašavaju upadima, a pogotovo da se upadi ne proglašavaju normalnim događajima.

2.2 Klasifikacija sistema za otkrivanje upada

Dva najčešća načina podjele sistema za otkrivanje upada su:

- Po lokaciji sa koje se prikupljaju informacije na osnovu kojih se donosi odluka da li je došlo do upada ili ne. Podaci se mogu skupljati na računaru ili na nekom mrežnom segmentu

- Po načinu na koji se na osnovu prikupljenih informacija donosi odluka da li je došlo do upada ili ne. Odluka se može donijeti ili tako da se u skupljenim podacima prepozna već poznati potpis napada, ili tako da se prepozna anomalija u ponašanju.

2.2.1 Sistemi za otkrivanje upada na računaru

Ovi sistemi su instalirani na samom računaru sa koga prikupljaju informacije na osnovu kojih donose odluku da li je došlo do upada u taj računar ili ne. Ustvari, prve ideje [40] i izvedbe [41] sistema za otkrivanje odnosile su se samo na računare. Inicijalno su informacije prikupljane iz zapisa o događajima u sistemu (*logs*), a kasnije je počelo posmatranje i sistemskih poziva, upotrebe resursa, podizanja privilegija, te promjena sistemskih datoteka.

Prednost ovakvih sistema je da imaju uvid u sve događaje na računaru koga štite i da uglavnom mogu vrlo precizno utvrditi da li se radi o upadu ili ne. Njihov veliki nedostatak je da štite samo jedan računar. To znači da treba instalirati, održavati i nadzirati sisteme na svim računarima koji traže zaštitu. Pošto u nekoj mreži može biti mnogo računara, to rješenje nije ekonomično. Takođe, takvi sistemi koriste resurse samog računara koga štite, pa njihov rad može ometati normalne funkcije računara. Kako je umrežavanje postalo

preovladavajući način korištenja računara i mreža glavni način razmjene informacija, većina pokušaja upada danas dolazi preko mreže. Ovo je dovelo do razvoja i sve većeg korištenja mrežnih sistema za otkrivanje upada.

2.2.2 Mrežni sistemi za otkrivanje upada

Mrežni sistemi posmatraju saobraćaj na nekom mrežnom segmentu i, na osnovu mrežnih paketa koje vide, donose odluku da li je u toku napad na neki računar u mreži. Prvi mrežni sistem za otkrivanje upada [42] pojavili su se nekoliko godina nakon sistema za otkrivanje upada na računaru. Prve analize [43] ukazale su na njihove velike potencijalne prednosti, ali i na poteškoće pri korištenju mrežnih paketa.

Prednost ovih sistema je što omogućavaju nadzor većeg broja umreženih računara sa samo jednim sistemom, koga je lakše instalirati, održavati i nadzirati nego veliki broj pojedinačnih sistema. Međutim, oni mogu otkriti samo upade koji dolaze preko mreže, ali ne i napade koji dolaze direktno putem konzole računara koji je napadnut. Takođe mrežni sistemi mogu samo pretpostaviti (mada uglavnom prilično tačno) kakav će efekat imati neki skup paketa na odredišni računar i na osnovu toga upozoriti na pokušaj upada. Još jedan problem u njihovoj primjeni je *fragmentacija*, to jest podjela većih paketa na manje. Fragmentacija mrežnih paketa se vrlo često koristi, ali je svi računari i operativni sistemi, ne interpretiraju isto, što predstavlja potencijalni problem u analizi. Konačno, šifriranje mrežnog saobraćaja znatno umanjuje, ako ne i onemogućava, sposobnost mrežnih sisteme za otkrivanje upada.

Savremeni sistemi za otkrivanje upada u poslovnim izvedbama uglavnom koriste distribuirani pristup, prvi put predložen u [44]. Ovaj pristup uvezuje, kombinuje i korelira informacije koje dolaze iz raznih sistema za otkrivanje upada. Takvi sistemi mogu štititi računare ili mrežne segmente. Tako se dobija bolje pokrivanje na nivou štićene organizacije, a otkrivanje upada je brže i tačnije.

2.2.3 Sistemi za otkrivanje upada na osnovu anomalija

Princip rada kod ovih sistema je da se upad prepoznaje tako što se otkriju neuobičajeni događaji u sistemu Prećutno se pretpostavlja da se može napraviti *model* koji opisuje normalne događaje. Model čini konačan skup normalnih raspona određenih parametara. U slučaju upada, vrijednosti nekih od tih parametara izlaze iz dozvoljenog raspona, i tako se utvrđuje *anomalija*. Tu ideju je prvi predložio Anderson u već pomenutom prvom radu na temu otkrivanja upada [40]. Prvi sistem [41] i prvi model [39] sistema za otkrivanje upada bili su zasnovani na otkrivanju anomalija.

Njihova najveća prednost je da mogu otkriti sasvim nove, do sada nepoznate, napade. Kako takvi sistemi zapravo i ne znaju šta su napadi, već samo znaju šta je uobičajeno ponašanje, nisu im potrebni podaci o postojećim napadima, te im zbog toga nije potrebno ažuriranje. Njihov nedostatak je što mogu neke normalne, ali nove i do sada neviđene događaje, pogrešno prepoznati kao pokušaje upada. Pravljenje modela normalnog ponašanja nije lako, a ni jednostavno, a veoma je važno jer od modela uvelike zavisi uspješnost sistema. Svakom takvom sistemu potreban je period učenja u kom se definiše normalno ponašanje. U periodu učenja ne bi trebalo da bude pokušaja upada u sistem, što je veoma teško obezbijediti. Zbog toga su ovakvi sistemi podložni poznatim problemima mašinskog učenja navedenim u [45].

2.2.4 Sistemi za otkrivanje upada na osnovu potpisa napada

Za razliku od prethodnih, ovi sistemi pokušavaju napraviti model zloćudnih događaja. Model se pravi koristeći *pravila*. Pravila su iskazi koje su tačni za neki određeni događaji. Iskazi mogu biti jednostavni poput skupa odredišnih adresa mrežnih paketa ili složeni poput niza sistemskih poziva. Model zloćudnih događaje je skup pravila koji opisuju događaje koji se dešavaju tokom pokušaja upada. Ta pravila se obično nazivaju potpisi napada. U pomenutom radu [42], posvećenom otkrivanju upada preko anomalija prvi

put se pominje i otkrivanje upada putem prepoznavanja poznatih potpisa napada.

Prvi rad [46] koji je postavio teoretske osnove ovakvog pristupa i koji je napravio prekretnicu u oblasti sigurnosti, pojavio se četiri godine kasnije 1994. godine. Nakon pojave ovog rada primat u komercijalnim izvedbama su preuzeli sistemi zasnovani na otkrivanju potpisa poznatih napada.

Prednost ovog sistema je tačno prepoznavanje napada, na osnovu potpisa. Ovi sistemi neće normalno ponašanje proglasiti napadom. Njihov dizajn je jednostavan, jer samo porede događaje sa potpisima. No kako potpisi postaju sve kompleksniji, ovaj tip sistema postaje sve složeniji. Njihov najveći nedostatak je da se novi napadi ne mogu prepoznati, čak i kada predstavljaju samo modificirane verzije napada sa poznatim potpisima. Zato je neophodno neprekidno ažuriranje skupa potpisa sa novo otkrivenim napadima. Kako se novi napadi javljaju u sve kraćim razmacima, ažuriranje postaje sve teže, a može postati i neizvodljivo u realnom vremenu.

U komercijalnim izvedbama se sistemi za otkrivanje upada na osnovu potpisa mnogo više koriste od sistema za otkrivanje anomalija [47], mada postoje izvedbe u kojima se kombinuje otkrivanje anomalija sa otkrivanjem napada na osnovu potpisa. U novije vrijeme predloženo je da se otkrivanje anomalija koristi za pravljenja potpisa [48][49].

2.3 Vrednovanje sistema za otkrivanje upada

Evaluacija sistema za otkrivanje upada nije ni jednostavan ni jednoznačan problem. Postoji veći broj izmjerivih kriterija koji se koriste za vrednovanje ovih sistema kao što su [50]:

- Broj napada koje može otkriti pod idealnim uslovima;
- Vjerovatnoća lažnih uzbuna;

- Vjerovatnoća otkrivanja upada;
- Otpornost na napade usmjerene protiv samog sistema za otkrivanje upada;
- Sposobnost da se brzo obradi veliki broj događaja - propusnost;
- Sposobnost pravljenja korelacija među događajima;
- Sposobnost da se otkrije do sada nepoznati napad;
- Sposobnost da tačno identifikuje o kom se napadu radi;
- Sposobnost da ustanovi da li je napad uspio.

Postoje i druge, nekvantitativne, karakteristike kao što su lakoća korištenja, lakoća održavanja i način postavljanja, te potrebe za resursima.

Četiri najjednostavnije i najčešće korištene karakteristike, koje se primijenjuju i u drugim oblastima donošenja binarnih odluka, a vuku korijene iz teorije statističkih grešaka [51], su:

- Tačno otkrivanje (TP – *true positives*) – sistem upozorava na stvarni pokušaj upada;
- Pogrešno otkrivanje (FP – *false positives*) – sistemu upozorava na pokušaj upada, kada takav pokušaj ne postoji;
- Tačno neotkrivanje (TN – *true negatives*)– sistem ne upozorava i nema pokušaja upada,
- Pogrešno neotkrivanje (FN – *false negatives*) – sistem ne upozorava iako postoji pokušaj upada.

Tačno otkrivanje i tačno neotkrivanje su očito poželjne karakteristike, jer sistem treba da upozorava samo u slučaju stvarnog pokušaja upada i ne treba da upozorava kada pravi pokušaj upada ne postoji.

Pogrešno otkrivanje negativno utiče na vjerovanje sistemu za otkrivanje upada, jer nakon više lažnih uzbuna moguća je tendencija operatora da ignorišu buduća upozorenja sistema. Sistemi zasnovani na otkrivanju anomalija imaju tendenciju da imaju veći broj pogrešnih otkrivanja. Naime, kod ovih sistema nova ponašanja i novi događaji, koji ne moraju biti pokušaji upada, ali odstupaju od modela normalnog ponašanja, smatraju se anomalijama i pogrešno proglašavaju pokušajima upada.

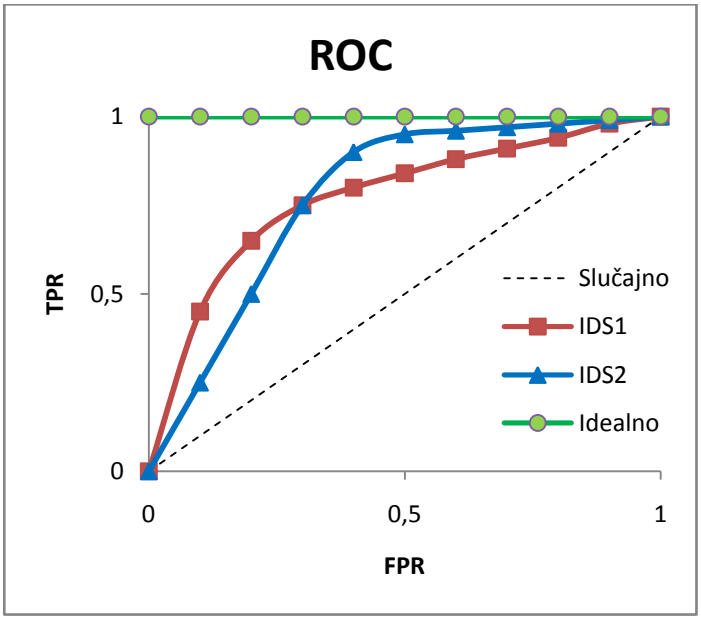
Pogrešno neotkrivanje može imati katastrofalne posljedice, jer daje lažni osjećaj sigurnosti da je sistem zaštićen i da otkriva upade, kada zapravo uopšte ne upozorava na postojeće pokušaje upada. Sistemi zasnovani na prepoznavanju potpisa poznatih napada pate od pogrešnog neotkrivanja. Ovi sistemi nisu u stanju prepoznati nove pokušaje upada čiji potpis ne poznaju. Oni čak nisu u stanju prepoznati nove verzije poznatih napada, ako se njihov potpis, ponašanje na osnovu koga se prepoznaju, dovoljno razlikuje od originalne verzije napada.

Navedene veličine daju osnovne informacije o uspješnosti sistema za otkrivanje upada, ali one mogu dati informaciju, odnosno rezultat, samo za jednu provjeru. Uobičajeno je grafičko predstavljanje uspješnosti sistema za otkrivanje upada uz pomoć takozvanih *ROC krivih* (*Receiver Operating Characteristics*). ROC krive [52] dolaze iz teorije prepoznavanja signala i razlikovanja signala od šuma, a sada se koriste u različitim oblastima istraživanja koja se bave problematikom klasificiranja i donošenja odluka, kao što je medicinska dijagnostika, mašinsko učenje i rudarenje podataka. To je korisna tehnika za organizaciju klasifikatora i vizuelizaciju njihovih performansi [53].

ROC krive za sisteme za otkrivanje upada imaju na apscisi *intenzitet pogrešnog otkrivanja* (FPR – *false positive rate*), a na ordinati *intenzitet tačnog otkrivanja* (TPR – *true positive rate*).

$$FPR = FP / (FP + TN) \qquad TPR = TP / (TP + FN)$$

gdje je TP broj tačno otkrivenih upada, FP broj pogrešno identifikovanih normalnih događaja kao upad, TN broj tačno identifikovanih normalnih događaja i FN broj upada koji nisu otkriveni.



Slika 2. Primjeri ROC krive

Jedna tačka na ROC grafu predstavlja uspješnost sistema za dati nivo osjetljivosti kao odnos između intenziteta tačno otkrivenih upada i intenziteta pogrešno otkrivenih upada. Idealan slučaj je kada je tačka u gornjem lijevom uglu (0,1) gdje su otkriveni svi upadi, bez lažnih uzbuna. ROC kriva se dobije variranjem osjetljivosti sistema na osnovu koje se na graf nanose tačke. Ekstremni slučajevi su nulta osjetljivost kada sistem ne registruje ništa, što

odgovara tački (0,0); te potpuna osjetljivost kada sve registruje kao upad, što odgovara tački (1,1). Iz ovog razloga sve ROC krive polaze iz (0,0) ka (1,1).

ROC kriva daje dobar vizuelni prikaz uspješnosti nekog sistema za otkrivanje upada. Kada je potrebna neka skalarna vrijednost preko koje ce se vršiti rangiranje, koristi se *površina ispod ROC krive* (AUC – *Area Under Curve*) [54]. Kada bi se napad otkrivao potpuno slučajnim pogađanjem ROC bi bio dijagonala od (0,0) do (1,1). Prema tome minimalni AUC je 0,5, a maksimalni 1. Naravno, što je veći AUC sistem je bolji. Važna statistička osobina AUC je da predstavlja vjerovatnoću da će sistem slučajno izabranom upadu dati veću vrijednost nego slučajno izabranom normalnom događaju [53].

Potrebno je napomenuti da sistem sa većim AUC ne mora u praktičnoj upotrebi biti bolji. Konkretna realizacija sistema može imati ograničenja na minimalni nivo otkrivanja ili maksimalni nivo lažnih uzbuna koje mogu definisati područje rada ili tačku na ROC. U ovom slučaju je bolji je onaj sistem koji je u tom području ili tački bliži idealnoj tački (0,1).

Pored ROC krive koja se najčešće koristi za ocjenu uspješnosti sistema za otkrivanja upada, potrebno je pomenuti da postoje i drugi mjere i pristupi za analizu klasifikacionih problema. Veličine preciznost (*precision*) i odziv (*recall*) uvedene su inicijalno za proučavanje uspješnosti sistema za pretraživanje informacija [55]. Kod sistema za binarno donošenje odluka preciznost se definiše kao odnos broja tačno otkrivenih uzoraka u odnosu na ukupan broj otkrivenih uzoraka. Ukupan broj otkrivenih uzoraka je zbir tačno i pogrešno otkrivenih uzoraka.

$$\text{Preciznost} = TP / (TP + FP)$$

Kod binarnog odlučivanja odziv, koji se tada naziva i osjetljivost (*sensitivity*), se definiše kao odnos broja tačno otkrivenih uzoraka u odnosu na ukupan broj

uzoraka koje je trebalo otkriti. Ukupan broj uzoraka koje je trebalo otkriti je zbir tačno otkrivenih i pogrešno neotkrivenih uzoraka.

$$\text{Odziv} = TP / (TP + FN)$$

Može se primjetiti da je odziv isto što i prethodno definisani intenzitet tačnog otkrivanja koji se koristi za konstrukciju ROC krive.

Kriva koja na apscisi ima odziv, a na ordinati preciznost naziva se u literaturi PR (Precision-Recall) kriva [56]. Ova kriva predlagana je kao alternativa ROC krivoj za slučajeva nesimetrične (*skew*) distribucije [57] [58].

Postoje i druge veličine koje se koriste za iskazivanje uspješnosti sistema, od kojih će ovdje biti pomenute još i tačnost (*accuracy*), F-mjera (*F-measure*) i specifičnost (*specifity*) [53]. Ove veličine se takođe računaju na osnovu četiri osnovne mjere tačnog i pogrešnog otkrivanja i neotkrivanja:

$$Tačnost = \frac{TP + TN}{(TP + FN) + (FP + TN)}$$

$$F - mjera = \frac{1}{\frac{1}{Preciznost} + \frac{1}{Odziv}}$$

$$Specifičnost = \frac{TN}{FP + TN}$$

U ovo razmatranje se može uključiti trošak koji je izazvan pogrešnim otkrivanjem, odnosno pogrešnim neotkrivanjem. U tom slučaju ako sa α označimo trošak pogrešnog otkrivanja, a sa β trošak pogrešnog neotkrivanja, i sa φ odnos ukupnog broja pozitivnih događaja (otkrivanja i neotkrivanja) i broja ukupnih događaja

$$\varphi = \frac{FP + TP}{FP + TP + FN + TN}$$

Funkcija troškova se može napisati kao:

$$C = FPR \cdot \alpha \cdot (1 - \varphi) + (1 - TPR) \cdot \beta \cdot \varphi$$

Tangenta ove funkcije troškova je prava sa koeficijentom

$$k = \frac{\alpha \cdot (1 - \varphi)}{\beta \cdot \varphi}$$

Koristeći ovu tangentu možemo utvrditi tačku na ROC krivoj sa najmanjim troškovima. [19]

Dobar pregled različitih načina vrednovanja, njihovog mjerenja i poređenja dat je u [50].

2.4 Otvorena pitanja sistema za otkrivanje upada

Krajem 1990-tih sistemi za otkrivanje upada ušli su u masovnu upotrebu i bili predstavljani kao proizvod bez kog se ne može i koji štiti resurse iza *firewall*-a [59]. U praktičnoj upotrebi se pokazalo da su ovi sistemi korisni, ali da ne mogu riješiti sve probleme, već predstavljaju samo dio sveobuhvatne zaštite sistema koji otkriva pokušaje upada. Kako se okruženje u kom oni rade i uslovi njihovog rada stalno mijenjaju, operator koji poznaje sistem je neizbježan učesnik. Pokušaji da se ovi sistemi iskoriste za kontrolu pristupa pokazali su se neuspješnim, jer za ovu namjenu postoje drugi bolji kontrolni mehanizmi i tehnologije, kako je navedeno u uvodnom poglavlju. Postoje i neka otvorena pitanja u sistemima za otkrivanje upada koja su vezana za detekcije upada. Neka od tih pitanja pomenuta su u prethodnom poglavlju o vrednovanju, a ostala će biti razmotrena u nastavku.

2.4.1 Opšta pitanja

Osnovno pitanje sistema za otkrivanje upada je da li se pokušaj upada uopšte može otkriti. Prilikom razmatranje računarskih virusa i antivirusnih

tehnologija, koje su srodne tehnologijama otkrivanja upada, dokazano je u [60] da je otkrivanje da li će neki programski kod učiniti nešto loše jednako teško kao i otkriti da li će se neki programski kod za neke date ulazne vrijednosti ikada završiti ili će se izvršavati u nedogled. Ovo je takozvani *problem zaustavljanja* (*halting problem*) [61], za kojeg je dokazano da nema rješenje. Prema tome otkrivanje upada je teško, ili bolje reći nerješivo u opštem slučaju. Nadalje, događaje koji se samo rijetko dešavaju gotovo je nemoguće otkriti, što je matematički dokazao Axelsson [62]. Dakle, kada je broj normalnih događaja neuporedivo veći od broja pokušaja upada, što je dugoročno posmatrano uvijek slučaj, radi takozvane *pogreške proporcije* (*base rate fallacy*) izbjegavanje lažnih uzbuna postaje veoma važno.

2.4.2 Otpornost samog sistema na napade

Sistem za otkrivanje upada može i sam biti meta napada pa je od presudne važnosti da sistem bude otporan na sve napade. Cilj napadača može biti da preuzme kontrolu nad sistemom za otkrivanje upada ili da prekine njegov rad. Mrežni sistemi koji pasivno posmatraju saobraćaj mogu takođe biti meta napada [63]. Srećom većina savremenih sistema se pokazala otpornom na ovu vrstu napada [50]. Druga vrsta napada usmjerena je na smanjivanje ili potpuno onemogućavanje otkrivanja pokušaja upada, odnosno praktično onesposobljavanje sistema. Klasifikacija ovakvih napada data je u [50]:

1. Zagušivanje sistema generisanjem velike količine događaja koja prevazilazi procesne mogućnosti sistema;

2. Zagušivanje sistema i/ili njegovog operatora, velikim brojem lažnih uzbuna putem generisanja događaja koji nisu napadi, ali su tako napravljeni da ih sistem prepoznaje kao napade;

3. Sakrivanje glavnog napada u velikom broju događaja koji jesu napadi, ali napadač koristi samo jedan događaj za upad, a ostali predstavljaju „dimnu zavjesu“.

2.4.3 Propusnost

Propusnost je mjera količine događaja koju sistem može da klasificira u određenom vremenu i ona je direktno povezana sa mogućnošću otkrivanja upada. Kako sistemi za otkrivanje upada treba da rade u vremenu bliskom realnom, propusnost bi trebala biti takva da sistem može obraditi sve događaje, odnosno da ni jedan ne propusti, a da to ne utiče negativno na brzinu rada sistema koji štiti. Sistemi koji rade na računaru troše resurse računara koji štite, ali imaju pristup svim događajima na računaru i uglavnom propusnost za njih nije veliki problem. Napadi na pojedini računar trebali bi da se mogu izvršiti na tom računaru, što znači da je obim događaja takav da se može otkriti. Izuzetak su napadi na dostupnost, ali oni su očigledni i stoga laki za otkrivanje.

Zahtjev za propusnost teže je zadovoljiti kod mrežnih sistema za otkrivanje upada, jer su oni veći nego zahtjevi za sisteme za otkrivanje upada na računaru. Ovi sistemi štite više od jednog računara i prate saobraćaj na mrežnim segmentima čiji obim može prelaziti desetine Gb/s. U slučaju preopterećenja oni počinju jednostavno da propuštaju neke pakete bez pregleda, jer ne mogu stići da obrade sve pakete koji prolaze kroz nadgledani mrežni segment. Prvi prijedlozi za rješenja ovog problema dati su u [64]. Na početku je propusnost povećavana tako da se analizira samo jedan dio mrežnog paketa i to najčešće zaglavlje. Tada su se pojavili napadi koji su se prenosili isključivo na aplikativnom nivou mrežnih paketa, pa je analiza sadržaja paketa postala krucijalna. Poboljšavanje algoritama radi povećanja propusnosti ipak ima svoje granice. Najbrža rješenja koja uspjevaju pratiti saobraćaj pri brzinama većim od 10 Gb/s su realizirana na posebnom hardveru [65] [66].

2.4.4 Akcije poslije otkrivanja upada

Poznata konsultantska i istraživačka firma iz oblasti informacija i tehnologije Gartner Inc. se 1997. izrazila veoma povoljno o sistemima za otkrivanje

upada [67]. Međutim, 2002. ista firma je proglasila da su ti isti sistemi mrtvi [68].

Pojavio se novi termin: *Sistemi za prevenciju upada*. Ovi sistemi su bili predmet velikih tehnoloških debata u sigurnosnoj zajednici. Zaključak je bio da, oni ipak ne mogu i ne treba da zamjene sisteme za otkrivanje upada iako zvuče mnogo bolje od sistema za otkrivanje upada. Naime, sistemi za prevenciju upada su ustvari uređaji za kontrolu pristupa. Dakle oni imaju drugačiju ulogu koja je komplementarna ulozi detekcije napada, jer sistemi za otkrivanje upada, pogotovo mrežni, nisu izvedeni tako da mogu prekinuti napad koji otkriju. Jedan mogući pristup je da se informacija o otkrivenom pokušaju upada proslijedi nekom uređaju za kontrolu pristupa koji može prekinuti napad. Međutim, sa ovim je neophodno biti izuzetno oprezan. Prekidanje konekcije na mreži nepovoljno utiče na dostupnost i lažna uzbuna može izazvati veliku štetu. Mogući su i namjerni napadi na ovakve automatizovane sisteme, čiji je cilj *uskraćivanje usluge* (*denial of service*). Napad se sastoji u iniciranju događaja koji će pogrešno biti proglašeni upadima i obustavljanju normalnog rada sistema koji se štiti.

Očigledna mjera poslije otkrivanja upada je njegovo zaustavljanje koje provodi operator ili neki automatski sistem. No to je samo jedan aspekt obrade incidenata koji ugrožavaju sigurnost. Prema [69] obrada sigurnosnih incidenata ima šest faza:

1. Priprema – Ova faza prethodi napadima u njoj se uspostavljaju procedure i mehanizmi za otkrivanje i odgovor na napade.

2. Identifikacija napada – Ova faza pokreće preostale.

3. Kontrola napada – U ovoj fazi se pokušava koliko je moguće umanjiti štetne posljedice napada.

4. Zaustavljanje napada – Ova faze se bavi zaustavljanjem napada i blokiranjem budućih sličnih napada.

5. Oporavak od napada – Povrat sistema u sigurno stanje.

6. Akcije nakon napada – Ova faza uključuje poduzimanje odgovarajućih akcija protiv napadača, identifikaciju problema u obradi incidenta i učenje na incidentu.

2.4.5 Posebna pitanja mrežnih sistema

Kako je ranije objašnjeno, ovi sistemi pasivno posmatraju saobraćaj na nekom mrežnom segmentu. Analizom saobraćaja oni pokušavaju da otkriju kako će paketi koje vide biti protumačeni na računarima koje štite, te kako će računari na njih reagovati. Nažalost, u mrežnom saobraćaju nema dovoljno informacija da bi se moglo sa sigurnošću utvrditi šta se dešava na računarima u mreži. Ptacek i Newsham [70] su još 1998 ukazali na ovaj problem i pokazali da se IP fragmentacijom mogu zavarati svi tada postojeći mrežni sistemi za otkrivanje upada. Takođe su pokazali da napadači pogodnim oblikovanjem i umetanjem paketa mogu navesti mrežni sistem za otkrivanje upada da sasvim krivo zaključi šta od saobraćaja dolazi u računar i na koji će način to biti u računaru protumačeno. Iako najnoviji mrežni sistemi uvode metode za otkrivanje i ovakvih pokušaja, ovo i dalje ostaje kao problem.

Treba napomenuti i da je stvarni mrežni saobraćaj prepun neobičnih paketa [71], koji nisu zloćudni, ali odstupaju od standarda. Problem je što ovakvi paketi gotovo uvijek generišu lažnu uzbunu, a to smanjuje efikasnost mrežnih sistema za otkrivanje upada.

Istorija sistema za otkrivanje upada ponekad poredi sa trkom u naoružanju [72]. Postoji brojni primjeri ove trke u oblasti mrežnih sistema. Najnoviji primjer je rad [73], gdje je pokazan način izvođenja napada koga ne mogu da otkriju metode predložene u [74], [75], [76] i [77].

2.5 Problemi novih mrežnih sistema za otkrivanje upada

Problemi aktuelnih mrežnih sistema za otkrivanje upada relevantni za disertaciju kao i stanje istraživanja u ovoj oblasti biće dati u nastavku.

2.5.1 Analiza mrežnih paketa

Tačnost otkrivanja pokušaja upada direktno je proporcionalna detaljnosti pregleda mrežnih paketa. Pregled mrežnih paketa sličan je pregledu putnika na aerodromu. Službenici koji kontrolišu putnike na ulasku u aerodrom, ili rade neku sličnu kontrolu, mogli bi detaljnim pregledom tačno ustanoviti šta svaki putnik unosi i tako otkriti skoro sve zlonamjerne putnike. No, detaljan pregled svakog putnika trajao bi predugo. Da se ne bi previše usporio protok putnika, trebao bi angažovati više službenika ili uspostaviti više kontrolnih linija, što bi moglo biti preskupo. Zato se u praksi koristi površnija, ali uglavnom zadovoljavajuća kontrola. Svi putnici prolaze kroz detektor metala, a njihov prtljag službenici pregledaju rendgenom u potrazi za sumnjivim predmetima. Pored toga, službenici su obučeni da prepoznaju neuobičajeno ponašanje putnika. Samo sumnjivi putnici se detaljnije pregledaju. No paralela nije sasvim tačna. Brzina analize mrežnih paketa treba da bude veća od brzine protoka paketa, jer se paketi ne smiju zadržavati na jednom mjestu radi pregleda.

Sadržaj mrežnog paketa može imati nekoliko omotnica, zaglavlja i ponegdje nastavaka. Danas dominiraju TCP/IP paketi kod kojih je sadržaj upakovan u transportnu (TCP ili UDP), mrežnu (IP) i omotnicu sloja veze podataka. Da bi se pregledao sadržaj paketa potrebno je prvo analizirati sve omotnice, od omotnice sloja podataka do transportne omotnice. Vrijeme pregleda se može donekle skratiti ako se koriste brzi računari ili posebno pravljeni uređaji. Najveći problem je analiza sadržaja, jer sadržaj može biti namijenjen različitim aplikacijama ili proizveden od strane različitih aplikacija. Aplikacije kojima je paket namijenjen su jedine koje ga mogu u potpunosti razumjeti. Postoji

teoretska mogućnost da se identifikuje smisao paketa tako da se na mrežnom sistemu za otkrivanje upada izvršavaju sve potencijalne aplikacija, no to zasada nije izvodljivo.

2.5.2 Metode analize

Nakon što su opisane neke poteškoće u analizi mrežnih paketa, biće navedene aktuelne ideje za realizaciju dovoljno brzih i dovoljno tačnih analiza.

Prvi mrežni sistemi za otkrivanje upada koristili su minimalni skup podataka iz paketa, kao što je zaglavlje i dužina. Poboljšanje metoda se kretalo u pravcu dublje analize paketa i analize više omotnica. Naprimjer, metode zasnovane na potpisima definisale su sumnjive kombinacije podataka u zaglavljima, kao što su neuobičajene ili nedozvoljene TCP i IP opcije. Metode otkrivanja anomalija formirale su model normalnog ponašanja takođe na osnovu skupa IP adresa, TCP/UDP portova i drugih podataka iz zaglavlja paketa.

Većina savremenih računarskih napada usmjerena je na aplikacije [78] iz dva osnovna razloga. Prvi, danas se većina sigurnosnih propusta nalazi u korisničkim aplikacijama [79]. Drugi, većinu upada na niže nivoe protokola se sada može efikasno spriječiti korištenjem standardne opreme za zaštitu računarskih mreža kao što je *firewall*. Zbog toga je postala neophodna analiza sadržaja paketa.

Sadržaj paketa je samo niz bajta. Ovaj niz uglavnom ima značenje tek kada se poveže sa sadržajem drugih paketa iz sesije. Ovo značenje jasno je samo izvorišnoj i odredišnoj aplikaciji. Za razliku od zaglavlja sadržaj nema fiksni format, manji skup ključnih riječi ili ograničen niz vrijednosti. U ovom nizu bajta bilo koji znak, odnosno vrijednost bajta, se može pojaviti na bilo kom mjestu u sadržaju. Ovo razmatranje je bitno radi metoda pristupa analizi ovog niza bajta.

Savremeni mrežni sistemi za otkrivanje upada na osnovu potpisa, kao što je Snort [80] ili Bro [63] porede sadržaj paketa sa poznatim nizovima ili kombinacijama bajta karakterističnim za poznate napade, nazvanim potpisi. Potpisi se brzo i često ažuriraju. Ipak, moguće je, a to se i dešava da se pojave novi napadi za koje još ne postoji potpis i koje ovi sistemi ne mogu prepoznati.

Pregled i analiza savremenih sistema za otkrivanje upada na osnovu anomalija u sadržajima mrežnih paketa tema je rada [81].

Savremena istraživanja okrenuta su uglavnom ka sistemima za otkrivanje upada koji otkrivaju anomalije putem analize sadržaja i konekcija. Naprimjer, u radu [74] se koristi specifično znanje o manjem broji najčešće korištenih mrežnih usluge u sistemu koji se štiti. Analizira se vrsta zahtjeva, dužina zahtjeva i frekventna raspodjela karaktera u zahtjevu. Ovo je jedan od prvih radova koji uključuje analizu sadržaja paketa. Međutim analiza dužine zahtjeva ima svoje nedostatke. Napadači uglavnom mogu u svoje pakete dodati nepotrebne bajte radi produžavanja ili podijeliti napad u manje pakete radi smanjivanja dužine. Slično je i sa frekventnom raspodjelom karaktera u sadržaju, ali ova metoda će se pojavljivati i u drugim radovima i ima određene veze sa pristupom u ovoj tezi. U radu [76] preporučuje se da se analizira samo prvih 48 bajta paketa, počinjući sa IP zaglavljem, i to za devet najčešćih protokola, pri čemu se gleda na frekvenciju pojave pojedinih vrijednosti bajta. Podjelom paketa na one koji pripadaju različitim protokolima moguće je bolje prilagoditi metodu otkrivanja svakom od protokola. Ipak analizirani dio sadržaja, koji je prilično kratak kada se odbije dio od 48 bajta koji pripada zaglavljima, nije dovoljan za pouzdano otkrivanje napada koji se prenose u sadržaju paketa. U [82] se predlaže kombinacija otkrivanja na osnovu potpisa i otkrivanja na osnovu anomalija, uz minimalno korištenje sadržaja paketa. Za otkrivanje se koriste prošireni konačni automati. Ideja povezivanja otkrivanja na osnovu potpisa i na osnovu anomalija je odlična i logična i to je jedan

pravac u kom se razvijaju sistemi za otkrivanje upada. Analiza korištenjem konačnih automata je jedan od pristupa koji se pokazao primjenljivim za sisteme za otkrivanje upada. U smislu analize sadržaja ovaj rad ne donosi puno novog. U [47] se uvodi dvoslojna arhitektura. U prvom sloju je vještačka neuronska mreža koja realizuje samoorganizujuću mapu. Mapa komprimuje sadržaj paketa na jedan bajt informacija. Drugi sloj uzima informacije iz zaglavlja i bajt dobiven iz prvog sloja i otkriva abnormalne slučajeve. Ovo je još jedan primjer kombinovanja pristupa. U ovom slučaju kombinuje se analiza zaglavlja koja je brza, sa analizom sadržaja koja je u principu komplikovanija. U tom radu se pokušava pojednostaviti analiza sadržaja sa njegovim pretvaranjem u jedan bajt koji bi trebao zadržati dovoljno informacija. Ovim pojednostavljenjem se mora izgubiti nešto informacija koje mogu biti bitne za otkrivanje upada. Korištenje vještačkih neuronskih mreža je takođe primjer jednog od pristupa koji se pokazao korisnim za sisteme koji otkrivaju napade otkrivanjem anomalija. Neophodno je reći da ovi radovi imaju dobre rezultate sa podacima koji su korišteni za testiranje. Njihov nedostatak je što nedovoljno analiziraju kompletan sadržaj paketa. Savremeni alati za manipulaciju paketa mogu napraviti isti napad u različitim oblicima paketa. Na ovaj način mogu organizovati paket da i dalje ima svoj napadački učinak, ali da bude organizovan tako da ga gornji sistemi ne mogu prepoznati kao pokušaj upada.

Posebnu grupu čine radovi posvećeni Web napadima koji analiziraju HTTP zahtjeve. U radu [83] model normalnog rada se zasniva na parametrima HTTP GET zahtjeva i njihovim vrijednostima, koji se nazivaju atributi. U obzir se uzima dužina atributa, raspodjela znakova, struktura, vrijednosti, te pojava nekog atributa. Ovaj pristup je razvijen i poboljšan u radovima [84] i [85]. Nešto drugačiji pristup analizi parametara HTTP zahtjeva napravljen je u radu [86], gdje se koristi indukcioni algoritam determinističkih konačnih automata za otkrivanje malicioznih zahtjeva. Sadržaj većine paketa koji se upućuju Web serveru je neki oblik GET zahtjeva. Analizom parametara ovih

zahtjeva moguće je prepoznati dovoljno drukčije koji mogu biti potencijalno opasni. Ovaj pristup koristi poznavanje HTTP protokola i pokazao je dobre rezultate. Primjedba je da u principu Web server ne mora dobiti GET zahtjev da bi bio napadnut, mada je to uobičajeni način. Ovaj pristup bi trebao otkriti zahtjeve koji su maliciozni, ali, kako je ranije rečeno, sadržaj paketa ne mora biti formatiran u očekivanom obliku zahtjeva. Moguće je zamisliti napadačke pakete u nekom formatu sadržaja koji neće biti otkriveni. Pristup korišten u ovoj tezi je nešto drugačiji i zapravo može biti komplementaran ovim pristupima.

Poseban pravac istraživanja je traženje izvršnog koda napada u sadržaju paketa. Ovi pristupi analiziraju sadržaj paketa ali ne radi pravljenja modela normalnog ponašanja. Cilj analize je da se na neki način otkrije niz bajta u sadržaju paketa koji predstavlja napadački kod. Otkrivanje pomoću statičke analize je predloženo u [72]. Izvršni kod napada prepoznaje se po karakterističnim nizovima bajta i po kontrolnim komandama. Otkrivanje karakterističnog niza komandi koji omogućava pokretanje napadačkog koda (*sled*) nakon preljeva međuspremnika tema je rada [87]. Pomoću strukturalne analize binarnog koda sadržaja i potpisa nalaze se mutacije poznatog crva (*worm*) [88], i tako otkrivaju modifikovane verzija postojećih napada. Tehnika apstrakcije koda predložena je za otkrivanje koda u sadržaju paketa u [89]. Emulacija na mrežnom nivou predlaže se kao metod za detekciju polimorfnog koda u [90]. Svi ovi radovi prave pretpostavke o tome kako napadački kod može izgledati i na osnovu njih rade analizu sadržaja paketa. S obzirom na brzinu pojavljivanja novih napada i mutacije poznatih napada upitno je koliko ovi pristupi mogu držati korak. Autor rada [91] smatra da se ne isplati modeliranje svih verzija polimorfnih nizova izvršnih bajta, da pravljenja potpisa ima ograničenja i da zato otkrivanje napada putem anomalija ima bolju budućnost.

Metoda opisana u disertaciji oslanja se na radove opisane u nastavku. U svim ovim radovima koristi se podjela ili grupisanje sadržaja paketa radi pravljenja modela. Načini podjele i grupisanja su različiti. Pristup predložen u [77] i poboljšani algoritam u [92] koriste frekvencije pojedinih bajta u sadržaju paketa kao osnovu za model. Ovo je najjednostavniji način podjele. Sadržaj se dijeli na pojedinačne bajte. Ovi radovi prave frekventnu analizu bajta u sadržaju paketa. Na osnovu ove analize prave profil normalnog ponašanja. Za pakete koji se analiziraju da li su napadački ili ne računa se pojednostavljena Mahalanobijeva udaljenost [93]. Na osnovu iznosa ove udaljenosti utvrđuje se da li je paket maliciozan ili ne. Rezultati koje su autori objavili su odlični, ali je jedno-bajtna analiza prejednostavna da ne bi mogla biti zaobiđena. U radu [73] je to i pokazano. Objašnjen je princip i napravljeni napadi koje jedno-bajtna analiza ne može otkriti.

Iz ovog razloga tekuća istraživanja počinju koriste nizove od više bajta. U radu [94] model se pravi na osnovu nizova uzastopnih bajta jedne fiksne dužine, takozvanih *n-grama*. N-grami će biti dodatno obrađeni kada se bude govorilo o metodi korištenoj u tezi. U tom radu se ne broji broj pojavljivanja svakog od n-grama već se samo evidentira da se pojavio. Iznos odstupanja paketa je jednostavan odnos broja n-grama u paketu koji se nisu pojavili tokom faze učenja i ukupnog broja n-grama u paketu. Napravljeni su različiti testovi sa dužinama od 3 do 8 bajta. Iako su rezultati koje su autori objavili odlični, ovaj pristup je vrlo osjetljiv na prisustvo napada u saobraćaju za učenje normalnog ponašanja. Samo jedan napad u ovom saobraćaju učiniće da svi njegovi n-grami postanu dio modela i budu apsolutno neprepoznatljivi kao napadi.

Prva logička podjela paketa na dijelove predložena je u [95]. Tačnije, ovaj rad se bavio analizom sadržaja radi pronalaska skupa separatora koji testirani skup paketa pogodno grupiše. Utvrđeni skup separatora zapravo predstavlja model. Paketi za koje se putem ove analize dobiju drugačiji separatori, smatraju se

neobičnim. Bitan rezultat ovog rada, korišten i u ovoj tezi, je prvi prijedlog skupa separatora za sadržaj HTTP paketa

Ove separatore koristi [96]. U tom radu pravi se model zasnovan na nizovima uzastopnih bajta u sadržaju paketa. Upoređeni su nizovi fiksne dužine - n-grami i nizovi promjenljive dužine razdvojeni separatorima - riječi. Bitan rezultat rada je potvrda da se model sa riječima može koristiti za uspješno otkrivanje upada. Pokazalo sa da ovaj model ne zaostaje po sposobnosti razlikovanja normalnog saobraćaja od upada za modelom koji se dobije istovremenim kombinovanjem više različitih n-grama. Ovaj drugi model je računarski mnogo zahtjevniji.

Potrebno je napomenuti da se pristup sa korištenjem se n-grama i riječi na različite načine počeo koristiti mnogo ranije i uspješnije kod sistema za otkrivanje upada na računaru [97] [98] [99] [100] [101]. Ipak ovi sistemi su dovoljno različiti od mrežnih da se ideje i pristupi ne mogu direktno primjeniti.

Metoda koja će prvo biti predstavljena u nastavku rada zasnovana je na pristupima iznijetim u prethodnim radovima. Ideja je da se podjela na riječi iz [96], kombinuje sa poboljšanjem jednostavnog računanja iz [94]. Na ovaj način trebao bi se dobiti model koji jednostavan za pohranjivanje i brz metod detekcije. Dodatno će se povezivati po dvije uzastopne riječi radi povećanja preciznosti modela i smanjivanja mogućnosti izbjegavanja otkrivanja. Izbjegavanje otkrivanja putem uklapanja napada u model normalnog saobraćaja je poseban problem sistema zasnovanih na otkrivanju anomalija. Jedini rad iz oblasti mrežnih sistema za otkrivanje upada koji se djelimično bavi ovim pitanjem je [94]. Ovom problemu će biti posvećeno više pažnje u radu. Početna metode biće unaprijeđena novim pristupom posuđenim iz kriptografije koji treba da onemogući ovakve napade.

3 ANALIZA SADRŽAJA MREŽNIH PAKETA

Kako je u opisu aktuelne problematike mrežnih sistema za otkrivanje upada navedeno, savremeni sistemi moraju da vrše analizu sadržaja mrežnih paketa. U prvom dijelu ovog poglavlja biće opisana predložena metoda analize. Rezultati testiranja ove metode sa stvarnim podacima biće dati u drugom dijelu.

3.1 Okruženje za primjenu predložene metode

Prije predstavljanja metode potreban je kratak prikaz okruženja u kome se ona primjenjuje, kao i definicija korištenih pojmova. Prvo će biti objašnjena struktura analiziranih paketa. Nakon toga biće obrazložen izbor protokola za testiranje. Na kraju uvoda biće naveden način dobivanja mrežnog saobraćaja koji će biti korišten za testiranje metode.

3.1.1 Struktura i veličina TCP/IP paketa

Svaki mrežni protokol ima sopstvenu strukturu paketa. Ovaj rad se bavi TCP/IP protokolima i paketima, koji dominiraju u savremenim mrežama. Najveća svjetska mreža Internet koristi TCP/IP. Ovdje će biti dati samo oni aspekti TCP/IP koji su neophodni za definisanje problema u ovoj disertaciji. Ostali detalji su dati u mnogobrojnoj literaturi kao što je [102] i [103].

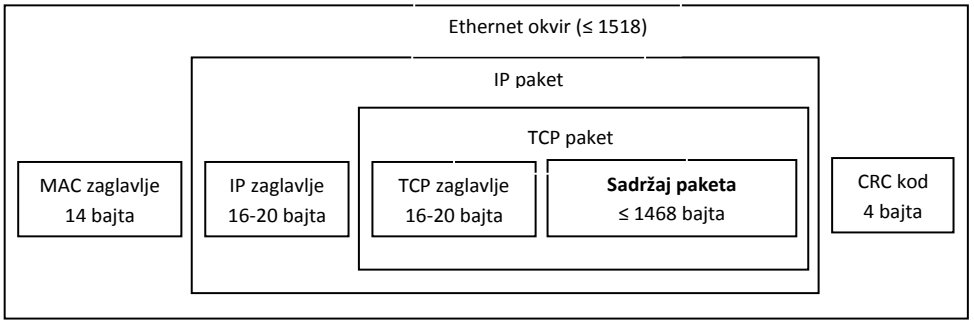
Kako je ranije rečeno, TCP/IP paket se sastoji od sadržaja, koji se prenosi od aplikacije na jednom računaru do aplikacije na drugom računaru i više zaglavlja, koja omogućavaju prenos tog sadržaja. Aplikacije koje među sobom komuniciraju mogu, ali i ne moraju, biti korisničke. Korisničkim aplikacijama se smatraju one koje ostvaruju interakciju sa korisnikom računara na kom se izvršavaju. Bitno je da te aplikacije koriste isti „jezik“, odnosno protokol, koji definiše sintaksu i semantiku razmjenjivanih poruka. Ove poruke čine sadržaj mrežnih paketa. Radi efikasnosti prenosa protokol sloja veze podataka koji definiše format paketa koji putuju po mediju postavlja ograničenje na

maksimalnu veličinu paketa. Veličina zavisi od prenosnog medija. Naziv mrežnog paketa na ovom nivou je okvir, jer je na početku zaglavlje a na kraju zaleđe (t*railer*) čija je svrha kontrola grešaka.

Maksimalna veličina sadržaja paketa jednaka je maksimalnoj veličini okvira umanjenoj za veličine zaglavlja i zaleđa svih nivoa. Originalne poruke se zato nekad moraju podijeliti u više dijelova. Ovi dijelovi se na prijemnoj strani prvo sastave u originalnu poruku, pa se onda predaju aplikaciji na prijemnom računaru. Za aplikacije je ovo rastavljanje i ponovno sastavljanje sadržaja neprimjetno.

Za analizu sadržaja mrežnih paketa bitna je činjenica da li je sadržaj nekog paketa poruka ili samo njen dio. Iscjepkanost poruka otežava, ako ne i onemogućava, semantičku analizu sadržaja mrežnih paketa, jer samo kompletna poruka ima neko značenje za aplikacije koje među sobom komuniciraju. Sa druge strane ograničenje na veličinu sadržaja paketa olakšava analizu zbog manjeg broja bita.

Ovaj rad se, bez umanjenja opštosti, bavi sa slijedećim skupom protokola. To su TCP transportni protokol, IPv4 mrežni protokol i Ethernet protokol sloja linka podataka. Veličine zaglavlja, zaleđa i sadržaja paketa su date u nastavku: Ethernet, odnosno IEEE 802.3, ima maksimalnu veličina okvira 1518 bajta, veličinu zaglavlja 14, a zaleđa 4 bajta [104]; veličina IPv4 zaglavlja je 16 do 20 bajta [105], (mada se u praksi uglavnom koristi 20); veličina TCP zaglavlja je takođe 16 do 20 bajta [106], ali se i ovdje obično koristi 20 bajta. Jednostavna računica pokazuje da je maksimalna veličina sadržaja paketa za ovu vrstu okvira 1468 bajta. Slika 3. prikazuje izgled navedenog okvira.



Slika 3. Ethernet – IPv4 – TCP okvir

Ako se koristi neki drugi transportni protokol ili neki drugi protokol sloja veze podataka, veličina sadržaja paketa može biti nešto manja ili veća. U svim praktičnim slučajevima veličina sadržaja paketa je oko 1500 bajta. Ova veličina koristiće se u daljem razmatranju analize sadržaja mrežnih paketa.

3.1.2 HTTP protokol

U upotrebi ima mnogo protokola, ali se neki koriste mnogo više od ostalih. Ti protokoli su najviše izloženi napadima, te im je zaštita najpotrebnija.

Predložena metode otkrivanja upada može se primjeniti na sve protokole. Za testiranje će se ipak koristiti samo HTTP na kom je zasnovan *World Wide Web* (Web). Važeća verzija protokola HTTP 1.1 je definisana u RFC 2616 [107]. Više informacija o samom protokolu se može pronaći u literaturi [108] [109]. U nastavku ce biti obrazložen ovaj izbor.

Web je danas postao gotovo sinonim za Internet. Termin „Povezivanje na Internet“ najčešće znači pristup Web-u. Gotovo da više ne postoji organizacija koja nema svoju Web lokaciju. Elektronsko poslovanje najčešće koristi Web bazirane aplikacije. E-pošta, najveći konkurent Web-u po popularnosti se sada gotovo redovno nudi i preko HTTP-a kao Webmail.

Standardni TCP port za HTTP je 80. Zbog ovolike upotrebe HTTP protokola za komunikaciju Web klijenata i Web servera ovaj port je uglavnom

otvoren za saobraćaj na većini *firewall*-a. Veliki broj Web aplikacija i njihov brzi razvoj neminovno dovodi do sigurnosnih propusta. Po statistikama SANS instituta sigurnosni propusti u Web aplikacijama čine gotovo jednu polovinu svih sigurnosnih propusta otkrivenih u 2007 godini [79]. Napadači obilato koriste otvoreni pristup i brojne sigurnosne propuste. Prema tekućim izvještajima o prijetnjama sigurnosti [110] i sigurnosnim trendovima [111], Web bazirani napadi čine većinu napada, a imaju i tendenciju porasta.

Otkrivanje pokušaja upada putem HTTP protokola predstavlja trenutno potencijalno najkorisniju mjeru zaštite informacionih sistema. Zbog toga je HTTP protokol izabran kao testni.

3.1.3 Mašinsko učenje

Kod sistema zasnovanih na otkrivanju anomalija tokom učenja se pravi model normalnog ponašanja sistema. Za učenje se koriste podaci za trening. Za pravljenje dobrog modela potrebni su adekvatni podaci. Adekvatnost ovdje znači da se koristi indikativna vrsta podataka, recimo bajti sadržaja paketa i, što je još važnije, da su to podaci uzeti u toku normalnog ponašanja sistema. Dobar model se može dobiti samo ako podaci za trening tačno odslikavaju ponašanje štićenog sistema. Ako podaci za trening ne uključuju sva normalna ponašanja štićenog sistema, sistem za otkrivanje će praviti lažne uzbune. S druge strane, ako se u ovim podacima nalaze podaci nastali kao posljedica napada, oni će ući u model normalnog ponašanja i neće moći biti otkriveni kao anomalije.

Mašinsko učenje, koja pokriva i pomenuta pitanja, temeljito je obrađeno u brojnoj literaturi poput [112] i [113]. Mašinsko učenje relevantno za sisteme za otkrivanje upada na osnovu anomalija, sa opisima poteškoća pri učenju i mogućnosti njihovog otklanjanja, obrađeno je u [45]. U ovom radu će se relevantni problemi učenja uvoditi i objašnjavati u toku predstavljanja metoda.

Otkrivanje upada biće posmatrano kao prepoznavanje paketa koji po svom sadržaju dovoljno odstupaju od utvrđenog normalnog sadržaja paketa.

3.1.4 Testni podaci

Način testiranja sistema za otkrivanje upada još nije precizno definisan. Pored toga, sistemi koji otkrivaju HTTP upade imaju dodatne specifičnosti [114].

Za testiranje nekog metoda otkrivanja upada od presudnog značaja su testni podaci. Većina istraživača je ranije imala svoje skupove testnih podataka, koji nisu bili dostupni drugim istraživačima. To je otežavalo provjere metoda i njihovo poređenja. Zato je Lincoln laboratorija na MIT (*Massachusetts Institute of Technology*) pod sponzorstvom DARPA (*Defense Advanced Research Projects Agency*) 1998 [115] i 1999 [116] kreirala velike skupove podataka za testiranje i učinila ih javno dostupnim. To su ustvari snimci vještački generisanog mrežnog saobraćaja koji treba da liči na pravi saobraćaj. Snimci saobraćaja iz 1998. pokrivaju: sedam sedmica sa normalnim saobraćajem u kom se pojavljuju poznati i evidentirani napadi, te dvije sedmice saobraćaja sa napadima koji nisu označeni. Snimci iz 1999. sadrže dvije sedmice saobraćaja bez napada, jednu sedmicu sa označenim napadima, te dvije sedmice sa neoznačenim napadima. Snimci sa označenim napadima služe za trening sistema, a oni sa neoznačenim za testiranje.

Koristeći te skupove Lincoln laboratorija je napravila veliko testiranje tadašnjih sistema za otkrivanje upada. Ovi skupovi podataka su još uvijek najpoznatiji i dugo vremena su bili veoma korišteni za testiranje.

Principijelni nedostaci DARPA skupova su primjećeni nedugo po njihovom objavljivanju [117] [118]. DARPA skupovi su nepogodni i za testiranje metoda predloženih u ovom radu. Dva su glavna razloga za to. Prvo, saobraćaj u ovim skupovima se prilično razlikuje od savremenog saobraćaja u računarskim mrežama i ne može se više smatrati adekvatnim za učenje normalnog saobraćaja. Simulirani napadi su zastarjeli i nisu karakteristični za

savremene računarske mreže. Drugo, u DARPA skupovima postoje samo četiri Web napada, što ni iz daleka nije dovoljno za temeljito testiranje i provjeru sistema za otkrivanje Web upada. Slično mišljenje dijele i drugi savremeni autori [114]. Većina radova napisanih u zadnjih pet godina relevantnih za ovu tezu [83], [84], [85], [86], [92] i [94] ne koristi DARPA podatke za testiranje već koristi uglavnom saobraćaj dostupan autorima radova koji nije dostupan za javnu analizu. Oni koji koriste DARPA testni skup [77] i [96] uz njega koriste i svoj skup podataka koji nije javno dostupan.

3.1.4.1 Normalan saobraćaj

Iz ovih razloga za testiranje je korišten stvarni saobraćaj, koji je prikupljen sa računarske mreže Elektrotehničkog fakulteta u Sarajevu. Snimanje je obavljeno tokom 12 dana novembra 2007. godine. Pohranjen je sav saobraćaj sa unutrašnje strane rutera koji povezuje Fakultet sa institucijom koja pruža uslugu povezivanja na Internet. Na ovaj način zabilježen je sav saobraćaj iz vanjskog svijeta prema svim serverima Fakulteta i obratno, kao i sav saobraćaj koji je kroz unutrašnji *firewall* iz lokalne mreže Fakulteta išao ka Internetu i obratno. Važno je istaći da je saobraćaj koji sa Interneta dolazi do mreže Elektrotehničkog fakulteta prilikom prolaska kroz vanjski ruter bio filtriran na osnovu pravila podešenih na ruteru. Ovim filtriranjem se sprečava prolazak saobraćaj koji mreža Fakulteta ne očekuje. Ovakvo filtriranje je uobičajeno i pomenuto je kao jedan od koraka sveobuhvatne zaštite računarske mreže. Snimljeni saobraćaj zato predstavlja saobraćaj koji stvarno dolazi do servera Fakulteta, koji je objekat zaštite.

Za trening sistema potreban je čist saobraćaj, bez napada, a stvarni saobraćaj često to nije. Zato je prikupljeni saobraćaj pročišćen koristeći i automatski i ručni pregled. Automatski pregled saobraćaja obavljen je pomoću Snort mrežnog sistema za otkrivanje upada zasnovanog na potpisima. Za ispravno funkcionisanje ovakvi sistemi trebaju biti fino podešeni i imati ažurne potpise napada. Podešavanje je proces izbora opcija zaštite koji odgovaraju servisima

koji štićena mreža nudi. Opcije zaštite servisa koji ne postoje u mreži se ne biraju jer nisu potrebne. Korišteni Snort je bio u potpunosti podešen za otkrivanje upada relevantnih za mrežu Elektrotehničkog fakulteta u Sarajevu. Ažurni potpisi svih poznatih napada dobiveni su sa Snort stranice za ažuriranje. Ručni pregled obavljen je uz saradnju sa mrežnim administratorom koji dobro poznaje računarsku mrežu na Fakulteta, servise koje nudi i uobičajeni mrežni saobraćaj. Na ovaj način dobiven je saobraćaj koji je očišćen od napada koliko je to moguće. Taj saobraćaj korišten je za trening sistema.

Pitanje realne mogućnosti dobivanja stvarnog saobraćaja u kome garantovano nema napada je još otvoreno [119]. Saobraćaj koji je očišćen od napada, kao ovaj korišten u tezi, ipak može sadržavati napade koji ne postoje u Snort bazi i koje ni administrator sistema ne može prepoznati. Saobraćaj u kom sigurno nema napada može biti samo vještački kreiran u kontrolisanom okruženju gdje je izvor svakog od paketa poznat i potpuno pouzdan da šalje samo čiste, nenapadačke pakete. Ovakav saobraćaj nije realan i iz njega se ne može napraviti realan model normalnih paketa. Iz ovog razloga bi praktično upotrebljive metode trebale da mogu trenirati normalno ponašanje na realnom pročišćenom saobraćaju, u kom makar teoretski može biti određen broj malicioznih paketa.

3.1.4.2 Napadi

Iako u stvarnom saobraćaju može biti napada, sigurno ih nema dovoljno za testiranje nekog metoda. Zbog toga je vještački generisan mrežni saobraćaj u kome ima napada. Za generisanje ovakvog saobraćaja korištena su tri alata.

Prvi alat je Nessus [120] koji se koristi za pronalaženje sigurnosnih propusta. Dio ovog alata je baza podataka o sigurnosnim propustima. Nessus omogućava otkrivanje mrežno dostupnih aplikacija na nekom računaru ili mreži i testira da li među pronađenim aplikacijama ima onih koje imaju neki

od sigurnosnih propusta koji se nalaze u Nessus bazi. Ono što je za ovaj rad bitno je da Nessus svoje testiranje obavlja na daljinu, putem slanja mrežnih paketa. Njihov sadržaj zavisi od aplikacija na sistemu koji se testira i sigurnosnih propusta čije se postojanje provjerava. Sadržaj Nessus paketa nije isti kao sadržaj normalnih paketa i sistem za otkrivanje upada bi trebao da razlikuje Nessus pakete od normalnih. Saobraćaj koji je generisao Nessus je snimljen i korišten za testiranje.

Drugi alat je Nikto [121], koji se isto koristi za otkrivanje sigurnosnih propusta na Web serverima. Slično kao i Nessus, Nikto ima bazu podataka sigurnosnih propusta. Nikto generiše HTTP pakete koje upućuje prema testiranom Web serveru. Na osnovu reakcije Web servera Nikto zaključuje da li postoji neki sigurnosni propust. Sadržaj Nikto paketa bi trebao biti dovoljno različit od sadržaja paketa koji normalno dolaze do Web servera, tako da bi ih sistem za otkrivanje upada morao otkriti. Saobraćaj koji je generisao Nikto je snimljen je i upotrebljen za testiranje.

Treći alat je Metasploit [122]. Metasploit je razvojno okruženje koje može da kreira različite napade koji odgovaraju različitim sigurnosnim propustima, i imaju različite efekte. Napadi se nalaze u mrežnim paketima, a zasnovani su na iskorištavanju sigurnosnih propusta. Otuda i dolazi engleski naziv koji se uglavnom ne prevodi: *exploit* (od engleskog iskoristiti). Ciljevi Metasploit napada pripadaju nekoj od poznatih kategorija već definisanih u sistematizaciju upada. Metasploit mrežni paketi se generišu kombinovanjem dvije komponenete. Jedna komponenta je kod napisan na osnovu sigurnosnog propusta koji se koristi, a druga je kod napisan na osnovu željenog cilja napada. U nedostatku odgovarajućeg prevoda biće navedeni engleski nazivi ovih komponenti, koji su uobičajeni i kod nas, a to su *exploit* i *payload*. Metasploit ima u bazi veliki broj *exploit* i *payload* paketa i lako ih kombinuje.

Potrebno je ponovo naglasiti da je Metasploit alat za testiranje sigurnosti. On omogućava pravljenje napada i testiranje otpornosti na njih, stručnjacima u oblasti zaštite informacionih sistema. Matesploit napadi su stvarni napadi kakvim su sistemi izloženi u realnosti i njih bi sistem za otkrivanje upada morao otkriti. Metsploit je korišten za pravljenje napada koji su bili usmjereni ka testnom serveru. Saobraćaj tokom ovih napada je sniman i korišten za testiranje metoda otkrivanja upada.

3.2 Analiza sadržaja razdvajanjem na riječi

Sistemi za otkrivanje upada zasnovani na otkrivanju anomalija se razlikuju po skupovima podataka koje koriste za kreiranje modela normalnog ponašanja. U nedavno objavljenom radu [114] koji se bavio poređenjem sistema za otkrivanje HTTP napada na osnovu anomalija, zaključeno je da sposobnost metode da predstavi više značenja HTTP zahtjeva poboljšava sposobnost razlikovanja između normalnog i nenormalnog ponašanja. Potpuno značenje može se dobiti samo na osnovu svih paketa iz HTTP zahtjeva, kojih može biti više od jednog. Značenje jednog paketa koji se analizira radi detekcije najbolje se može utvrditi analizom njegovog cjelokupnog sadržaja. Sa druge strane podjelom paketa na manje dijelove dobija se manji model i jednostavnija detekcija. Pokazalo se da podjela na pojedinačne bajte nije dobra jer je ovakve metode detekcije relativno lako zaobići [73]. Neophodno je koristiti nizove uzastopnih bajta kao osnovu za model. Nizovi bajta mogu biti fiksne dužine (n-grami) ili promjenljive dužine razdvojeni separatorima (riječi). U radu [96] pokazano je da korištenje riječi daje marginalno lošije rezultate od korištenja kombinovanja više različitih n-grama uz n puta manje računarsko opterećenje. Iz ovog razloga riječi su korištene za kreiranje modela normalnog ponašanja.

3.2.1 Razdvajanje na riječi

Ova metoda modeliranja normalnog ponašanja zasniva se na razdvajanju sadržaja mrežnih paketa na riječi. Riječ ima isti smisao kao u normalnom

jeziku, to je niz uzastopnih simbola između dva znaka razdvajanja - separatora. U govornom tekstu separatori su prazna mjesta i znakovi interpunkcije, kao što su tačka, zarez i slično. U sadržaju mrežnih paketa uzastopni znakovi koji čine riječi su bajti, a separatori su posebne vrijednosti bajta koje je potrebno utvrditi. Izbor separatora nije jednoznačan i očigledan, kao što je slučaj u govornom tekstu, a zavisi od korištenog aplikativnog protokola.

Metoda izbora separatora je tema [95] gdje su separatori osnova modela normalnog ponašanja. Za svaki paket metoda određuju separatore na osnovu postavljenih kriterija. Neobični paketi su oni koji po utvrđenim kriterijima imaju neuobičajene separatore. Na rezultatima ovog rada predložen je prvi metod koji koristi riječi kao osnovu modela normalnog ponašanja [96]. Ideja tog rada je zasnovana na analizi sličnoj onoj koja se koristi za jezičku analizu teksta radi kategorizacije [123] a koristi geometrijsko predstavljanje riječi [124]. Ono što je iz tog rada bitno i što će ovdje biti iskorišteno, je prijedlog skupa separatora za HTTP protokol.

Na osnovu 15 predloženih znakova iz [96], analize HTTP protokola i analize saobraćaja, te dodatnih testiranja sa stvarnim testnim saobraćajem, izabran je skup separatora od 20 znakova. Testiranjem je utvrđeno da se tako dobija najveći procenat smisaonih riječi iz sadržaja normalnih paketa. Smisaone riječi su ili ključne riječi HTTP protokola ili riječi prirodnog jezika koji koristi Web dokument. Što ima više riječi koje imaju smisla, sistem bi trebao bolje „razumjeti“ tekst i bolje prepoznavati nenormalne zahtjeve. Taj skup od 20 separatora je:

TAB LF CR SPACE " & ( ) , . / : ; < = > ? [\ ]

Bajtovi koji nemaju vizuelnu prezentaciju napisani su sa svojim uobičajenim engleskim skraćenicama. ASCII vrijednosti ovih znakova su:

9 10 13 32 34 38 40 41 44 46 47 58 59 60 61 62 63 91 92 93

Ovdje je neophodno staviti napomenu da se ispostavlja da izbor skupa separatora nije od presudnog značaja, što će se pokazati u narednom poglavlju.

Ako se dosljedno primijeni definicija da je riječ niz uzastopnih bajta između dva separatora, dobivene riječi mogu imati proizvoljnu dužinu. Da bi se smanjio broj riječi i tako pojednostavio model, uvedena su ograničenja na maksimalnu i minimalnu dužinu niza. Minimalna dužina niza je tri bajta. Pošto ovi nizovi bajta nisu riječi nekog jezika oni ne moraju neophodno imati smisao. Cilj je da se prepoznaju nizovi koji su neobični i predstavljaju napad ili neki njegov dio. Niz bajta koji je kraći od tri je prekratak da bi bio bitan dio bilo kakvog napada. Maksimalna dozvoljena dužina niza je 16. Ako nakon 16 bajta nema separatora niz se završava i slijedeći bajt predstavlja početak novog niza - riječi. Ovo ograničenje na dužinu uspješno je primjenjeno u [96].

Nakon što je utvrđen način formiranja riječi za model, bilo je potrebno napraviti efikasan način pohranjivanja riječi. Za svaku riječ koja se pojavljuje u normalnom saobraćaju vodi se evidencija koliko puta se pojavila. Uobičajen način pohranjivanja ovakvih informacija su *hash* tabele, koje omogućavaju brzo pohranjivanje i iščitavanje informacija uz efikasno korištenje memorijskog prostora.

Za svaku riječ izračuna se vrijednost *hash* funkcije nad tom riječi. Rezultat h*ash* funkcije je indeks, jedinstvena vrijednost različita za svaku riječ. Veličina indeksa je u pravilu mnogo manja od veličine riječi. Umjesto pohranjivanja parova (riječ, broj pojavljivanja) pohranjuju se parovi (indeks, broj pojavljivanja) što je prostorno efikasnije. Uobičajeni način pohranjivanja ovih parova je tabela, odnosno niz, jer se indeks može generisati kao cijeli broj i biti indeks niza. Ovaj niz naziva se *hash* tabela. Za pretraživanje i ažuriranje

vrijednosti potrebno je obaviti dvije operacije *hash*-iranje i čitanje/pisanje u niz na mjesto na koje pokazuje indeks.

Postoji čitav niz pitanja oko izbora *hash* funkcije i uslova koje ona treba da zadovoljava. Očigledno je da mora biti brza i ne smije za dva različita ključa generisati isti indeks, što se naziva kolizija. Više detalja o *hash* funkcijama može se naći u [125]. U ovom radu korištena je *hash* funkcija koju je dao Bob Jenkins u [126], a koja se u sličnim namjenama pokazala kao brza i pouzdana.

Pomoću brojanja riječi u normalnom saobraćaju dobijaju se njihove frekvencije pojavljivanja. U fazi detekcije frekventna raspodjela riječi u sadržajima paketa koji se analiziraju poredi se sa raspodjelom normalnog saobraćaja. Pretpostavka je da će zloćudni paketi imati raspodjelu bitno drugačiju od normalne. Prvi dio modela čini frekventna raspodjela riječi normalnog saobraćaja nazvana model frekvenci.

3.2.2 Redoslijed riječi

U smisaonim rečenicama govornog jezika riječi obično imaju neki redoslijed. Riječi u HTTP zahtjevu takođe imaju uobičajeni redoslijed. Za svaku riječ moguće je utvrditi vjerovatnoću da poslije nje dolazi neka duga riječ. Vjerovatnoće prelaza sa riječi na riječ čine drugi dio predloženog modela normalnog sadržaja mrežnih paketa, nazvan model prelaza. Pretpostavka je da će zloćudni paketi imati znatno drugačiji redoslijed riječi od normalnih, što će se moći utvrditi korištenjem matrice prelaza. Ideja je bliska Markovljevom modelu [127] čija je upotreba za otkrivanje anomalija predložena još u prvom radu o sistemima za otkrivanje upada [39].

Posebno pitanje predstavlja način pohranjivanja vjerovatnoća prelaza. Matrica koja sadrži vjerovatnoće svih prelaza ima broj elemenata jednak kvadratu broja riječi, što može biti preveliko za praktičnu upotrebu. Srećom, statističke osobine sadržaja paketa dozvoljavaju da se za model prelaza koristi matrica

reducirane dimenzije. Konstrukcija reducirane matrice biće objašnjena tokom prikaza realizacije i testiranja.

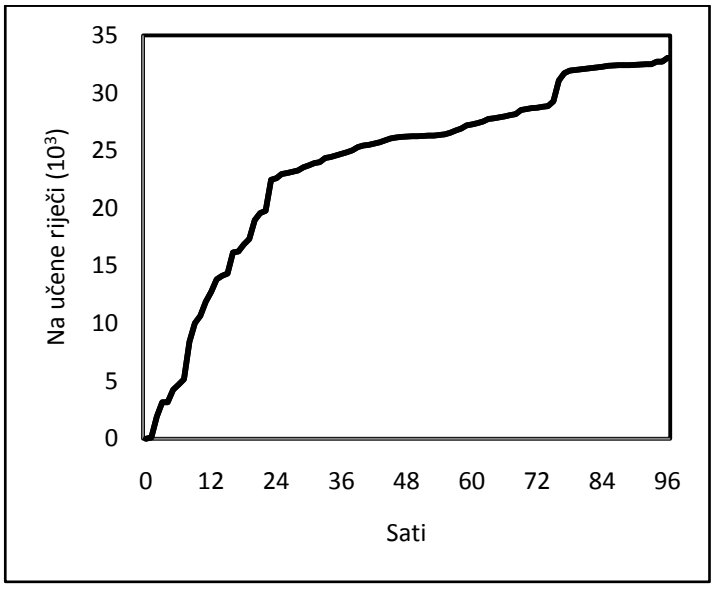
Metoda se zasniva na pretpostavci da će model frekvenci kombinovan sa modelom prelaza biti dovoljan za sigurnu detekciju napada. Druga pretpostavka je da je upotreba ovakve kombinacije otporna na imitacijske napade. Ovi napadi pokušavaju napadačke pakete napraviti sličnim normalnom paketu tako da u njega dodaju što više uobičajenih elemenata modela normalnog ponašanja. Konkretno, za model koji je zasnovan na frekvencijama pojavljivanja riječi, napadači mogu u napadački paket pored zloćudnog sadržaja ubaciti uobičajene riječi iz normalnog saobraćaja. Ako se to uradi, frekventna raspodjela riječi u napadačkom paketu će manje odudarati od normalne. To nije uvijek moguće niti lako uraditi, ali jeste izvodljivo. Slično dodavanje riječi koje bi se uklapalo u model prelaza je takođe teoretski izvodivo. Međutim, svaka promjena napadačkog paketa mora biti takva da ne naruši željeni, napadački efekat paketa, pri čemu se mora voditi računa i o ograničenoj veličini paketa. Ubacivanje kombinacija riječi u sadržaj napadačkog paketa, uz navedena ograničenja, znatno je teže nego ubacivanje pojedinačnih riječi.

Kriterijum odstupanja od modela na kom se zasniva detekcija takođe je napravljen tako da bude otporan na ubacivanje uobičajenih riječi i prelaza u napadačke pakete. Ova osobina će biti predstavljena kada bude objašnjavan metod računanja odstupanja.

3.2.3 Učenje riječi

Prvi korak u realizaciji i testiranju predložene metode bilo je učenje riječi koje se javljaju u sadržaju paketa testnog saobraćaja sa Elektrotehničkog fakulteta u Sarajevu, očišćenom od napada. Učenje riječi je zapravo pohranjivanje broja pojavljivanja svake riječi u *hash* tabelu. Sadržaj svih HTTP paketa upućenih ka Web serveru analiziran je i, na osnovu navedenog skupa od 20 separatora,

razdvojen na riječi. Svako pojavljivanje riječi povećavalo je vrijednost brojača za tu riječ u *hash* tabeli. Nakon analize 96 sati saobraćaja ukupan broj različitih riječi je prestao bitno da raste. Taj broj bio je nešto preko 33 000. Dodatni sati saobraćaja su malo uticali na povećanje broja riječi odnosno pojavljivanje novih riječi. Ovo je uzeto kao indikacija da je model dovoljno naučio normalno ponašanje sistema. Slika 4. prikazuje broj naučenih riječi kao funkciju broja sati analiziranog saobraćaja.



Slika 4. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja

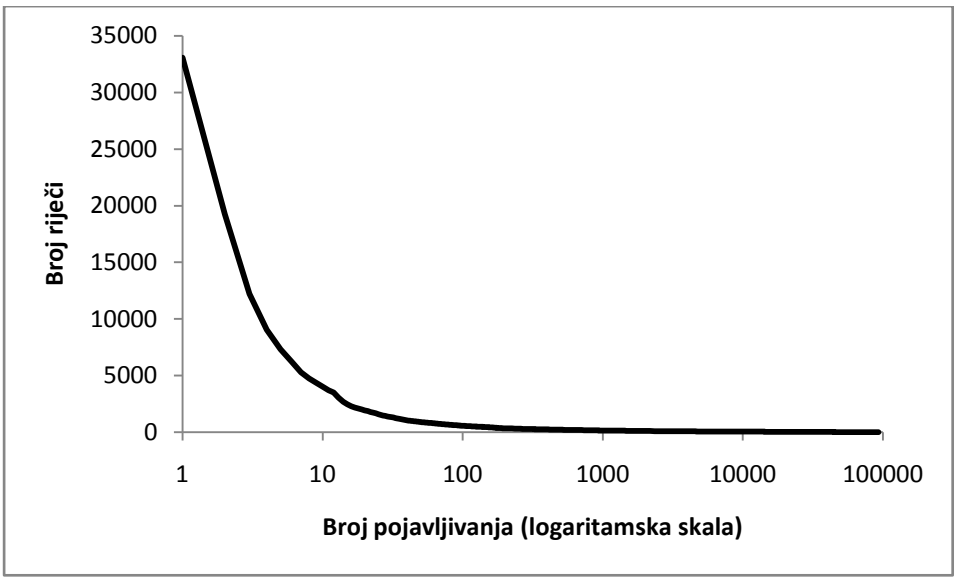
Pitanje kada je sistem dovoljno naučio je inače jedno od otvorenih pitanja mašinskog učenja. Jedino rezultati testiranja pokazuju da li je pretpostavka o dovoljnoj količini učenja tačna. Ako je sistem dovoljno naučio, broj lažnih uzbuna će biti mali jer će biti mali broj novih riječi i prelaza među normalnim paketima. U suprotnom slučaju, broj lažnih uzbuna će biti veliki.

3.2.4 Učenje prelaza

Sljedeći korak učenja je bio evidentiranje prelaza. Za to treba naći koliko puta se posle neke riječi pojavljuje svaka od riječi. Iz toga se mogu izračunati

elementi matrice prelaza. Dimenzije ove matrice bi bile 33 000 x 33 000. Ovakva matrica bi zahtijevala ogroman memorijski prostor i neprihvatljiva je za praktičnu realizaciju.

Frekvencija broj riječi u *hash* tabeli ima neravnomjernu raspodjelu. Neke riječi se u normalnom saobraćaju pojavljuju nekoliko desetina hiljada puta, a neke samo nekoliko puta. Slika 5. pokazuje kako se veliki broj riječi pojavljuje samo nekoliko puta, a jako mali broj preko 1000 puta. Graf je dat u logaritamskoj skali radi bolje preglednosti. Ova distribucija predstavlja standardni Zipf-ov zakon [128] koji je u analizi teksta dobro poznat i korišten. Prema ovom rezultatu čini se da se i sadržaj HTTP paketa ponaša po ovom zakonu. U dostupnoj literaturi nije bilo moguće pronaći ovu vrstu testiranja i zaključaka o odnosu HTTP poruka i Zipf-ovog zakona. Teorija prikupljanja informacija analizira Zipf-ov zakon i dokazuje da riječi u sredini ove distribucije imaju najveće značenje. Ova činjenica iskorištena je da se umjesto velike i potpune rijetke matrice prelaza koristi dosta manja, bolje popunjena matrica prelaza. Ova reducirana matrica prelaza formirana je naime samo između riječi koje se u normalnom saobraćaju pojavljuju više od 10 puta. Takvih riječi je 11,24% pa je reducirana matrica oko 80 puta manja. Ostali prelazi se ne evidentiraju i smatraju se rijetkim, odnosno neuobičajenim. Pretpostavlja se da se na ovaj način znatno ne gubi informacija, a olakšava se praktična realizacija. Kako bi pored nefrekventnih trebalo odbaciti i najfrekventnije riječi to će biti učinjeno prilikom računanja odstupanja.



Slika 5. Broj riječi u hash tabeli kao funkcija broja pojavljivanja (logaritamska skala)

3.2.5 Detekcija

Detekcija je proces u kom se analiziraju paketi koji nisu bili dio saobraćaja na osnovu kog se formirao model normalnog ponašanja. Ti novi paketi se analiziraju i utvrđuje se po nekom kriteriju njihovo odstupanje od modela, odnosno njihova neobičnost. Ako je model dobro napravljen i kriteriji odstupanja dobro postavljeni, zloćudni paketi će biti neobični. Normalni paketi bi se trebali dobro uklapati u model.

Kako je napravljeni model normalnog ponašanja napravljen na osnovu riječi može se smatrati nekom vrstom jezičkog modela. Ipak, neophodno je napomenuti da pošto se radi o sadržaju mrežnih paketa koji čine samo dijelove poruka ove riječi nisu prave riječi u smislu značenja ili pravila nekog jezika. Model će sa izabranim skupom delimitera podjeliti pakete u riječi koje bi trebale da imaju značenje u HTTP protokolu. Međutim ove rezervisane riječi protokola nemaju nastavke, odnosno ne mijenjaju se kao riječi govornog jezika po padežima, rodu, broju. Zbog ovoga ne postoje slične riječi, odnosno

riječi koje imaju zajednički neki dio. Pristupi analizi jezičkih modela mogu biti od koristi ali se ne mogu direktno primjeniti.

Primjenu n-grama za kategorizaciju teksta uveo je Suen [129]. Različite mjere sličnosti su korištene za poređenjue frekvencija n-grama. Unutrašnji proizvod između vektora frekvencija predložen je u [124], a Manhattan i Canberra udaljenost u [123]. Novi pristupi kategorizaciji teksta predlažu korištenje *kernel* funkcija kao mjere sličnosti, što omogućava razmatranje konteksta informacija [130][131][132].

Korištenje n-grama i riječi u sistemima za otkrivanje upada počelo je prvo kod sistema koji otkrivaju upade na računaru. Forrest i ostali bili su prvi i predložili su pravljenje baze svih n-grama sistemskih poziva koji su posljedica normalnog rada programa [97]. Neobičnim su se smatrali nizovi sistemskih poziva koji su odudarali od onih u bazi. Na ovim idejama predloženi su i drugačiji pristupi kao što je primjena sakrivenih Markovljevih modela u [133], vještačkih neuralnih mreža u [134] i indukcionih algoritama [135]. U novije vrijeme, modeli zasnovani na-gramima korišteni su i za otkrivanje zloćudnog koda u programima i dokumentima [136][137].

Prve upotreba n-grama za mrežne sisteme za otkrivanje upada počele su u zadnjih nekoliko godina i to pojedinačnim bajtima 1-gramima [74][77][92].

Jedina dva rada koja vrše analizu više-bajtnih nizova iz sadržaja paketa koriste sopstvene kriterije za neobičnost paketa. U [94] se koriste nizovi od n bajta (*n-gram*) gdje je n parametar sistema. Formula za računanje odstupanja od normalnog modela je jednostavna. Iznos odstupanja jednak je odnosu broja novih i ukupnog broja n-grama u paketu. Pretpostavka je da je paket neobičniji što ima više n-grama koji se nisu pojavili u saobraćaju za učenje. Dobro je što je formula jednostavna jer to znači da je računanje brzo. Nedostatak je što je neophodno imati potpuno čist saobraćaj, bez napada, za trening. Svaki napad koji se pojavi u saobraćaju za učenje biće apsolutno

nemoguće otkriti. Kako je potpuno čist saobraćaj gotovo nemoguće imati, to je ovaj nedostatak ozbiljna prepreka za praktično korištenje predložene metode.

Jedini metod koji koristi riječi kao osnovu modela normalnog ponašanja [96] radi analizu sličnu jezičkoj analizu teksta radi kategorizacije. Kriteriji neobičnosti računa se kao srednja udaljenost riječi od određenog broja najbližih susjeda. Autori ne daju brzinu procesiranja, a metoda izgleda računarski komplikovana, te bi mogla biti sporija nego što je prihvatljivo.

Kako u navedenim radovima nije pronađen odgovarajući način računanja odstupanja paketa od modela normalnog saobraćaja koji bi bio pogodan, odnosno dovoljno brz i tačan, predlaže se pristup baziran na ideji iz [94]. Potrebno je napraviti formulu po kojoj se računa odstupanje od predloženog modela koji se sastoji od frekvencije pojavljivanja riječi u saobraćaju za učenje. Formula treba da bude jednostavna, odnosno brza za računanje, i treba da toleriše eventualno mali broj napada u saobraćaju za učenje. Riječi koje se rijetko pojavljuju u normalnom saobraćaju treba da povećavaju iznos odstupanja. Riječi koje se često pojavljuju u normalnom saobraćaju treba da minimalno utiču na iznos odstupanja. Ako se za doprinos kriteriju neobičnosti riječi iz sadržaja paketa koji se analizira uzme inverzna vrijednost broja pojavljivanja te riječi u saobraćaju za učenje postići će se željeni efekat. Kriteriji neobičnosti riječi može biti prosječna vrijednost svih pojedinačnih doprinosa kriteriju neobičnosti riječi. Radi bržeg računanja iznosa kriterija neobičnosti riječi, inverzne vrijednosti broja pojavljivanja za sve riječi se mogu unaprijed izračunati i pohraniti prije procesa detekcije.

Slično rezonovanje može se primjeniti i na prelaze između riječi. U ovom slučaj bi doprinos nekog prelaza bio inverzna vrijednost broja takvih prelaza u normalnom – naučenom – saobraćaju. Usrednjena suma doprinosa bi predstavljala kriteriji neobičnosti prelaza. I za ovaj račun bi bilo pogodno

koristiti inverzne vrijednosti broja pojavljivanja prelaza u saobraćaju za učenje. Kombinacijom ova dva kriterija mogla bi se postići sigurnija detekcija napada.

3.2.5.1 Detekcija na osnovu kriterija neobičnosti riječi

Na osnovu prethodnog razmatranja napravljen je prvi pokazatelj u ovom radu kojim se mjeri neobičnost paketa. Za svaki paket izračunat je iznos odstupanja od modela po kriteriju neobičnosti riječi po formuli:

$$S_w = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n(w_i)} \qquad (1)$$

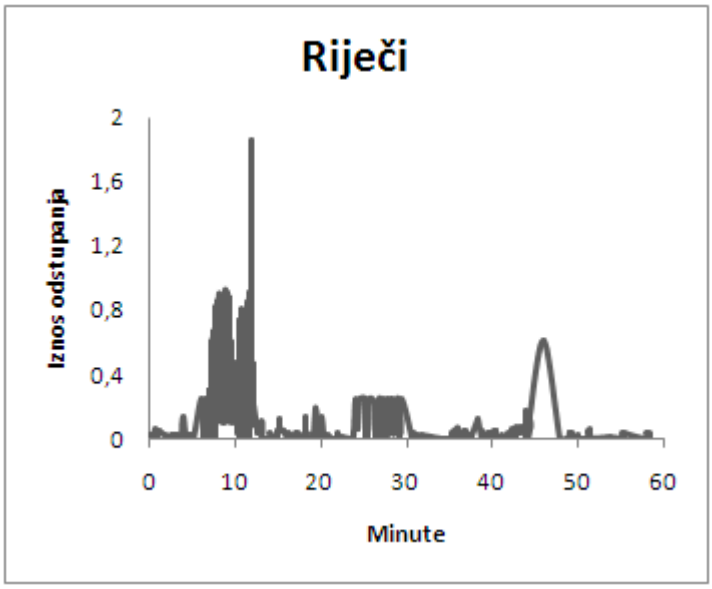
U formuli je k broj riječi u sadržaju paketa, a $n(w_i)$ je broj pojavljivanja riječi w_i u modelu normalnog ponašanja. Za riječi iz sadržaja paketa koji se analizira koje se nisu pojavile u normalnom modelu vrijednost $n(w_i) = 0$. U tom slučaju vrijednosti $1/n(w_i)$ bi bila beskonačna. Umjesto beskonačnosti vrijednost tog sabirka postavlja se na dva. Ova vrijednost je dvostruko veća od vrijednosti sabirka za riječ koja se pojavljuje samo jednom, za koju je $n(w_i) = 1$. Riječi koje se nisu pojavile u saobraćaju za učenje će imati najveći doprinos iznosu odstupanja, ali taj doprinos neće biti toliki da su doprinosi od riječi koje se rijetko pojavljuju zanemariv.

Riječi koje se rijetko pojavljuju u normalnom saobraćaju povećavaće iznos kriterija neobičnosti riječi. Sa druge strane, riječi koje se često pojavljuju u normalnom saobraćaju će vrlo malo doprinositi tom kriteriju. Na ovaj način znatno je smanjen uticaj mogućeg ubacivanja često korištenih riječi u zloćudni paket da bi izgledao sličniji normalnim paketima. Pretpostavlja se da je ovaj kriterij robustan u odnosu na manje prisustvo napada u saobraćaju za učenje. Naime, ako je u tom saobraćaju bilo malo napada, riječi koje se nalaze u napadu spadaće u rijetke riječi. Kada se analizira saobraćaj u kome se nalazi takav napad, kriterij će biti naravno manji nego da je učenje vršeno na saobraćaju bez tog napada, ali pošto će se javiti dosta rijetkih riječi, kriterij će

biti ipak dovoljno velik da ukaže na neobičnost sadržaja paketa. Ovo je veoma važna osobina jer je teško biti potpuno siguran da u saobraćaju za učenje zaista nema ni jednog napada. Ovakvim kriterijem neobičnosti riječi dobiva se praktično upotrebljiva metoda tolerantna na postojanje ograničenog broja napada u saobraćaju iz kog se pravi model normalnog ponašanja. Provjera sa postojanjem napada u saobraćaju za učenje biće napravljena kasnije.

Za prvu provjeru korektnosti iznesenih pretpostavki i pogodnost predložene formule računanja odstupanja od modela normalnog ponašanja po kriteriju neobičnosti riječi korišten je ranije pomenuti alat za otkrivanje sigurnosnih propusta na Web serverima, Nikto. Kao što je rečeno, Nikto pregleda sistem tako da generiše HTTP pakete različitog sadržaja koje upućuje Web serveru koji se testira. Tokom jednog sata normalnog saobraćaja pokrenut je Nikto pregled testnog Web servera. Sadržaj Nikto paketa bi trebao biti dovoljno različit od sadržaja paketa koji normalno dolaze do Web servera, tako da bi sistem za otkrivanje upada trebao moći otkriti Nikto testove. Ovaj sat saobraćaja snimljen je i analiziran. Za sadržaj svakog od paketa tokom ovog sata izračunato je odstupanje od modela po formuli (1).

Slika 6. prikazuje iznose odstupanja od modela tokom ovog sata. Nikto pregled trajao je oko šest minuta od šeste do dvanaeste minute. Iznosi za cijeli sat su prikazani da bi se vidjela razlika između normalnog i Nikto saobraćaja. Nikto saobraćaj je jasno vidljiv na slici. Iznosi po kriteriju neobičnosti riječi za ove pakete su osjetno veći nego za normalne pakete. Neki od normalnih, nenapadačkih ili preglednih, paketa imaju veći iznos odstupanja od drugih. Ovo pitanje će nešto kasnije biti razriješeno. Rezultati potvrđuju pretpostavku da će neuobičajeni paketi, kao što je Nikto pregled, imati različitu frekventnu raspodjelu riječi od normalnih. Takođe, formula (1) se pokazala kao potencijalno dobra za ocjenjivanje ove različitosti.



Slika 6. Iznos odstupanja po kriteriju neobičnosti riječi za sat saobraćaja koji uključuje Nikto pregled

3.2.5.2 Detekcija na osnovu kriterija neobičnosti prelaza

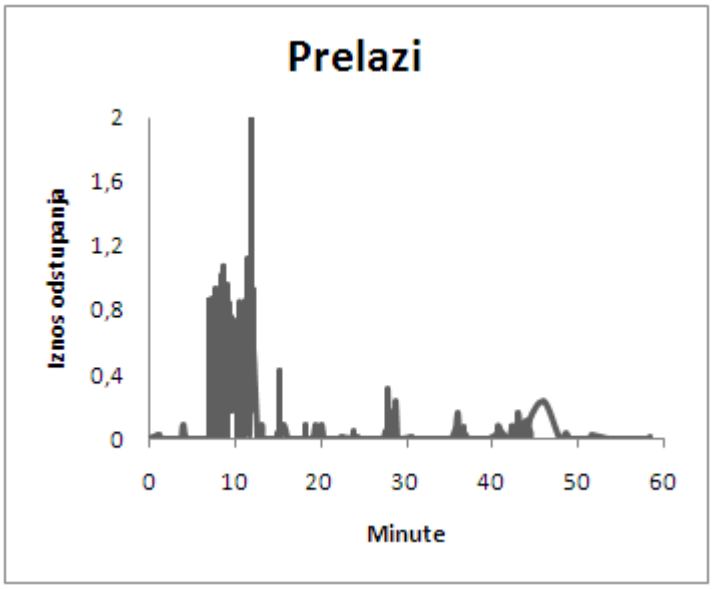
Prilikom učenja, odnosno kreiranja modela normalnog ponašanja, pored učestalosti pojavljivanja riječi, evidentirani su i prelazi između njih. Ova informacija iskorištena je za drugi kriteriji odstupanja paketa od normalnog modela. Drugi kriteriji zasnovan je na poređenju prelaza sa riječi na riječ između paketa koji se analizira i modela normalnog saobraćaja. Za svaki paket izračunat je iznos odstupanja od modela po kriteriju neobičnosti prelaza po formuli:

$$S_t = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n(t_i)} \qquad (2)$$

U formuli je m broj prelaza sa riječi na riječ u sadržaju paketa, a $n(t_i)$ je broj pojavljivanja prelaza t_i u modelu normalnog ponašanja. Vrijednost m je za jedan manja od vrijednosti k, broja riječi u sadržaju paketa, iz formule (1). Slično kao i kod prvog kriterija, mogući su prelazi koji ne postoje u modelu te

imaju vrijednost $n(t_i) = 0$. I u ovom slučaju umjesto beskonačne vrijednosti $1/n(t_i)$ vrijednost tog sabirka postavlja se opet na dva.

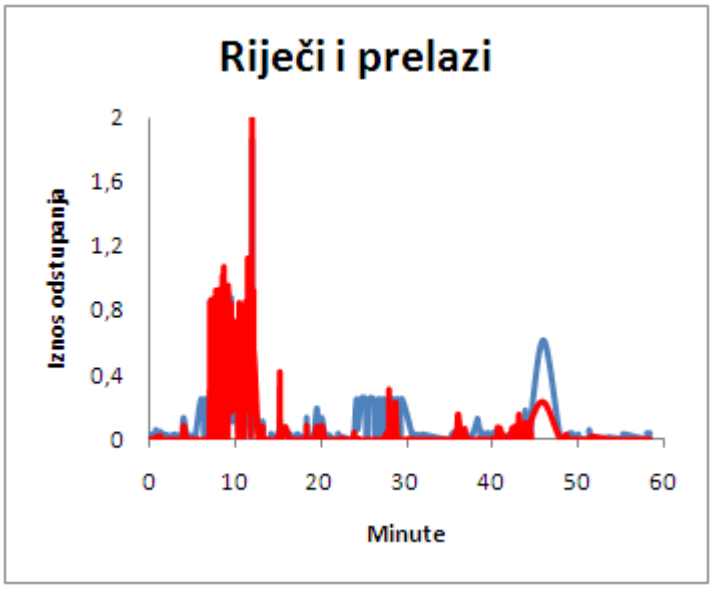
Formula (2) je vrlo slična formuli (1), pa se može primjeniti slično rezonovanje. Prelazi sa riječi na riječ u paketu koji se analizira, a koji ne postoje ili su rijetki u normalnom saobraćaju će povećavati iznos odstupanja. Prelazi koji su česti u normalnom saobraćaju malo će doprinositi zbiru. Ovo je u skladu sa ranije pomenutim iskustvom iz teorije prikupljanja informacija da pored neferekventnih prelaza, koji su odbačeni tokom formiranja modela, treba odbaciti i najfrekventnije, što je ovom formulom eftektivno i učinjeno. Radi provjere uspješnosti formule proveden je test sa istim satom saobraćaja kao i za prvi kriteriji. Rezultati su predstavljeni na slici 7. Na prvi pogled rezultati su vrlo slični onim na osnovu frekvencije riječi. Može se primjetiti da je iznos odstupanja za Nikto pakete uglavnom veći, a za normalne pakete uglavnom manji nego na slici 6.



Slika 7. Iznos odstupanja po kriteriju neobičnosti prelaza za sat saobraćaja koji uključuje Nikto pregled

3.2.5.3 Detekcija na osnovu upotrebe oba kriterija

Iako se na slici 7. zbog velikog broja paketa to ne vidi, pažljivijom analizom brojeva pokazala se jedna važna činjenica. Ako je neki normalan paket imao veći iznos odstupanja obično je to bilo ili po prvom, ili drugom kriteriju, a vrlo rijetko po oba. Sa druge strane, Nikto paketi su obično imali veće iznose odstupanja po oba kriterija. Slika 8. prikazuje iznose odstupanja po formuli (1) i po formuli (2) na istom grafu. Na ovoj slici se može uočiti gore navedena pojava.

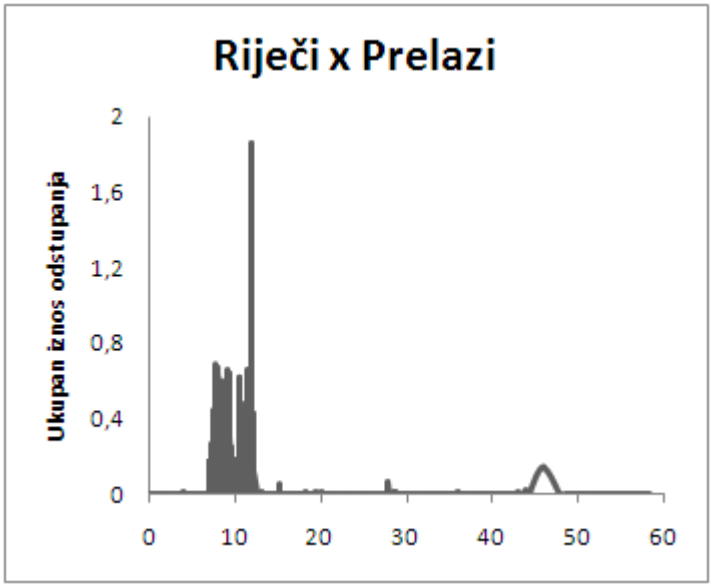


Slika 8. Iznosi odstupanja po kriteriju neobičnosti riječi i po kriteriju neobičnosti prelaza za sat saobraćaja koji uključuje Nikto pregled

Navedeno svojstvo je vrlo pogodno i ukazuje na koji način se mogu kombinovati gornja dva kriterija, odnosno formule za izračunavanje odstupanja od modela normalnog ponašanja. Pošto oba iznosa odstupanja trebaju biti velika da bi se paket smatrao neuobičajenim, množenje iznosa odstupanja po osnovu riječi i iznosa odstupanja po osnovu je logičan izbor. Na osnovu ovoga jednostavna formula za računanje ukupnog odstupanje glasi:

$$S = S_w * S_t \qquad (3)$$

S će biti nazvan kriteriji neobičnosti teksta. Grafički prikaz vrijednosti ovog kriterija za isti sat saobraćaja sa Nikto pregledom prikazan je na slici 9. Rezultati kriterija neobičnosti teksta su mnogo bolji od bilo kog od pojedinačnih kriterija, kako je i predviđano. Nikto paketi imaju znatno veće iznose odstupanja od modela normalnog saobraćaja nego normalni paketi. Iznosi odstupanja za normalne pakete su jako mali uz par manjih izuzetaka. Čak i ovi izuzeci imaju mnogo manje iznose odstupanja od Nikto paketa.

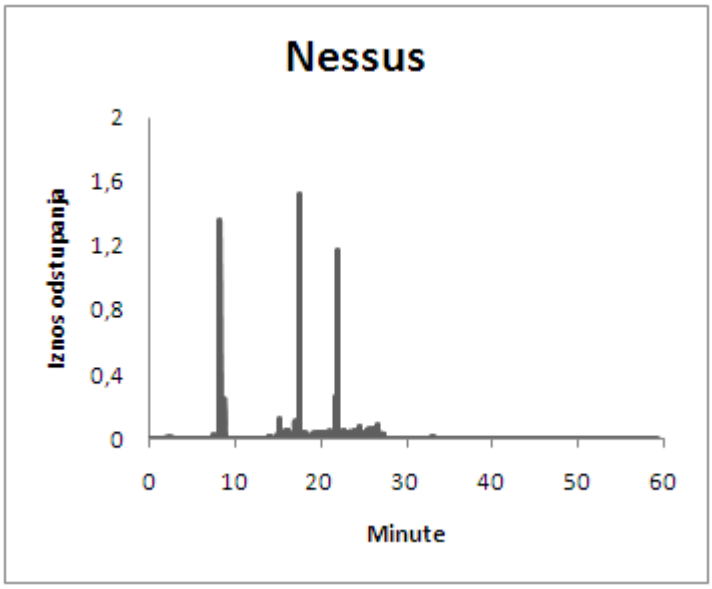


Slika 9. Ukupan iznos odstupanja po kriteriju neobičnosti teksta za sat saobraćaja koji uključuje Nikto pregled

Radi preglednijeg vizuelnog prikaza samo je polovina teoretske skale prikazana na slici 9. Grafički prikaz dat je radi principijelne ilustracije predložene metode. Stvarno otkrivanje upada koristi iznose, a ne njihove grafičke prikaze. Jedina stvar koja je bitna je da neuobičajeni paketi imaju znatno veće iznose odstupanja od normalnih paketa, što ovdje jeste postignuto.

Za dodatnu inicijalnu provjeru korektnosti iznesenih pretpostavki i predložene formule računanja odstupanja od modela normalnog ponašanja korišten je još jedan od ranije pomenutih alata za otkrivanje sigurnosnih propusta, Nessus. Nessus generiše pakete različitog sadržaja koje upućuje serveru koji se testira. Tokom jednog sata normalnog saobraćaja pokrenut je Nessus pregled testnog servera. Sadržaj Nessus paketa bi trebao biti dovoljno različit od sadržaja paketa koji normalno dolaze do servera, tako da bi sistem za otkrivanje upada trebao moći otkriti Nessus testove. Ovaj sat saobraćaja snimljen je i analiziran. Za sadržaj svakog od HTTP paketa tokom ovog sata izračunato je odstupanje od modela po formuli (3). Slika 10. daje grafički prikaz izračunatih iznosa odstupanja za sve pakete tokom tog sata. Nessus pregled trajao je oko 25 minuta od 4. do 29. minuta. Iznos odstupanja tokom ovih 25 minuta uglavnom je veći od iznosa odstupanja u periodu kad nije bilo Nessus saobraćaja. Međutim samo jedan dio paketa ima znatno veći iznos odstupanja. Razlog za ovo leži u činjenici da Nessus traži sigurnosne propuste za veliki broj aplikacija, a ne samo za Web server. Samo jedan dio Nessus saobraćaja sastoji se od HTTP paketa i sadržaji tih paketa su imali veliki iznos odstupanja. Paketi koji nisu HTTP nisu ni analizirani i u tim trenucima nema visokih iznosa odstupanja na grafu.

Nakon što su preliminarni testovi dali obećavajuće rezultate pristupilo se detaljnom testiranju predložene metode sa pravim savremenim napadima i stvarnim saobraćajem. Procedura i rezultati testiranje biće dati u nastavku.



Slika 10. Iznos odstupanja po kriteriju neobičnosti teksta za sat saobraćaja koji uključuje Nessus pregled

3.2.6 Testiranje metode

U prvom testu su računati kriteriji neobičnosti teksta za stvarne savremene napade. Za pravljenje ovih napada i za kreiranje mrežnih paketa kojim se ovi napadi prenose i izvršavaju korišten je ranije opisani alat za ovu namjenu Metasploit. Kako je HTTP protokol izabran kao testni, napravljeni su Web napadi koji pokušavaju iskoristiti slabosti u Web serverima ili drugim često korištenim Web aplikacijama. Kako je ranije objašnjeno, Metasploit napad, kao uostalom i pravi napad, sastoji se iz dva dijela. Prvi dio napada je kod napisan na osnovu sigurnosnog propusta koji se koristi, a drugi je izvršni kod napisan na osnovu željenog cilja napada. Izbor sigurnosnih propusta koje se pokušalo iskoristiti napravljen je tako da pokriva savremene operativne sisteme Windows, Linux i BSD, najčešće korištene Web servere kao što su Apache i IIS, te druge Web aplikacije. Sigurnosti propusti su iz različitih godina od 2001. do 2007. Skup sigurnosnih propusta korištenih za testiranje sa njihovim oznakama u CVE bazi, koja se uobičajeno referencira radi jednoobraznog označavanje dat je u Tabeli I.

Tabela I. Sigurnosni propusti na kojim su bazirani testni napadi (*exploits*)

Br.	Metasploit naziv	CVE oznaka
1	Apache Chunked-Encoding	2002-0392
2	Apache mod_jk overflow	2007-0774
3	Apache mod_rewrite	2006-3747
4	BSD Mercantec SoftCart CGI Overflow	2004-2221
5	HP OpenView Network Node Manager CGI Buffer Overflow	2007-6204
6	IIS 5.0 IDQ Path Overflow	2001-0500
7	IIS ISAPI w3who.dll	2004-1134
8	Oracle 9i XDB HTTP PASS	2003-0727
9	Xitami If_Mod_Since	2007-5067

Skup Metsploit kodova koji su korišteni kao drugi dio napada i koji definišu cilj napada na udaljenom računaru dat je u tabeli II.

Tabela II. Metsploit izvršni kodovi korišteni kao ciljevi testnih napada (*payloads*)

Br.	Metasploit kod	Objašnjenje
1	adduser	Dodaje korisnika
2	exec	Izvršava komandu
3	meterpreter-reverse_tcp	Otvara konekciju ka napadaču i ubacuje meterpreter server DLL
4	shell-bind_tcp	Očekuje konekciju i pokreće komandnu liniju

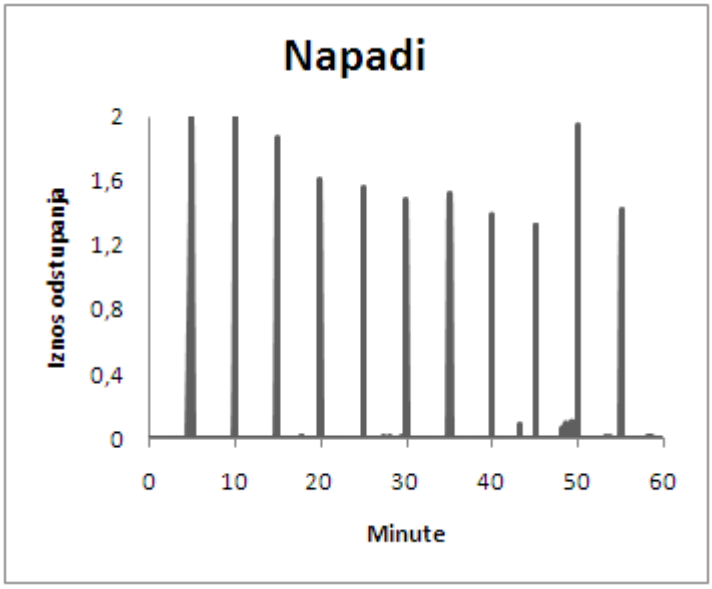
5	shell-reverse_http	Očekuje konekciju ka napadaču usmjerenu preko HTTP i pokreće komandnu liniju
6	shell-reverse_tcp	Otvara konekciju ka napadaču i pokreće komandnu liniju
7	vncinject-reverse_tcp	Otvara konekciju ka napadaču, ubacuje VNC server DLL i pokreće ga
8	vncinject-reverse_http	Otvara konekciju ka napadaču usmjerenu preko HTTP, ubacuje VNC server DLL i pokreće ga

Za prvi test napravljeno je 11 napada uparivanjem sedam slabosti i sedam izvršnih kodova. Korištene kombinacije su navedene u tabeli III. Ovih 11 napada ubačeno je u normalni saobraćaj tokom jednog sata. Svakih pet minuta pokrenut je jedan od napada. Slika 11. daje grafički prikaz izračunatih iznosa odstupanja za sve pakete tokom tog sata.

Tabela III. Kombinacije slabosti i izvršnih kodova korištene za prvi test

Br.	Sigurnosni propust	Izvršni kod
1	Apache Chunked-Encoding	meterpreter-reverse_tcp
2	Apache Chunked-Encoding	shell-reverse_http
3	Apache mod_jk overflow	adduser
4	Apache mod_rewrite	shell-bind_tcp
5	Apache mod_rewrite	vncinject-reverse_tcp
6	IIS 5.0 IDQ Path Overflow	shell-reverse_http
7	IIS 5.0 IDQ Path Overflow	shell-reverse_tcp
8	IIS ISAPI w3who.dll	exec

9	IIS ISAPI w3who.dll	shell-reverse_tcp
10	Oracle 9i XDB HTTP PASS	shell-reverse_tcp
11	Xitami If_Mod_Since	shell-reverse_tcp



Slika 11. Iznosi odstupanja za sat saobraćaja koji uključuje 11 napada

Na slici 11. jasno je uočljiv trenutak početka svakog napada. Pošto je kriteriji neobičnosti teksta računat za svaki paket, a ovih jedanaest napada ima različit broj paketa, širina vrhova koji indiciraju napade je različita. Vizuelni prikaz dat je samo kao ilustracija. Brojčani rezultati koji su presudni za detekciju su takođe odlični. Najmanji iznos kriterija od svih napadački paketa bio je 1,15. To je mnogo veće od iznosa odstupanja za bilo koji normalan paket. Ovaj test potvrđuje da metoda jasno razlikuje napadačke od normalnih paketa.

Radi detaljnijeg testiranja i numeričkog predstavljanja rezultata napravljene su dodatne kombinacije svih devet sigurnosnih propusta i svih osam izvršnih kodova. Ukupan broj napada povećan je na 17. Ukupan broj paketa u svim napadima zajedno bio je 197. Tabela IV daje pregled korištenih kombinacija.

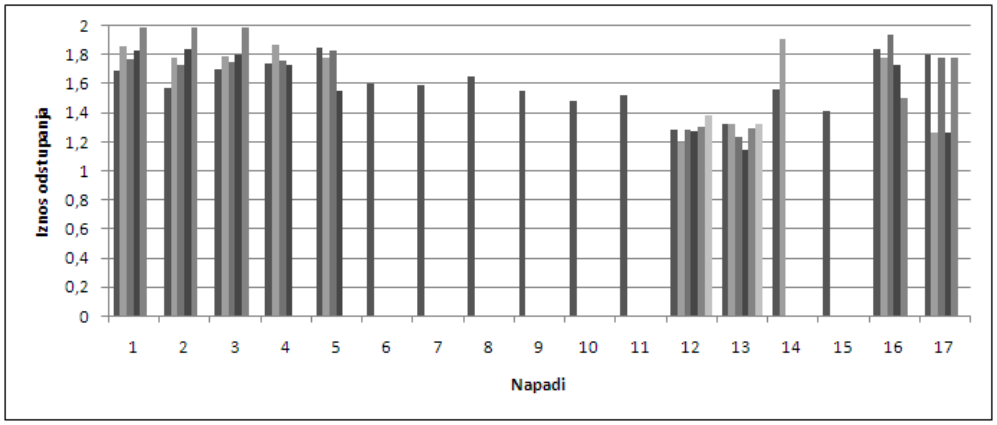
Tabela IV. Kombinacije slabosti i izvršnih kodova korištene za drugi test

Br.	Sigurnosni propust	Izvršni kod
1	Apache Chunked-Encoding	adduser
2	Apache Chunked-Encoding	meterpreter-reverse_tcp
3	Apache Chunked-Encoding	shell-reverse_http
4	Apache mod_jk overflow	adduser
5	Apache mod_jk overflow	shell-reverse_tcp
6	Apache mod_rewrite	shell-bind_tcp
7	Apache mod_rewrite	shell-reverse_tcp
8	Apache mod_rewrite	vncinject-reverse_http
9	Apache mod_rewrite	vncinject-reverse_tcp
10	IIS 5.0 IDQ Path Overflow	shell-reverse_http
11	IIS 5.0 IDQ Path Overflow	shell-reverse_tcp
12	IIS ISAPI w3who.dll	exec
13	IIS ISAPI w3who.dll	shell-reverse_tcp
14	Oracle 9i XDB HTTP PASS	shell-reverse_tcp
15	Xitami If_Mod_Since	shell-reverse_tcp
16	HP OpenView Network Node Manager CGI Buffer Overflow	shell-reverse_tcp
17	BSD Mercantec SoftCart CGI Overflow	shell-reverse_tcp

Na slici 12. prikazani su iznosi odstupanja za prvih šest paketa svakog napada. Iznos odstupanja sadržaja svakog paketa svakog napada je veoma veliki u odnosu na iznose za normalne pakete. I za ove napade najniži iznos za bilo

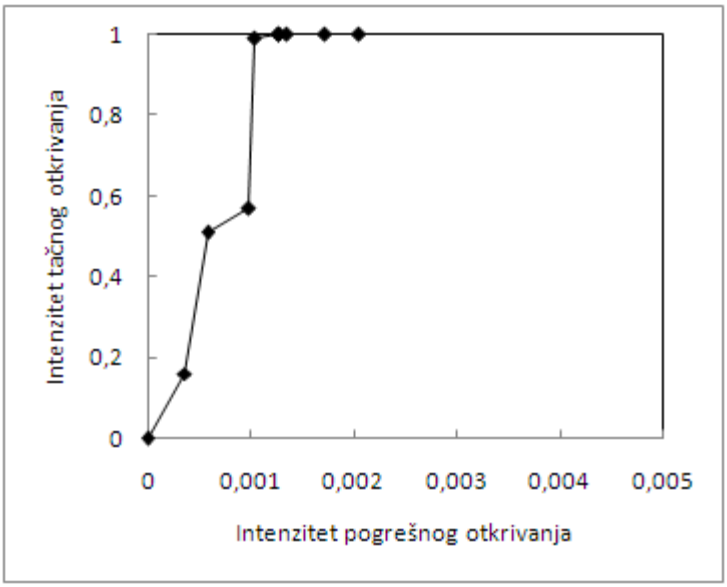
koji paket je 1,15. Ako se ovo postavi kao granica između normalnih i nenormalnih paketa svaki pojedinačni paket svih testiranih napada biva otkriven. Treba napomenuti da je za praktično otkrivanje napada često dovoljno otkriti samo jedan njegov paket. Najniži iznos kriterija za paket sa najvišim iznosom odstupanja u bilo kom od napada je 1,33. Sada se granica normalnosti može postaviti čak na 1,3; a da se otkriju svi testirani napadi.

Što je ova granica viša to su manje šanse da neki od normalnih paketa bude proglašen napadom odnosno da se pojavi lažna uzbuna. Kako su dosada testirani normalni paketi imali male iznose kriterija koji nikad nisu prešli vrijednost 0,2 izgleda da postoji veliki raspon u vrijednosti normalnih i nenormalnih paketa. Ovo je vrlo važna i dobra osobina koja indicira da bi metoda dala dobre rezultate i za druge napade i za druge profile normalnog saobraćaja.



Slika 12. Iznosi odstupanja paketa 17 testnih napada

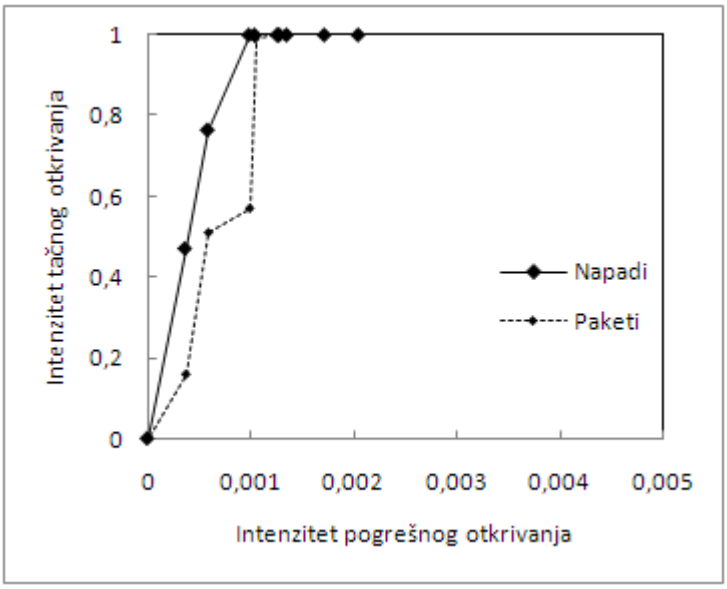
U slijedećem testu računate su vrijednosti kriterija za normalne pakete. Radi toga je analizirano šest dana saobraćaja sa Web servera Elektrotehničkog fakulteta u Sarajevu. Broj normalnih paketa sa visokim iznosom odstupanja od modela normalnog ponašanja bio je mali. Radi preglednijeg prikaza ukupne uspješnosti predložene metode izračunata je ROC kriva koje je prikazana na slici 13.



Slika 13. ROC kriva

Za granicu normalnosti uzimane su vrijednosti od 0,2 do 2, sa korakom 0,2 da bi se dobile tačke na krivoj. Skala intenziteta pogrešnog otkrivanja, umjesto uobičajenog raspona od 0 do 1, je samo od 0 do 0,005 da bi se bolje vidjele promjene krive u dijelu skale gdje se ona mijenja. Sama potreba za prikazivanjem manjeg dijela skale govori o uspješnosti metode. Kada se prag normalnosti postavi na 1 mogu se otkriti svi napadi uz prosječno 12 lažnih uzbuna na dan.

Važno je reći da je ROC kriva pravljena na nivou pojedinačnih paketa, a ne napada. Pošto je za otkrivanje pokušaja upada uglavnom dovoljno otkriti jedan od paketa napada, ROC kriva koju neki autori koriste, na kojoj se prikazuje intenzitet tačnog otkrivanja napada, a ne paketa, je još bolja, odnosno ima veći intenzitet tačnog otkrivanja. Ova ROC kriva za napade data je na slici 14. Na slici je i prethodna ROC kriva za pakete, nacrtana isprekidanom linijom, radi poređenja. U nastavku rada koristiće se ROC kriva na nivou paketa.



Slika 14. ROC kriva za napade

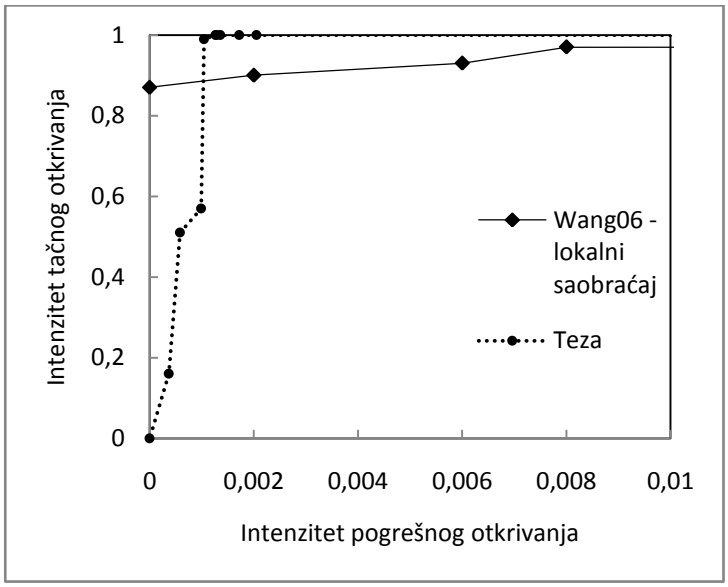
Tabela V daje jedan drugačiji prikaz uspješnosti, koji se ponekad koristi jer umjesto procenta navodi prosječan broj lažnih uzbuna dnevno.

Tabela V. Odnos procenta otkrivenih napada i broja lažnih uzbuna dnevno u zavisnosti od praga normalnosti

Prag normalnosti	Otkrivenih napadačkih paketa (%)	Lažnih uzbuna dnevno
	od **197** napadačkih paketa	od **9120** paketa
0,2	100%	19
0,4	100%	16
0,6	100%	12
0,8	100%	12
1	100%	12
1,2	100%	12
1,4	99%	10

1,6	57%	9
1,8	51%	5
2	16%	0

Radi vrednovanja rezultata ROC krive su upoređene sa novijim rezultatima autora koji se bave sličnom problematikom. Stvarna uporedba rezultata je otežana razlozima navedenim u [114]. To dolazi prije svega zbog korištenja različitih skupova napada i različitog normalnog saobraćaja. Dijagrami krivih su skalirani prema rezultatima.

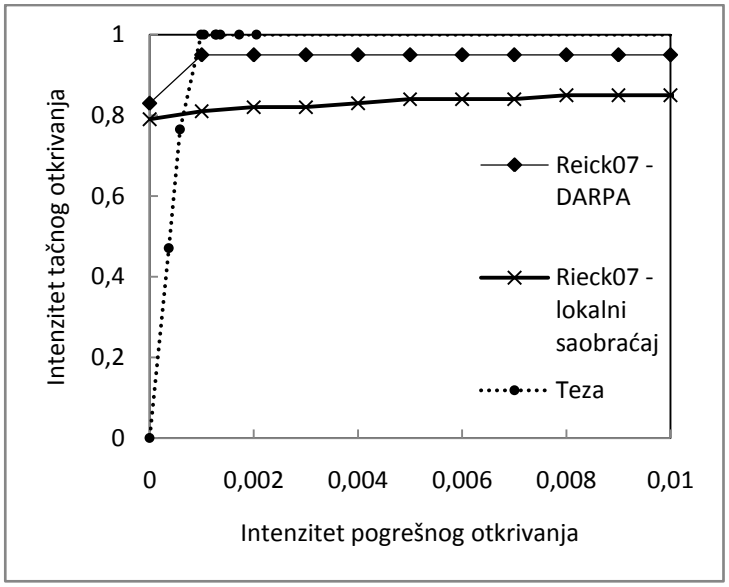


Slika 15. Uporedba ROC krive sa ROC krivom iz [94]

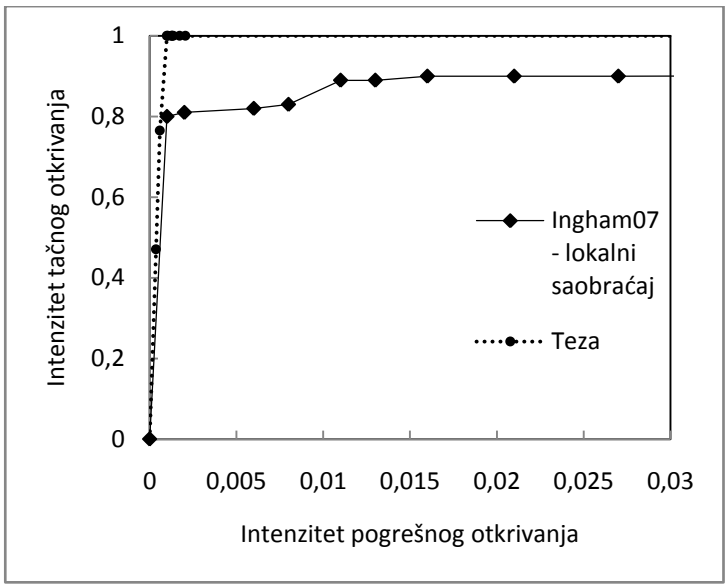
Na lici 15. su uporedno prikazane ROC kriva iz ove teze i ROC kriva iz [94]. Neophodno je napomenuti da je ovdje prikazana samo jedna kriva iz [94] i to ona najbolja. Samo jedna tačka, početna tačka, sa te krive je iznad ROC krive iz teze. Ukupna površina ispod ROC krive (AUC) iz teze je za 1,5% veća od AUC za drugu krivu. Treba reći da se u tom radu za testiranje koristio saobraćaj sa univerziteta autora i napadi iz tog saobraćaja, te drugi napadi iz

nepoznatog izvora. Ovaj saobraćaj i napadi nisu javno dostupni za analizu i poređenje.

Na lici 16. su uporedno prikazane ROC krive iz ove teze i ROC krive iz [96]. U tom radu su za testiranje korištena dva skupa podataka DARPA i onaj koji su autori sami napravili. Normalan saobraćaj za ovaj drugi skup je kombinacija vještački generisanog lokalnog saobraćaja i stvarnog saobraćaja sa interneta. Napadi su uglavnom napravljeni korištenjem istog alata kao i u ovoj tezi Metasploit. Ovaj testni skup podataka nije javno dostupan radi poređenja. Prikazane ROC su najbolje krive za HTTP saobraćaj iz tog rada. Većina radnih tačaka ROC krive iz teze ja bolja od tačaka sa druge dvije krive. AUC iz teze je za 2,5% veća od AUC za krivu sa DARPA testnim skupom i za 7,6% veća od krive za drugi testni skup.



Slika 16. Uporedba ROC krive sa ROC krivom iz [96]



Slika 17. Uporedba ROC krive sa ROC krivom iz [86]

Na lici 17. su uporedno prikazane ROC kriva iz ove teze i ROC kriva iz [86]. Neophodno je napomenuti da je ovdje prikazana samo jedna kriva iz [86] i to ona najbolja. ROC kriva iz teze je u potpunosti iznad druge ROC krive. Ukupna površina ispod ROC krive (AUC) iz teze je za 5,2% veća. Treba reći da se u tom radu za testiranje koristio saobraćaj iz organizacija autora i napadi iz tog saobraćaja. Ovaj saobraćaj nije javno dostupan za analizu i poređenje.

Ovdje su napravljene uporedbe samo sa najnovijim i najboljim rezultatima javno dostupnim u vrijeme pisanja teze iz radova koji se bave problematikom otkrivanja upada u sadržaju HTTP mrežnih paketa.

3.2.7 Otpornost metoda na napade u saobraćaju za učenje

U 3.2.5 iznesena je hipoteza da bi predloženi metod trebao biti otporan na manji broj napada u saobraćaju na osnovu kog se formira model normalnog ponašanja. Pretpostavka je zasnovana na načinu na koji se formulama (1), (2) i (3) izračunava kriteriji neobičnosti paketa. Riječi i prelazi koji se rijetko pojavljuju imaju mnogo veći uticaj na ukupni iznos odstupanja nego riječi i

prelazi koji se pojavljuju često. Ako u saobraćaju za učenje ima mali broj napada, riječi i prelazi iz sadržaja paketa tih napada bi i dalje trebali biti rijetki i rezultirati iznosom odstupanja mnogo većim nego za normalne pakete.

Radi provjere ove hipoteze obavljen je poseban test. Napadi 2, 3 i 8 iz tabele III, ubačeni su u testni saobraćaj. Tako su riječi i prelazi iz ovih napada postale dio modela normalnog ponašanja, i tako je dobiven drugi model normalnog ponašanja. Nakon toga su analizirani paketi sa tim istim napadima i izračunato je odgovarajuće odstupanje. Iznosi odstupanja za stari i novi model normalnog ponašanja su dati u Tabeli VI.

Tabela VI. Iznosi odstupanja sadržaja paketa istih napada prije nego su uključeni u saobraćaj za učenje i nakon toga

Broj napada u tabeli III	Stari model	Novi model
2	1,697458	0,568382
2	1,788083	0,900336
2	1,745897	0,877035
2	1,803798	0,004466
2	1,987421	0,000016
3	1,736318	0,872588
3	1,864706	0,936275
3	1,761585	0,885549
3	1,728683	0,000946
8	1,720629	0,484422
8	2,392497	0,602977

8	2,574378	0,64711
8	2,5483	0,639208
8	2,609535	0,655562
8	1,661021	0,416667

Iznosi odstupanja, nakon što su napadi uvršteni u saobraćaj za učenje, su se prepolovili. To nije dobro, ali je očekivano, a proizilazi iz formula za računanje iznosa odstupanja. Posmatrano drugačije, iznosi odstupanja za sadržaje svih paketa, osim tri, su još uvijek znatno veći od ovih iznosa za normalne pakete. Iako je iznos odstupanja za posljednja dva u prvom i posljednji paket u drugom napadu vrlo mali, postoji dovoljno paketa u svim napadima čiji je iznos odstupanja dovoljno veliki da se otkrije svaki od napada. Potrebno je reći da je ovo najnepovoljniji slučaj sa aspekta iznosa odstupanja. U saobraćaj za učenje, koji je navodno bez napada, ubačen je isti napad koji se kasnije pokušalo otkriti. Realnija situacija je da je u saobraćaju za učenje bilo napada, ali vjerovatno ne baš istih kao oni koje se pokušava otkriti. U tim slučajevima metoda bi dala još bolje rezultate.

Ovi rezultati potvrđuju određeni stepen otpornosti na nečist saobraćaj za učenje. Ovakav nečist saobraćaj je realnost i metode koje treba da se praktično upotrebljavaju morale bi biti spremne da izađu na kraj sa tim. Otpornosti na napade u saobraćaju za učenje je veoma važna osobina, ali nema puno metoda koje imaju tu osobinu. Jedna od rijetkih metoda koja koristi više bajta iz sadržaja paketa [94], uopšte ne može otkriti napade koji su postojali u saobraćaju za učenje.

3.3 Realizacija i performanse

Sistem na kom se realizira predložena metoda napravljen je kao program koji se izvršava na računaru, napisan u C programskom jeziku, a ima oko 1000

linija koda. Za snimanje mrežnog saobraćaja korištena je programska biblioteka *libpcap*. Za *hash*-iranje je korišten kod *hash* funkcije Boba Jenkinsa [126]. Kod je razvijen, preveden u izvršni i testiran koristeći razvojno okruženje KDevelop 3.5.1 na operativnom sistemu openSUSE 10.3. Računar na kom je rađeno testiranje ima centralnu procesorsku jedinicu AMD Athlon 3000+ na 2000 MHz i 1 GB radne memorije.

Performanse sistema, odnosno njegova brzina procesiranja paketa utvrđeni su na osnovu ranije opisanih testiranja. Ostvarena brzina iznosila je oko 100 paketa u sekundi. Ova brzina je bila više nego dovoljna da zadovolji potrebe ne testiranoj računarskoj mreži Elektrotehničkog fakulteta u Sarajevu koja ima 10 Mb/s vezu sa Internetom. Prosječno vrijeme potrebno da se obradi jedan dan snimljenog mrežnog saobraćaja bilo je oko pet minuta. Izražavanje navedene brzine u bitima u sekundi, nije jednostavno.

Gruba procjena mogla bi se napraviti ako bi se 100 paketa u sekundi pomnožilo sa dužinom paketa od 1500 bajta, pa sa osam da bi se dobili biti. U tom slučaju propusnost sistema bi bila 1,2 Mb/s, što izgleda malo u odnosu na savremene brzine koje se izražavaju u Gb/s. Ipak ovo nije loš rezultat u praksi. Naime, izmjerena brzina od 100 paketa u sekundi uključuje samo HTTP pakete i to samo one koji imaju sadržaj. Obavezni paketi koji inicijaliziraju i raskidaju TCP konekciju nisu uključeni u ovaj broj.

Pristup Internetu sa Gb/s brzinama je još daleko od uobičajenog. Nadalje, stvarni pristup Internetu, odnosno brzina kojom vanjski korisnici pristupaju Web serverima se uglavnom dijeli i sa drugim serverima tako da HTTP saobraćaj koristi samo dio dostupne propusnosti. Tu je još i pitanje brzine kojom štićeni Web serveri mogu da obrađuju zahtjeve. Sa druge strane izmjerena brzina je dobivena u razvojnom okruženju bez posebne optimizacije koda po brzini izvršavanja i na običnom korisničkom računaru starom preko dvije godine. Sistem se pokazao dovoljno brzim za praktičnu upotrebu u okruženju u kom je testiran. Također ima još dosta prostora za

povećanje brzine ako se za to ukaže potreba. Poređenje sa drugim rijetkim radovima koji predlažu sisteme koji sličnu analizu sadržaja paketa radi otkrivanja pokušaja upada [94] [96] [86] nije moguće jer njihovi autori ne daju ove podatke.

3.4 Zaključak

Predložena je metoda analize sadržaja mrežnih paketa. Metoda je zasnovana na riječima, nizovima uzastopnih bajta odvojenih separatorima. Utvrđen je skup od 20 separatora. Model normalnog ponašanja zasnovan je na frekvenciji riječi i frekvenciji prelaza sa riječi na riječ u sadržaju normalnih paketa. Stvarni saobraćaj ka serverima Elektrotehničkog fakulteta u Sarajevu, očišćen od napada, poslužio je za mašinsko učenje broja riječi i prelaza sa riječi na riječ. Za razvrstavanje mrežnih paketa na osnovu sadržaja na normalne i nenormalne, postavljena su dva kriterija. Jedan kriteriji zasnovan je na riječima, a drugi na prelazima između riječi u sadržaju analiziranog paketa. Kombinacijom ovih kriterija dobivena je konačna formula za računanje iznosa odstupanja od modela normalnog saobraćaja. Formula je jednostavna i brzo se računa jer se neki dijelovi kriterija mogu preliminarno izračunati i pohraniti. Uspješnost detekcije pomoću predloženog modela i kriterija provjerena je višestrukim testiranjem. Prvo je testirana sposobnost razlikovanje normalnog saobraćaja od nenormalnog saobraćaja koji nije klasični napad. Ova provjera obavljena je sa alatima za pronalazak sigurnosnih propusta. Napravljeni su i testovi sa većim brojem pravih napada. Konačno provjereno je koliko lažnih uzbuna metoda generiše sa stvarnim saobraćajem sa Elektrotehničkog fakulteta u Sarajevu. Svi testovi su dali očekivane dobre rezultate, koji se vide iz ROC dijagrama. Rezultati su bolji od rezultata drugih istraživača iz ove oblasti što je pokazano uporedbom ROC krivih i AUC. Pretpostavljena otpornost metoda na napade u saobraćaju za učenje je bila predmet slijedećeg testa. Test je potvrdio određeni stepen tolerancije na mali broj napada u saobraćaju za učenje.

Predloženi metod napravljen je da bude otporan na imitacijske napade. Napadački paketi koji pored zloćudnog sadržaja imaju i česte riječi i prelaze iz normalnog saobraćaja sa ciljem uklapanja u model trebali bi biti otkriveni zahvaljujući formulama koje su postavljene da se onemogući ovo prikrivanje. Premda su poduzete sve ove mjere još uvijek postoji teoretska mogućnost izbjegavanja otkrivanja napada izvedenog na ovaj način. U nastavku će biti predložen koncept koji bi trebao drastično smanjiti i tu teoretsku mogućnost.

4 SISTEM ZA OTKRIVANJE UPADA SA KLJUČEVIMA

Kod svakog sistema zaštite, kada se pronađe način da se otkrije i spriječi neki proboj sistema zaštite, napadači proučavaju metodu otkrivanja i sprečavanja da bi je mogli zaobići. Trenutno je veoma važna zaštita digitalnih sadržaja, gdje se ova trka odvija velikom brzinom. Svaki novi sistem zaštite od kopiranja muzike i filmova vrlo brzo biva zaobiđen. Najsvježiji primjer je *Advanced Access Content System* (AACS) [138], standard za distribuciju sadržaja i upravljanje digitalnim pravima. Ovaj standard je napravljen da zaštiti novu generaciju optičkih diskova, *Blue-ray* i HD-DVD od neovlaštenog pristupa i kopiranja. Standard je objavljen u februaru 2006 godine, a već u decembru iste godine su pronađeni akademski propusti [139] i objavljen praktičan način da se zaštita zaobiđe [140].

Slična stvar se dešava i sa sistemima za otkrivanje upada. Posebnu opasnost predstavljaju imitacijski napadi. Imitacijski napad je prvi put definisan u [141] kao zlćudni napadački kod koji imitira ponašanje aplikacije i tako izbjegava otkrivanje na osnovu anomalije. Ova definicija je data u kontekstu sistema za otkrivanje upada na računaru, ali je primjenljiva i na mrežne sisteme za otkrivanje upada. Prvi stvarni imitacijski napadi su opisani su relativno nedavno, 2002. godine, u [142] i [143]. Da bi se napravio imitacijski napad treba znati model normalnog ponašanja. Model se može napraviti na osnovu snimka normalnog ponašanja i poznavanja algoritma za izradu modela.

Ideje za izbjegavanje detekcije od strane mrežnih sistema za otkrivanje upada na osnovu anomalija novijeg su datuma. Na osnovu prijedloga objavljenih 2005. u [144] sljedeće godine se objavljuje se rad [73] i njegovo poboljšanje [145] koji pokazuje kako se mogu napraviti napadi koje ne može otkriti gotovo niti jedan od savremenih mrežnih sistema za otkrivanje upada.

Kriptografija je praktično uspjela riješiti problem probijanja zaštite na osnovu poznavanja metode, primjenom ključa. Savremene kriptografske metode su opšte poznate, a ipak ne postoje praktične metode za neovlašteno dešifriranje. Data Encryption Standard (DES) [146] bio je zvanični standard za šifriranje u SAD od 1977. do 2002., a u ostatku svijeta bio je najčešće korišteni standard. DES nikad nije bio probijen, to jest nije nađen način da se zaobiđe dešifriranje. Advanced Encryption Standard (AES) [147] je uveden jer savremeni računari mogu isprobati sve kombinacije DES ključa u dovoljno kratkom vremenu.

Metoda predložena u prethodnom poglavlju napravljena je da bude potencijalno otporna na imitacijske napade. Ipak, činjenica da je poznat način kreiranja modela i računanja odstupanja od njega uvijek omogućava razmatranje i pronalaženje načina uklapanja napada u model. Očigledno je da je potrebno pronaći pristup koji bi onemogućio i ovu teoretsku mogućnost. U nastavku će biti izloženo kako je sličan problem riješen u kriptografiji i na koji način se iste ideje mogu primijeniti na sisteme za otkrivanje upada.

Prvo će biti opisan osnovni kriptografski princip, a zatim će biti objašnjeno kako se taj princip može primjeniti na sisteme za otkrivanje upada. U ovoj metodi se pristup iz prethodnog poglavlja proširuje primjenom ključa kod definisanja normalnog ponašanja. Metoda zasnovana na ključevima biće testirana na jednom stvarnom sistemu.

4.1 Kerckhoffs-ov princip

Godine 1883. Kerckhoffs je predložio šest principa dizajna praktičnih kriptografskih šifratora [148][149]. Tih šest originalnih principa u slobodnijem prevodu glase:

1. Sistem mora biti praktički, ako ne matematički, nedešifrabilan;

2. Sistem šifriranja ne smije se oslanjati na tajnovitost svog načina rada i njegovo padanje u ruke neprijatelju ne smije predstavljati problem;

3. Razmjena ključeva među učesnicima u komunikaciji mora biti laka, a ključevi jednostavni za pamćenje. Zamjena ključeva novim mora biti lako izvodljiva po volji učesnika u komunikaciji;

4. Sistem mora biti kompatibilan sa telegrafskim (čitaj savremenim) načinom komuniciranja;

5. Sistem mora biti prenosiv i za njegovo korištenje ne smije biti potrebno više ljudi;

6. Konačno, shodno okolnostima primjene, sistem mora biti lak za korištenje i ne smije zahtjevati veliki mentalni napor ili pamćenje velikog broja pravila.

Drugi od ovih šest principa poznat je kao Kerckhoffs-ov princip i smatra se jednim od osnovnih postulata savremene kriptografije. Prvi dio principa govori o tome da se šifrirana komunikacija ne bi smjela oslanjati na uvjerenje da oni koji prisluškuju ne poznaju način šifriranja. Princip da se podrazumjeva da „protivnik poznaje sistem koji se koristi" navodi i Shannon 1949. u radu o teoriji tajnih komunikacija koji je postavio osnove savremene kriptografije [150]. Ovaj iskaz poznat je kao Shannon-ova maksima. Među osam osnovnih principa dizajna sigurnosti koje su 1975. predložili Saltzer i Schroeder [17] nalazi se i princip otvorenog dizajna koji kaže da sigurnost mehanizma ne treba da zavisi od tajnosti njegovog dizajna ili načina realizacije. Ovo je opšte prihvaćen princip sigurnosti i predstavlja suprotnost drugom pristupu koji se naziva sigurnost zasnovana na nepoznavanju sistema (*security through obscurity*).

Pošto se kriptografija bavi omogućavanjem tajnog komuniciranja, neophodno je da u sistemu komuniciranja, čiji dizajn nije tajan, postoji neka tajna

informacija koju znaju samo učesnici u komunikaciji. Drugi dio Kerckhoffs-ovog principa zapravo kaže da ta tajna informacija treba biti takve prirode da njeno otkrivanje ne predstavlja katastrofalan događaj radi kog se sistem više ne može koristiti. Sistem koji je pao protivniku u ruke može da se i dalje koristi, samo treba promijeniti tajnu informaciju koja se naziva se ključ. Termin dolazi iz oblasti fizičke sigurnosti gdje se ovaj princip odavno primjenjuje. Dizajn mehaničkih brava je uglavnom poznat, ali ne omogućava otvaranje brave. Za otvaranje je neophodan ključ koga bi trebalo da imaju samo ovlaštene osobe. Gubitak ključa ne zahtjeva dizajniranje novog tipa brave, već samo njenu promjenu i podešavanje za novi ključ. Ovaj pristup je primjenjen na savremenim elektronskim hotelskim bravama i omogućava da svaki novi gost pomoću jednostavnog reprogramiranja brave dobije svoj tajni elektronski ključ, pohranjen na kartici.

Kerckhoffs-ov princip se često iskazuje i na slijedeći način: „Sigurnost sistema ne leži u sigurnosti algoritma već u sigurnosti ključa“. Ključ može biti lozinka, PIN ili bilo šta što samo ovlašteni znaju ili imaju.

4.2 Primjena Kerckhoffs-ovog principa na sisteme za otkrivanje upada

Iako je navedeni princip odavno poznat i dugo se koristi, ne samo u kriptografiji i već i u drugim mehanizmima računarske sigurnosti, do sada nije korišten za sisteme za otkrivanje upada.

Primjena Kerckhoffs-ovog principa na sisteme za otkrivanje upada dala bi dodatnu zaštitu od napada. Ako je algoritam detekcije poznat, ali otkrivanje upada zavisi o nekoj tajnoj informaciji koja nije poznata napadaču, tada je vrlo teško napraviti napad koji neće biti otkriven, jer napadač ne zna kako takav napad treba da izgleda. U skladu sa Kerckhoffs-ovim pravilima svaka lokacija treba da ima svoj ključ, a u slučaju da neprijatelj otkrije ključ, treba da bude moguća njegova laka zamjena sa drugim ključem.

Naredno pitanje je na koji način traba napraviti takav sistem. Sistemi zasnovani na potpisima napada nisu dobri kandidati, jer samo prepoznaju potpise. Sistemi zasnovani na otkrivanju anomalija prave model normalnog ponašanja. Postoje različiti načini pravljenja modela, ali svima je zajedničko da imaju skup informacija koje prikupljaju i na osnovu kojih prave model, ta da imaju neki način poređenja događaja koji se analiziraju sa modelom. Izbor informacija, njihovo grupisanje i uticaj na model definišu se nekim skupom parametara. Različite vrijednosti tog skupa bi generisale različite modele za jedan isti skup normalnih događaja. Otkrivanje odstupanja od modela bilo bi takođe zasnovano na tom skupu koji bi mogao predstavljati tajnu informaciju i igrati ulogu ključa. Kako model i računanje odstupanja zavise od napadačima nepoznatih parametra, napadač ne zna šta treba imitirati. Ključ treba biti lako promjenljiv, a pravljenje novog modela brzo.

U nastavku će biti predstavljena jedna konkretna realizacija ove ideje koja koristi model normalnog ponašanja baziran na riječima i prelazima.

4.3 Realizacija sistema za otkrivanje upada sa ključevima

Metoda otkrivanja mrežnih upada predložena u prvom dijelu ovog rada zasniva se na analizi sadržaja mrežnih paketa podijeljenih na riječi. Riječ je definisana kao niz uzastopnih simbola između dva separatora. U sadržaju mrežnih paketa uzastopni znakovi koji čine riječi su bajti, a separatori su posebne utvrđene vrijednosti bajta. Na osnovu prijedloga nekih autora [95] [96] i na osnovu dodatnih testiranja sa stvarnim saobraćajem, utvrđen je skup od 20 separatora, koji je korišten za dalja testiranja metoda. Sa ovim skupom separatora postignuti su odlični rezultati detekcije koji su i predstavljeni u radu.

Tokom testiranja za utvrđivanje najboljeg skupa separatora utvrđeno je da se svaki skup separatora jednoznačno preslikava u skup normalnih riječi. Izabrani skup separatora daje najveći procenat smisaonih riječi za HTTP

saobraćaj. Međutim, sa aspekta otkrivanja anomalija nema nekog principijelnog razloga da se izabere neki skup separatora umjesto drugog. Svaki skup separatora proizvešće svoj skup „normalnih“ riječi. Ovaj skup normalnih riječi koristi se za pravljenje modela zasnovanog na učestalosti pojavljivanja riječi i prelaza sa riječi na riječ. Prilikom detekcije taj isti skup separatora koristi se za razdvajanje na riječi sadržaja paketa analiziranog mrežnog saobraćaja. Dobivene riječi i prelazi porede se modelom normalnog ponašanja i utvrđuje se odstupanje, te otkriva da li se radi o pokušaju upada ili ne.

Skup separatora čini se kao dobar kandidat za ključ. Koristeći jednu istu metodu generisanja modela za normalno ponašanje, neki skup separatora će se jednoznačno preslikati u model normalnog ponašanja. Napadači mogu znati metod generisanja i imati uzorke normalnog saobraćaja ali ako ne znaju skup separatora, biće im biti veoma teško da generišu model normalnog ponašanja. Zbog toga će generisanje modela biti teoretski moguće, ali će praktično biti neizvodivo, ili će zahtjevati previše vremena i resursa. Ako ne znaju skup separatora, napadači ne mogu znati šta su normalne riječi ili prelazi, pa ne mogu napraviti imitacijski napad.

Skup separatora kao ključ zadovoljava Kerckhoffs-ove principe. Svaki server ili mrežni segment može da ima svoj ključ. Skup separatora se lako može promijeniti i napraviti novi model normalnog ponašanja. Pravljenje novog modela nije teško ni dugotrajno, ako postoje adekvatni pohranjeni uzorci normalnog saobraćaja.

Prilikom izbora skupa znakova za metodu izloženu u prethodnom poglavlju vodilo se idejom iznesenom u [114]. Po toj ideji bolja detekcija postiže se ako model smislenije predstavlja HTTP zahtjeve. Zbog ovoga je izabran skup separatora na osnovu kojeg se dobije najveći broj smisaonih riječi. Promjenom separatora skup riječi će se promijeniti i više neće biti smislene.

Posljedica ovoga može biti veliko povećanje broja riječi. Takođe moguće je i odstupanje od teoretske distribucije.

Ranije u radu je utvrđeno da je broj pojavljivanja riječi u skladu sa Zipf-ovim zakonom, dobro poznatim i korištenim u analizi teksta. U uvodu u izdanje Zipf-ove knjige iz 1965 Miller je dao komentar da se i slučajno generisan tekst ponaša po Zipf-ovom zakonu [151]. Ovaj komentar dokazan je u [152]. U tom radu je pokazano slijedeće. Slučajan niz znakova iz ograničenog skupa uz proizvoljno izabran znak kao separator niza na riječi generiše frekventnu distribucija riječi u skladu sa poopštenim Zipf-ovim zakonom koji je dao Mandelbrot [153]. Ovaj poopšteni oblik ponekad se naziva i Zipf-Mandelbrot-ov zakon i pokazuje zavisnost učestalosti pojavljivanja riječi od njenog ranga. Ovaj zakon iskazuje se slijedećom formulom:

$$f = \frac{k}{(r + B)^{\propto}} \qquad (4)$$

Ovdje su B i α konstante, r rang riječi, a k takođe konstanta koja se uglavnom koristi za skaliranje.

Sadržaj HTTP zahtjeva može se posmatrati i kao niz slučajnih znakova (vrijednosti bajta) iz ograničenog skupa od 256 vrijednosti. Prijedlog je da se separatori biraju proizvoljno. Na osnovu gore pomenutog dokaza može se očekivati da promjena separatora ne utiče na odstupanje od distribucije. Ova pretpostavka biće provjerena testiranjem.

Što se povećanja broja riječi tiče može se očekivati određeno povećanje koje ne bi trebalo biti tako veliko. Razlog za ovo je slijedeći. Izabrani skup separatora trebao bi da podjeli HTTP zahtjeve na smisaone riječi. Za prve pakete u zahtjevima koji obično imaju ključne riječi i URL-ove (adrese) ovo će biti slučaj. Za priličan broj paketa, pogotovo one koji ne prenose tekstualni sadržaj, neće biti generisane smisaone riječi, poput ključnih riječi HTTP

protokola ili riječi nekog jezika. Rezultat je da optimizirani skup separatora zaista generiše jedan broj smisaonih riječi, ali takođe generiše i besmislene riječi. Slučajno izabrani skup separatora će rezultirati sa manje smisaonih riječi, ako ih uopšte bude, nego optimizirani skup separatora, ali broj besmislenih nema razloga da bude veći. I ova pretpostavka će biti provjerena testiranjem.

Najvažnija i prva pretpostavka da promjena skupa separatora neće negativno uticati na mogućnost razlikovanja normalnog i zlоćudnog saobraćaja za konkretnu metodu, biće temeljito testirana u nastavku.

4.4 Testiranje mogućnosti otkrivanje upada za sistem sa ključem

Testiranje predloženog sistema za otkrivanje upada kod kojeg je model normalnog ponašanja napravljen sa ključem izvršeno je na isti način kao i za sistem opisan u prvom dijelu rada. Za učenje riječi i prelaza korišten je ranije opisani testni saobraćaj sa Elektrotehničkog fakulteta u Sarajevu, koji je prethodno očišćen od napada.

Za provjeru uspješnosti otkrivanja nenormalnog saobraćaja i napada korišten je ranije opisani saobraćaj dobiven korištenjem alata za otkrivanje sigurnosnih propusta, Nikto i Nessus, i razvojnog okruženja koje omogućava kreiranje različitih napada, Metasploit. Testiranje sa istim podacima omogućava upoređivanje rezultata. Cilj testiranja je da se provjeri da li i sistemi sa proizvoljnim skupom separatora uspješno razlikuju normalan saobraćaja i pokušaje upada.

U cilju provjere obavljen je veći broj testova sa različitim skupovima separatora. Varirana je veličine skupa (broj separatora) i njegovi elementi (vrijednosti pojedinih separatora). Rezultati su upoređeni sa onim za originalni skup separatora, koji će se dalje nazivati optimizirani skup. U nastavku će biti

dati detaljni rezultati za dva skupa, te zbirni rezultati za ostale testirane skupove. Način izbora separatora biće objašnjen uz svaki test.

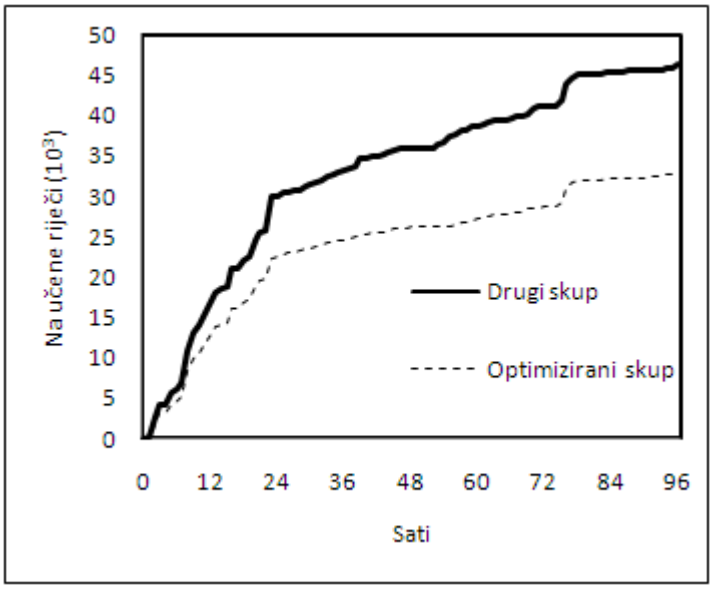
4.4.1 Test sa 20 separatora

Za prvi test skup separatora imao je isti broj elemenata kao i optimizirani skup. Članovi su bile slučajno izabrane ASCII vrijednosti između 9 i 127. Ostale moguće ASCII vrijednosti nisu korištene, jer predstavljaju znakove koji se rijeđe pojavljuju u tekstualnim protokolima kao što je HTTP. Kasnije su rađeni i testovi bez ovog ograničenja čiji će rezultati biti izloženi u narednim poglavljima. U ovom, i svim drugim testovima, slučajno izabrane ASCII vrijednosti dobivene su korištenjem Linux komande „rand" za generisanje pseudo slučajnih brojeva u zadatom opsegu. U daljem tekstu kada se kaže da su elementi skupa separatora generisani slučajno misli se korištenjem komande „rand". ASCII vrijednosti elemenata skupa separatora za ovaj test bile su:

15, 19, 20, 30, 35, 37, 38, 41, 47, 56, 58, 63, 68, 69, 90, 97, 107, 109, 114, 122

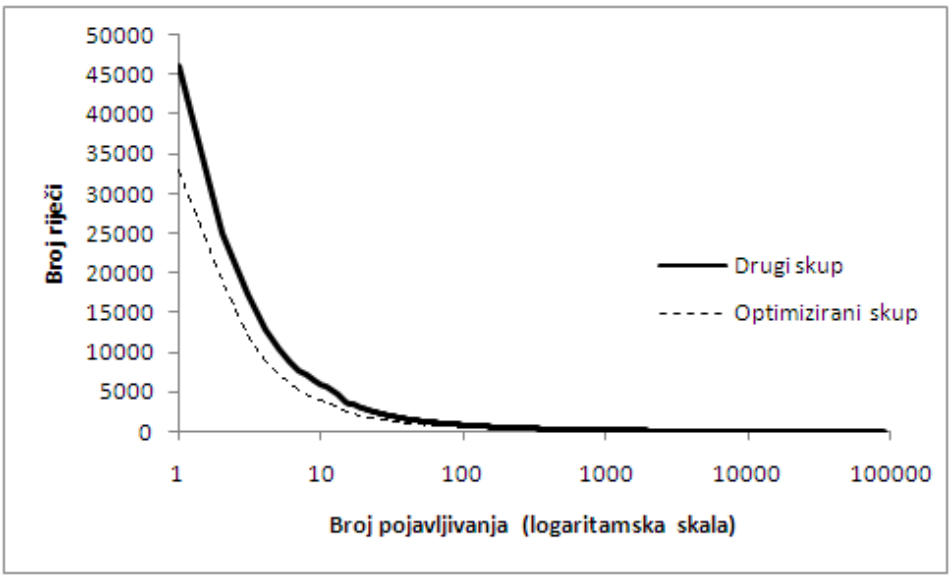
Sa ovim skupom separatora izvršeno je učenje riječi iz istog normalnog saobraćaja kao i za originalni sistem. Slika 18. prikazuje broj naučenih riječi u zavisnosti od broja sati normalnog saobraćaja korištenog za učenje.

Radi poređenja isprekidanom linijom je prikazan i broj riječi za optimizirani skup. Ponovo se vidi da se broj naučenih riječi stabilizira nakon 96 sati učenja. To znači da je 96 sati dovoljno dug interval za učenje. Ukupan broj riječi je međutim bio preko 46 000, što je povećanje od oko 40% u odnosu na originalni sistem. Ovo se moglo očekivati, jer ovaj skup separatora nije optimiziran za podjelu sadržaja HTTP paketa na smisaone riječi. Ipak ovo povećanje se ne može smatrati ogromnim i preprekom za korištenje. Značajan porast broja riječi ipak ne prelazi memorijske zahtjeve za formiranje *hash* tabele na normalnom hardveru.



Slika 18. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za drugi skup separatora

Slika 19. pokazuje broj riječi u *hash* tabeli kao funkciju njihovog broja pojavljivanja. Graf je dat u logaritamskoj skali radi bolje preglednosti.



Slika 19. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za drugi skup separatora (logaritamska skala)

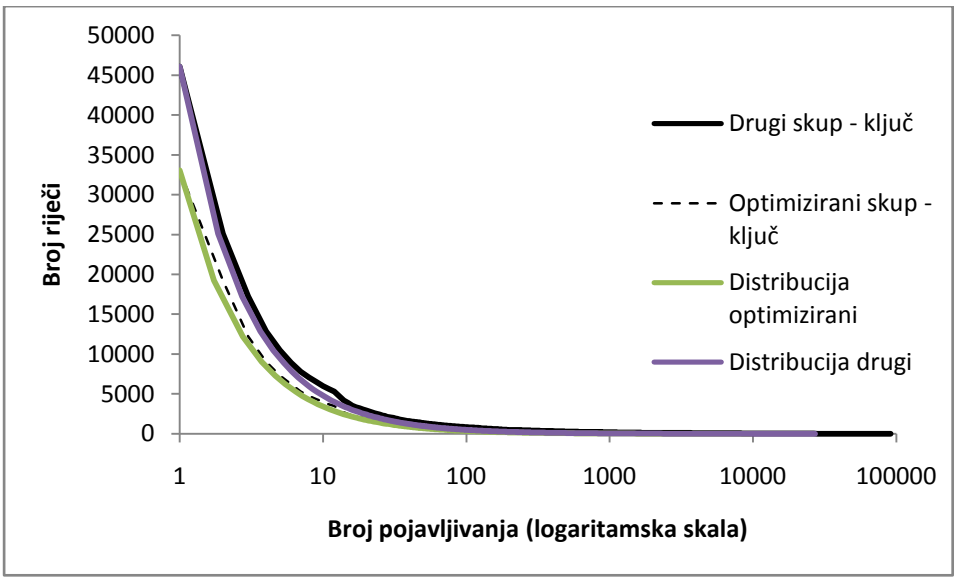
Ponovo je vidljiva ista tendencija kao i za optimizirani skup (prikazan isprekidanom linijom). Neke riječi se u normalnom saobraćaju pojavljuju nekoliko desetina hiljada puta, a neke samo nekoliko puta. Da bi se potvrdilo da se raspodjela nije promijenila izračunati su parametri raspodjele Zipf-Mandelbrot zakonu za obje krive.

Ukupan broj mogućih znakova za obje krive je 256 (2^8). Od ovih znakova 20 su separatori. Ovaj broj označimo sa S. Preostalo je 236 znakova od kojih se formiraju riječi. Ako ovaj broj označimo sa M, prema [152] moguće je izračunati parametre raspodjele (4) prema slijedećim formulama:

$$\propto = \frac{\log (\mathrm{M} + \mathrm{S})}{\log (\mathrm{M})} \qquad (5)$$

$$B = \frac{\mathrm{M} - 1}{\mathrm{M}} \qquad (6)$$

Izračunate vrijednosti su α = 1, 01489 i B = 1,00426. Na osnovu uvrštavanja ovih vrijednosti u formulu (4) i skaliranja prema broju riječi izračunate su krive raspodjele broja riječi i broja pojavljivanja za riječi dobivene sa optimiziranim i drugim, slučajnim, skupom separatora. Originalne krive i odgovarajuće izračunate distribucije prikazane su na slici 20.



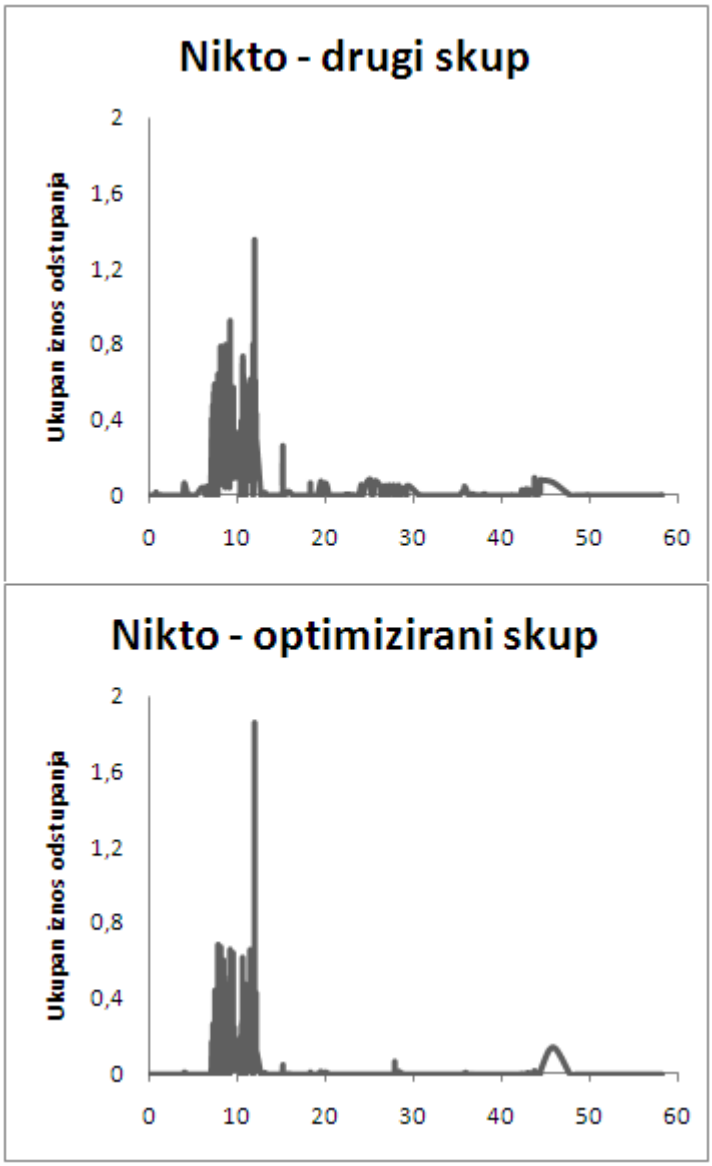
Slika 20. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za drugi skup separatora i aproksimacija formulom (4) (logaritamska skala)

Kako se vidi analitičke distribucije se gotovo potpuno poklapaju sa eksperimentalnim krivim. Uporedbom vrijednosti funkcije i eksperimentalnih podataka u pojedinim tačkama ustanovljene su određene male razlike do nekoliko procenata u dijelu skale do 100 pojavljivanja koji je najvažniji za način na koji su ovi rezultati iskorišteni.

Na osnovu analitičkog izraza za distribuciju moguće je izračunati da se 9,67% riječi pojavljuje više od 10 puta. Ovo je u skladu sa pretpostavkom korištenom prilikom kreiranja matrice prelaza za originalni sistem. Prema tome evidentiranje prelaza između riječi moguće je uraditi na isti način kao i za originalni sistem. Matrica prelaza formirana je samo za riječi koje se u normalnom saobraćaju pojavljuju deset ili više puta. Ostali prelazi se ne evidentiraju i smatraju se rijetkim, odnosno neuobičajenim.

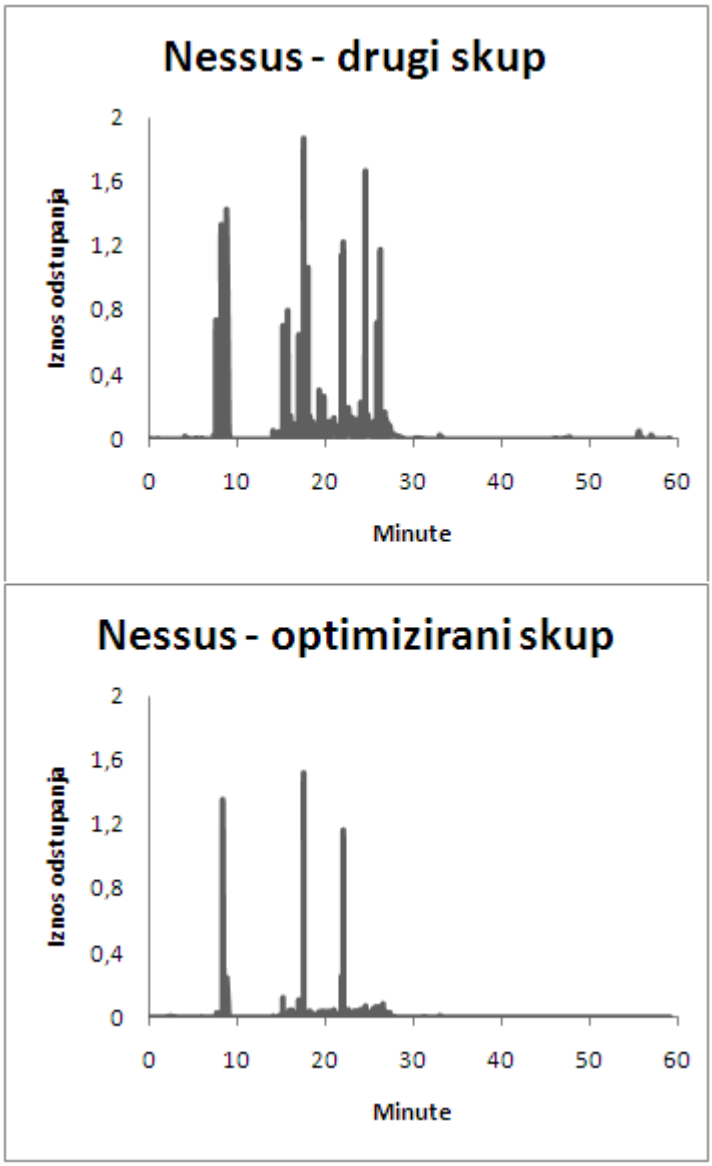
Nakon što je sistem sa ovim skupom separatora prošao proces učenja normalnih riječi i prelaza među njima, provjerena je sposobnost sistema da detektuje upad.

Za prvu provjeru napravljen je isti test, opisan za originalni sistem, sa istim satom normalnog saobraćaja tokom kog je pokrenut i Nikto pregled testnog Web servera. Slika 21. prikazuje iznose odstupanja od modela normalnog saobraćaja za sve pakete tokom tog sata. Rezultati su slični onim dobivenim sa optimiziranim skupom separatora, koji su takođe prikazani radi poređenja. Iznosi odstupanja za normalne pakete su nešto veći u prosjeku, ali su i iznosi odstupanja za Nikto pakete u prosjeku nešto veći. U svakom slučaju postoji jasna razlika između jednih i drugih paketa. Primjena stohastički izabranog skupa separatora nije u ovom slučaju negativno uticala na mogućnost detekcije.



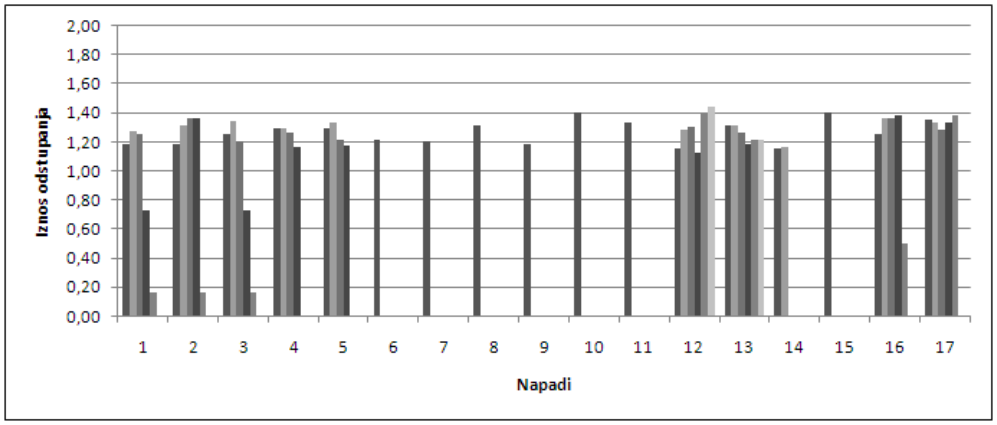
Slika 21. Iznosi odstupanja za sat saobraćaja koji uključuje Nikto pregled za drugi i optimizirani skup separatora

Rezultati drugog testa, sa satom normalnog saobraćaja tokom kog je pokrenut Nessus pregled testnog servera, prikazani su na slici 22. Rezultati su čak i bolji nego sa optimiziranim skupom separatora jer su iznosi odstupanja za Nessus pakete u prosjeku veći. Ni u ovom slučaju primjena stohastički odabranog skupa separatora nije negativno uticala na mogućnost detekcije.



Slika 22. Iznosi odstupanja za sat saobraćaja koji uključuje Nessus pregled za drugi i optimizirani skup separatora

Naredni test se bavio otkrivanja pravih napada. Korišteni su isti napadi kao i za originalni sistem navedeni u tabeli IV. Na slici 23. prikazani su iznosi odstupanja za prvih šest paketa svakog napada.



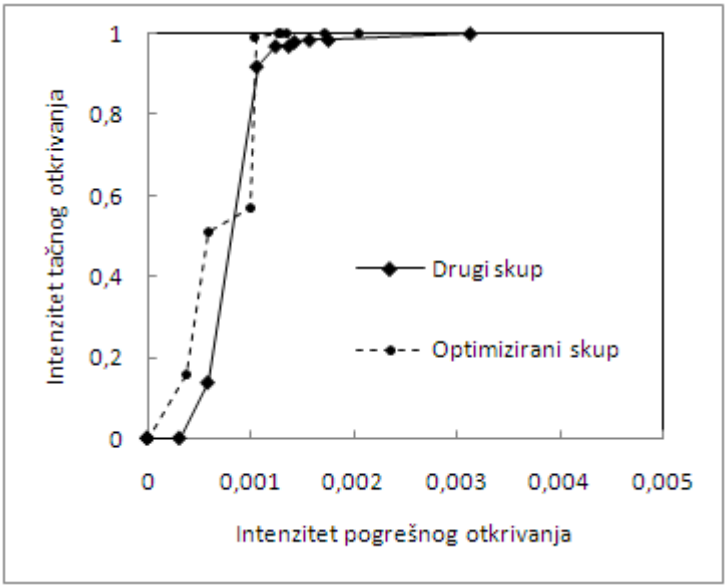
Slika 23. Iznosi odstupanja paketa 17 testnih napada za drugi skup separatora

Iznosi odstupanja za skoro sve pakete iz svih napada su manji nego iznosi sa optimiziranim skupom separatora. Međutim ovi iznosi su i dalje mnogo veći od iznosa za normalne pakete. Iznosi odstupanja za posljednje pakete u napadima 1, 2, 3, pa i 16 su prilično mali. Razlog za ovo je što su ti paketi kao posljednji u napadu relativno kratki pa se desilo da je sa ovim skupom separatora za ove pakete došlo do određenog poklapanja između dobivenih riječi i prelaza u paketu i skupa naučenih normalnih riječi i prelaza.

Ovo je situacija koja se može očekivati i ne može se spriječiti. Jedan od razloga izbora prikaza rezultata ovog skupa separatora u radu jeste da se ukaže na ovakvu mogućnost dobivanja nižeg iznosa odstupanja za završne pakete napada koji mogu biti kratki pri slučajnom izboru skupa separatora. Niži iznos odstupanja za pojedine pakete iz napada ne bi trebalo da predstavlja smetnju detekciji, ukoliko ostali paketi iz napada imaju velike iznose odstupanja. Izuzimajući ovih nekoliko paketa najniži iznos odstupanja za bilo koji drugi napadački paket je 1,15. Najniži iznos odstupanja za paket sa najvišim iznosom odstupanja u bilo kom od napada je 1,17. Ako se granica normalnosti postaviti na ovaj iznos moguće je otkriti sve testirane napade i gotovo sve njihove pakete. Kako su dosada testirani normalni paketi sa ovim skupom separatora imali male iznose odstupanja koji nikad nisu prešli

vrijednost 0,2 i u ovom slučaju da postoji velika razlika u iznosu odstupanja između normalnih i nenormalnih paketa. Ova činjenica, kao i predstavljeni rezultati testiranja sa pravim napadima, podržavaju iznesenu pretpostavku da primjena slučajnog skupa separatora nije bitno negativno uticala na mogućnost detekcije pokušaja upada, drugim riječima optimizacija skupa znakova ne utiče znatno na sposobnost detekcije.

Za završnu provjeru ukupne uspješnosti sistema sa ovim skupom separatora konstruisana je ROC kriva koje je prikazana na slici 24.



Slika 24. ROC kriva za drugi skup separatora

Na slici 24. prikazana je isprekidanom linijom i ROC kriva za optimizirani skup separatora radi poređenja. Obje ROC krive dobivene su na isti način, pomjeranjem granica normalnosti od 0,2 do 2 sa korakom 0,2, i sa istim skupom podataka. I na ovom grafu skala intenziteta pogrešnog otkrivanja je samo od 0 do 0,005 radi bolje preglednosti. ROC kriva za sistem sa slučajnim skupom separatora bliska je sa onoj za optimizirani skup. Činjenica je u da većini radnih tačaka optimizirani skup separatora ima nešto bolji odnos intenziteta tačnog i pogrešnog otkrivanja. Međutim, za prag normalnosti od

1,6 sistem sa novim skupom separatora čak ima bolju vrijednost ovog odnosa. Radi numeričkog poređenja kompletnih ROC krivih upoređeni su AUC (iznosi površine ispod krivih). AUC za ROC krivu sa novim skupom separatora manji je samo za 0,014% od onog za ROC krivu za optimizirani skup separatora.

Važno je ponoviti da je ROC kriva pravljena na nivou pojedinačnih paketa, a ne napada. Pošto je za otkrivanje pokušaja upada uglavnom dovoljno otkriti jedan od paketa napada ROC kriva na nivou napada bi bila još bolja, jer na nju ne bi negativno uticali neki napadački paketi koji su sa ovim skupom separatora imali mali iznos odstupanja od modela normalnog ponašanja i negativno uticali na intenzitet tačnog otkrivanja. Ono što je bitno je mogućnost korištenja sistema sa različitim skupom separatora. Zamjena optimiziranog skupa separatora sa slučajno generisanim skupom uz određena ograničenja nije bitno uticala na sposobnost sistema da razlikuje normalni od zloćudnog saobraćaja.

Kao još jedan dokaz uspješnosti u tabeli VII na drugi način je uporedno prikazana uspješnost otkrivanja napada za optimizirani i sistem sa novim skupom separatora. Umjesto procenta intenziteta pogrešnog otkrivanja dat je prosječan broj lažnih uzbuna dnevno što je za praktičan rad sistema i zahtjeve koje postavlja na operatora često važnija i konkretnija informacija. Ovaj broj neznatno je veći za novi skup separatora uz gotovo istu uspješnost otkrivanja. Ovakvo povećanje prosječnog broja lažnih uzbuna dnevno ne bi trebalo predstavljati posebnu smetnju za praktičnu upotrebi sistema.

Kako su oba sistema testirana sa istim normalnim saobraćajem i napadima dobiveni rezultati su uporedivi. Ovi rezultati potvrđuju mogućnost korištenja predloženog sistema za otkrivanje upada sa slučajno odabranim skupom od 20 separatora kao ključem.

Tabela VII. Odnos procenta otkrivenih napada i broja lažnih uzbuna dnevno u zavisnosti od praga normalnosti za drugi i optimizirani skup separatora

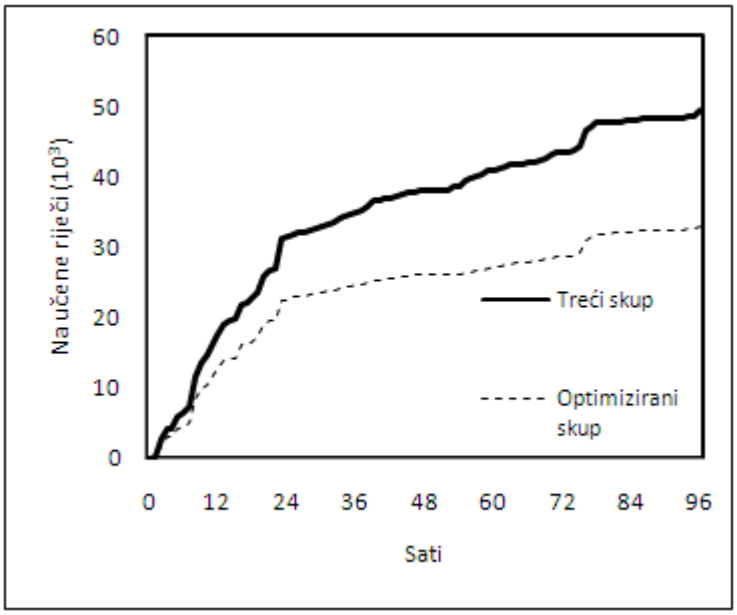
Prag normalnosti	Otkrivenih napadačkih paketa (%)	Lažnih uzbuna dnevno	Otkrivenih napadačkih paketa (%) (optimizirani)	Lažnih uzbuna dnevno (optimizirani)
	od **197** ukupno	od **9120** paketa	od **197** ukupno	od **9120** paketa
0,2	100%	29	100%	19
0,4	100%	16	100%	16
0,6	98%	14	100%	12
0,8	98%	13	100%	12
1	98%	13	100%	12
1,2	97%	11	100%	12
1,4	97%	10	99%	10
1,6	92%	5	57%	9
1,8	14%	3	51%	5
2	0%	0	16%	0

4.4.2 Test sa 15 separatora

Naredni test čiji će detaljni rezultati biti prikazane rađen je sa skupom od 15 separatora. I ovaj put je na isti način generisano 15 slučajnih brojeva između 9 i 127. Ovi brojevi uzeti su kao ASCII vrijednosti separatora u testnom skupu. Konkretne ASCII vrijednosti 15 elemenata skupa bile su:

9, 20, 34, 36, 53, 60, 63, 64, 66, 69, 71, 97, 103, 111, 116

Sa ovim skupom separatora izvršeno je učenje riječi iz istog normalnog saobraćaja kao i za prethodne skupove znakova. Slika 25. prikazuje broj naučenih riječi u zavisnosti od broja sati normalnog saobraćaja korištenog za učenje za ovaj i za optimizirani skup (isprekidana linija).



Slika 25. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za treći skup separatora

Ponovo je vidljiva ista tendencija da se broj naučenih riječi stabilizira nakon 96 sati učenja što potvrđuje pretpostavku od o dovoljnom periodu učenja. Ukupan broj riječi je gotovo 50 000, što je relativno malo povećanje od oko 7% u odnosu na prethodni skup od 20 separatora. Ovo povećanje je rezultat manjeg broja separatora, čime se dobiva više mogućih kombinacija dužih riječi. Ni ovo povećanje se ne može smatrati ogromnim i preprekom za korištenje. Povećanje nema znatnijeg uticaja na veličinu *hash* tabele i performanse sistema.

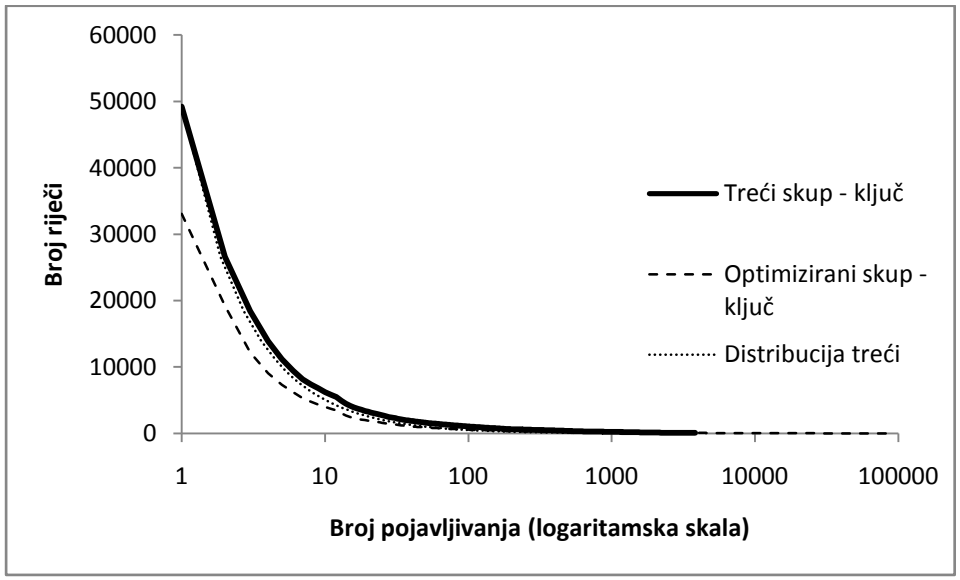
Broj različitih riječi kao funkcija pojavljivanja dat je na slici 26. Radi provjere uklapanja eksperimentalno dobivenih vrijednosti sa pretpostavljenom Zipf-

Mandelbrot distribucijom izračunati su parametri distribucije (5) i (6). Pri ovome su korišteni slijedeći podaci.

S = 15 – broj separatora

M = 256 – 15 = 241 - broj znakova za pravljenje riječi

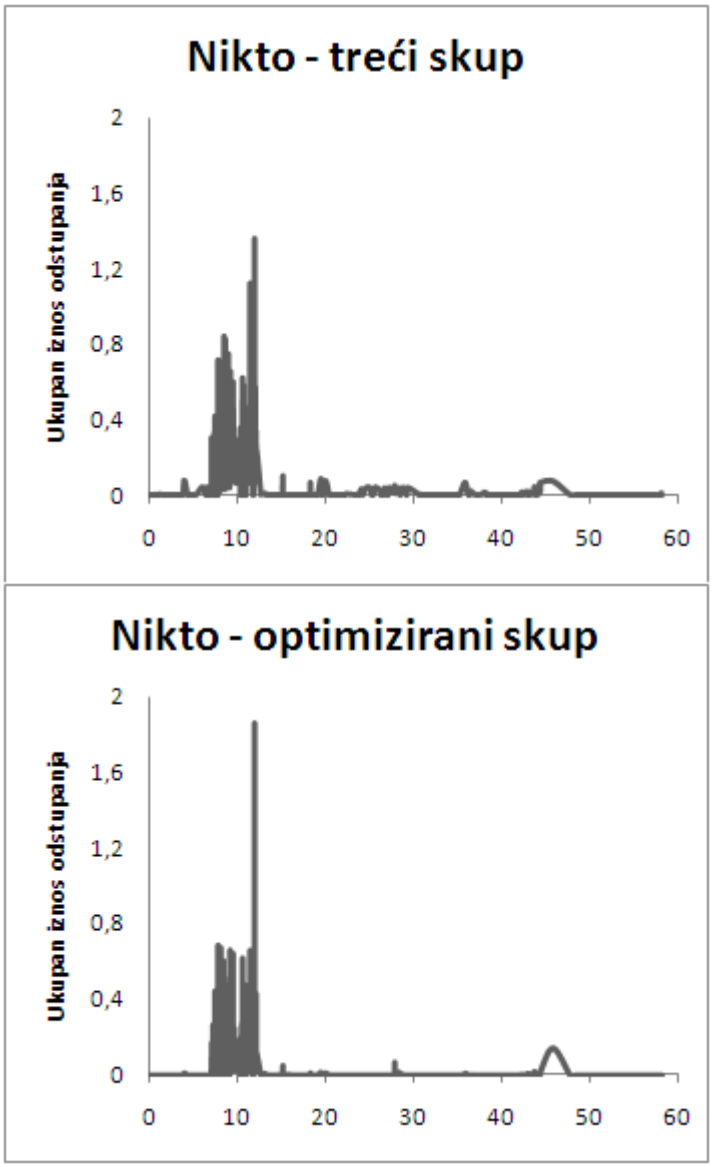
Izračunate vrijednosti parametara su α = 1, 01101 i B = 1,00417. Na osnovu uvrštavanja ovih vrijednosti u formulu (4) i skaliranja prema broju riječi nacrtana je i očekivana distribucija na istoj slici. Ponovo je dobiveno odlično poklapanje između eksperimentalne krive i analitičke distribucije. Znači da su ponovo zadovoljeni uslovi za korištenje evidentiranja manjeg broja prelaza. Nakon dodatnih testiranja sa još različitih skupova separatora provjeriće se da li je kriva ovakva za sve skupove.



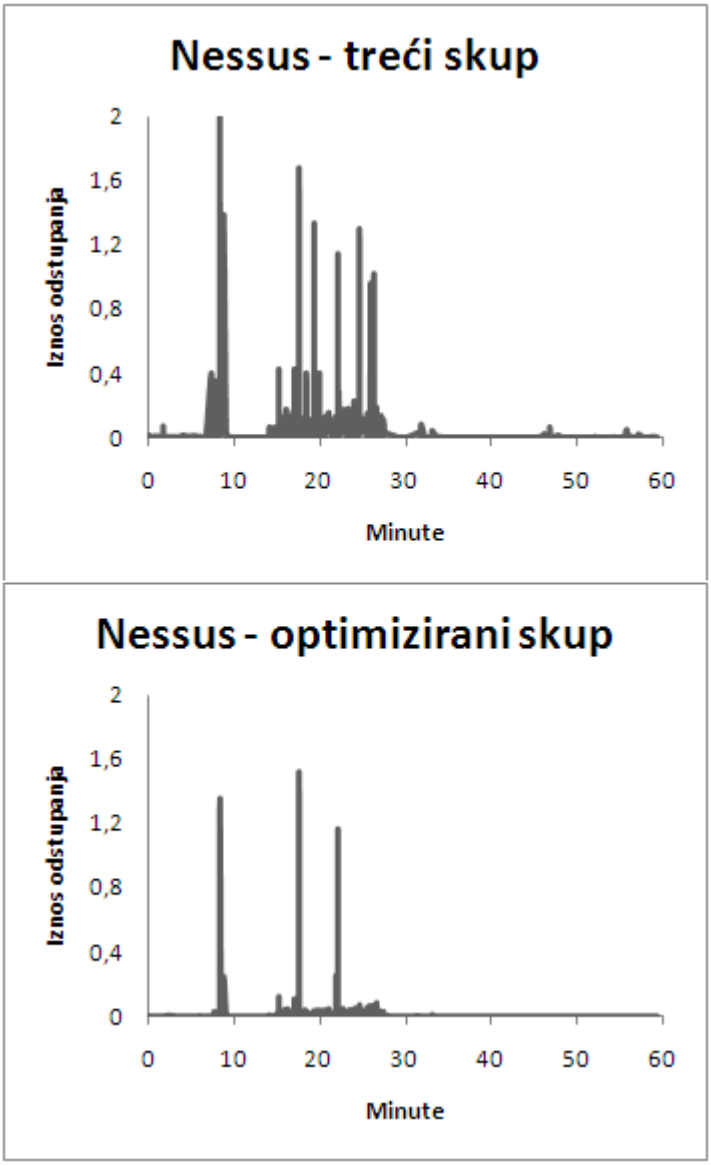
Slika 26. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za treći skup separatora (logaritamska skala)

Nakon što je sistem sa ovim skupom separatora prošao proces učenja normalnih riječi i prelaza među njima pristupilo se provjeri sposobnosti detekcije.

Rezultati iznosa odstupanja za sat sa Nikto saobraćajem dati su na slici 27. Rezultati iznosa odstupanja za sat sa Nessus saobraćajem dati su na slici 28. I za ovaj skup separatora iznosi odstupanja su znatno veći za pregledne Nikto i Nessus pakete nego za normalne pakete. Ni ova promjena skupa separatora nije negativno uticala na mogućnost razlikovanja običnih i neobičnih, preglednih, paketa.

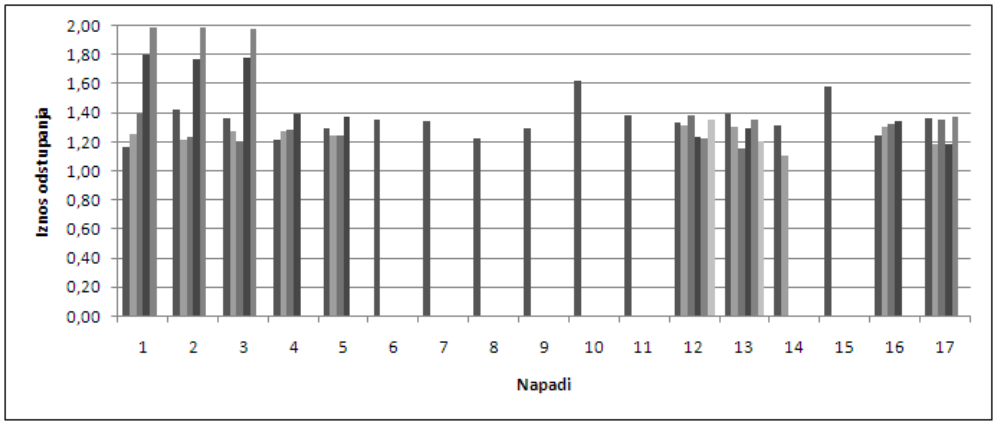


Slika 27. Iznosi odstupanja za sat saobraćaja koji uključuje Nikto pregled za treći i optimizirani skup separatora



Slika 28. Iznosi odstupanja za sat saobraćaja koji uključuje Nessus pregled za treći i optimizirani skup separatora

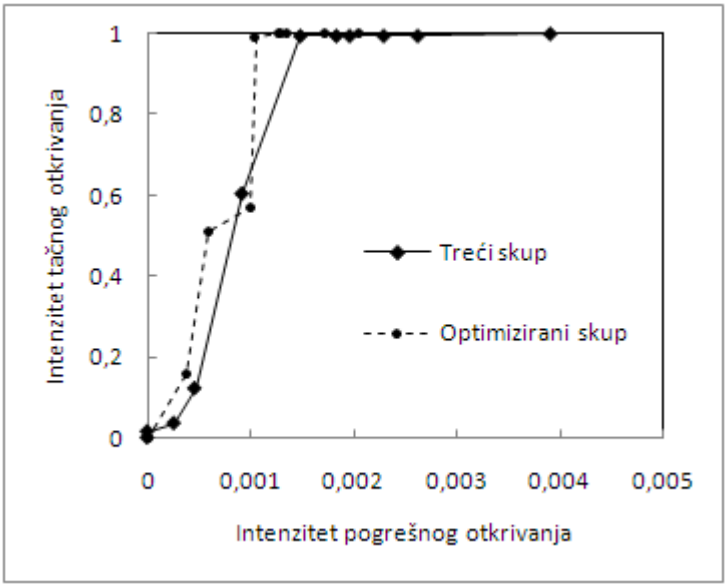
Rezultati testa uspješnosti otkrivanja pravih napada iz tabele IV prikazani su na slici 29.



Slika 29. Iznosi odstupanja paketa 17 testnih napada za treći skup separatora

Iznosi odstupanja su u prosjeku nešto manji nego sa optimiziranim skupom separatora, ali nešto veći nego sa drugim testiranim skupom od 20 separatora. Svi iznosi su i dalje mnogo veći od iznosa za normalne pakete. Ovaj put nema pojave malih iznosa odstupanja za posljednje pakete napada kao za drugi skup separatora. Kako je rečeno, ta pojava je posljedica slučajnog izbora separatora i nema ozbiljniji negativni efekat na sposobnost otkrivanja pokušaja upada. Najniži iznos odstupanja za bilo koji paket je 1,1. Ako se ovo postavi kao granica između normalnih i nenormalnih paketa svaki pojedinačni paket svih testiranih napada moguće je otkriti. Najniži iznos odstupanja za paket sa najvišim iznosom odstupanja u bilo kom od napada je 1,22. Ako se granica normalnosti postaviti na ovaj iznos moguće je otkriti sve testirane napade. Ovi rezultati potvrđuju da se promjenom skupa separatora nisu bitno smanjili iznosi odstupanja napadačkih paketa od modela normalnog ponašanja, odnosno da se nije umanjila sposobnost detekcije pokušaja upada.

Ukupna uspješnost predloženog sistema za otkrivanje upada sa ovim skupom od 15 separatora prikazana je putem ROC krive na slici 30.



Slika 30. ROC kriva za treći skup separatora

Na slici 30. prikazana je isprekidanom linijom i ROC kriva za optimizirani skup separatora radi poređenja. Obje ROC krive dobivene su na isti način, pomjeranjem granica normalnosti od 0,2 do 2 sa korakom 0,2, i sa istim skupom podataka. ROC kriva i za sistem sa skupom od 15 separatora slična je onoj za optimizirani skup. Za pragove normalnosti od 1,4 i 1,6 sistem sa skupom od 15 separatora čak ima bolji odnos intenziteta tačnog i pogrešnog otkrivanja od onog sa optimiziranim skupom. Radi numeričkog poređenja kompletnih ROC krivih upoređeni su AUC. AUC za ROC krivu sa skupom od 15 separatora manji je samo za 0,017% od onog za ROC krivu za optimizirani skup separatora. Zamjena optimiziranog skupa separatora sa slučajno generisanim skupom uz određene ograničenja minimalno je uticala na sposobnost sistema da razlikuje normalni od zloćudnog saobraćaja.

Tabela VIII na drugi način uporedno prikazuje uspješnost otkrivanja napada za optimizirani i sistem sa skupom od 15 separatora. Procenat uspješno otkrivenih napadačkih paketa je gotovo identičan za oba skupa, uz nešto manji broj lažnih uzbuna za optimizirani skup. Ova razlika je dovoljno mala da ne predstavlja smetnju praktičnoj upotrebi sistema.

Tabela VIII. Odnos procenta otkrivenih napada i broja lažnih uzbuna dnevno u zavisnosti od praga normalnosti za treći i optimizirani skup separatora

Prag normalnosti	Otkrivenih napadačkih paketa (%)	Lažnih uzbuna dnevno	Otkrivenih napadačkih paketa (%) (optimizirani)	Lažnih uzbuna dnevno (optimizirani)
	od **197** ukupno	od **9120** paketa	od **197** ukupno	od **9120** paketa
0,2	100%	36	100%	19
0,4	100%	24	100%	16
0,6	99%	21	100%	12
0,8	99%	18	100%	12
1	99%	17	100%	12
1,2	99%	14	100%	12
1,4	99%	8	99%	10
1,6	60%	4	57%	9
1,8	12%	2	51%	5
2	4%	0	16%	0

I testovi sa ovim skupom od 15 separatora potvrđuju mogućnost korištenja predloženog sistema za otkrivanje upada sa ovim, različitim, slučajno generisanim, skupom separatora kao ključem.

4.4.3 Zbirni rezultati testova sa različitim skupovima separatora

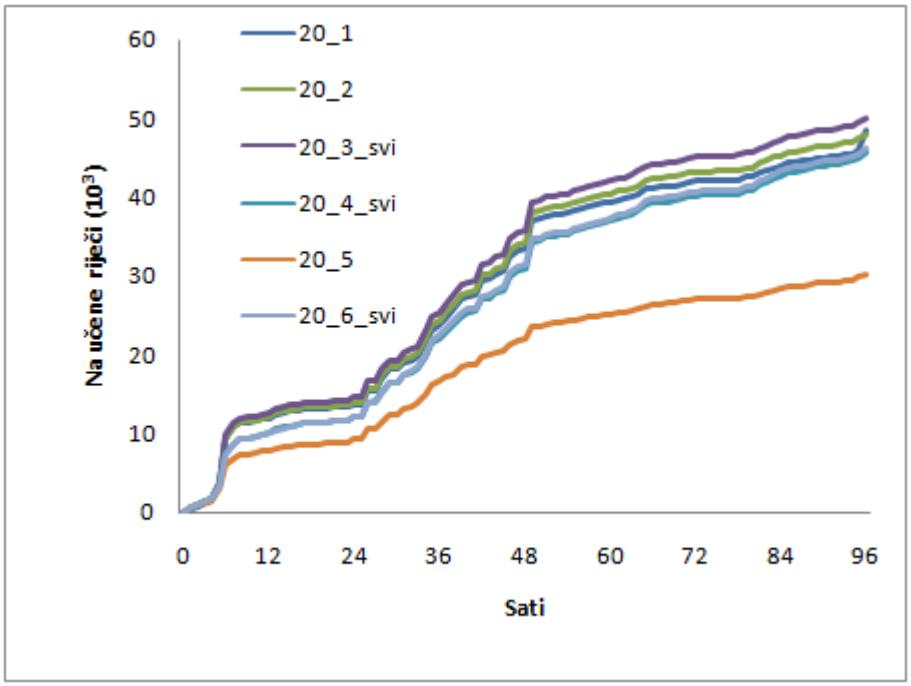
Detaljni testovi dva slučajno generisana skupa separatora potvrdili su da se ovim promjenama nije umanjila sposobnost predloženog sistema za otkrivanje upada.

Da bi se dalje potvrdila teza da skup separatora ne utiče na sposobnost otkrivanja upada napravljeni su dodatni testovi sa većim brojem slučajno odabranih skupova separatora. Korišteni skupovi separatora, način njihovog generisanja i rezultati testiranja biće dati u nastavku. Radi preglednosti, rezultati testova grupisani su po broju znakova.

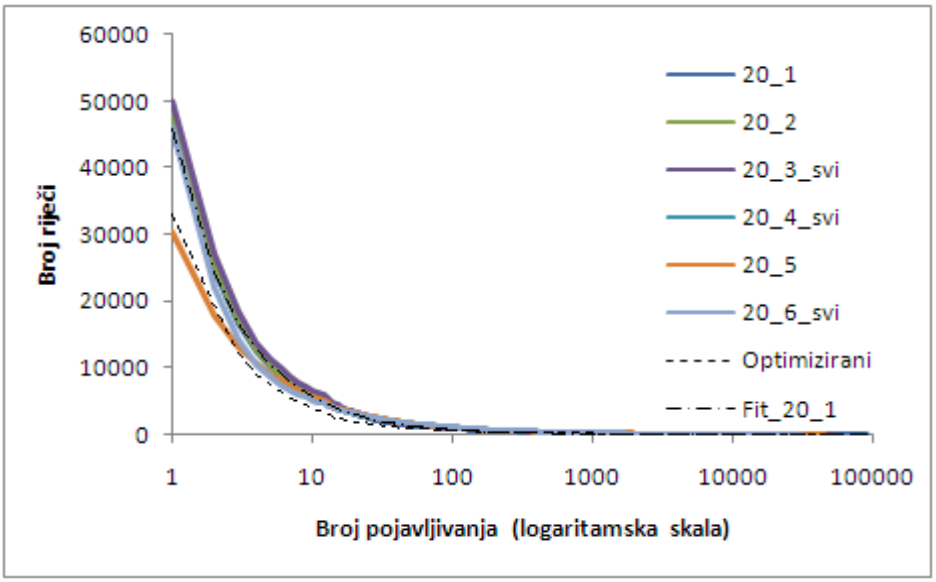
4.4.3.1 Testovi sa 20 separatora

Prva grupa testova napravljena je sa šest skupova separatora od po 20 elemenata. Tri skupa (prvi, drugi i peti) su imala slučajno generisane elemente sa ASCII vrijednostima od 9 do 127. Preostala tri skupa (treći, četvrti i šesti) su imale slučajno generisane elemente sa ASCII vrijednostima u punom opsegu od 0 do 255.

Slika 31. prikazuje porast broja naučenih riječi kao funkciju broja sati učenja za svih šest skupova separatora od 20 elemenata i za optimizirani skup. Za sve testne skupove vidljiva je ista tendencija da se broj naučenih riječi stabilizuje, odnosno da je vrijeme od 96 sati učenja dovoljno. Svi skupovi osim jednog (petog) rezultiraju sličnim brojem naučenih riječi koji je oko 40% veći od broja riječi za optimizirani skup. Broj naučenih riječi za peti skup bio je čak manji nego za optimizirani skup. Očigledno da je ovom slučaju slučajni izbor separatora bio pogodan za podjelu sadržaja HTTP paketa na manji broj različitih riječi.

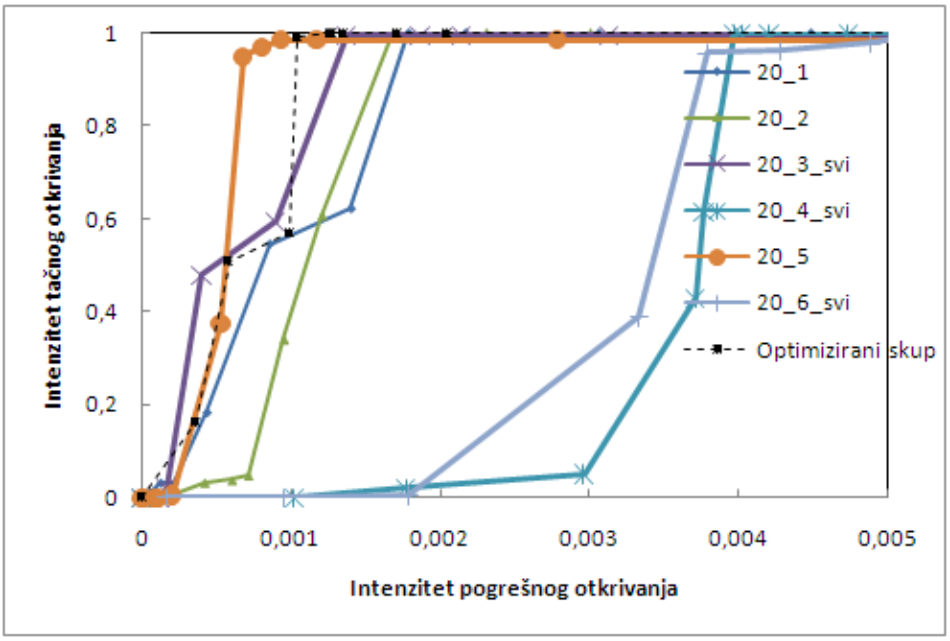


Slika 31. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 20 elemenata



Slika 32. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 20 elemenata (logaritamska skala)

Na slici 32. je dijagram koji pokazuje kako broj riječi koje se pojavljuju veći broj puta brzo opada za svih šest skupova. Za sve skupove ova kriva i kriva distribucije definisana formulom (4) se dobro poklapaju. Fit za prvi skup nacrtan je radi ilustracije. Očigledno je da je ova zavisnost pravilo. Oblik ove krive ne zavisi od skupa separatora. Zaključak je bitan jer potvrđuje mogućnost korištenja manjeg skupa riječi za pohranjivanje prelaza.



Slika 33. ROC kriva za skupove separatora od 20 elemenata

Slika 33. prikazuje ROC krive za sve skupove od 20 znakova. Četiri krive su bliske onoj za optimizirani skup (prikazana isprekidano). Dvije krive, za četvrti i šesti skup, imaju nešto veći iznos intenziteta pogrešnog otkrivanja u pet tačaka krive. Dva skupa kojima odgovaraju ove krive su sa ASCII vrijednostima separatora u rasponu od 0 do 255. Potrebno je naglasiti da su i ove nešto lošije krive još uvijek dobre. One su uporedive ili bolje od ROC krivih sličnih sistema iz ranije pomenutih radova.

Radi numeričkog poređenja kompletnih ROC krivih upoređeni su AUC svih krivih sa AUC za ROC krivu optimiziranog skupa. Procenti za koji su ovi AUC manji od onog za ROC krivu za optimizirani skup separatora dati su u tabeli IX:

Tabela IX. Razlika između AUC ROC krivih za skupove separatora od 20 elemenata i AUC ROC krive za optimizirani skup separatora

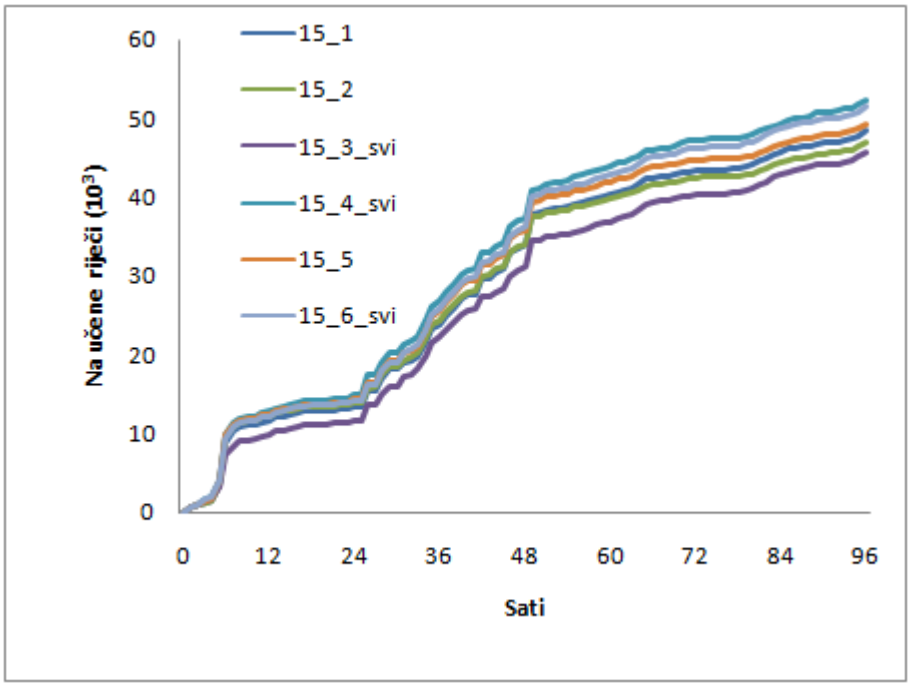
20_1	20_2	20_3_svi	20_4_svi	20_5	20_6_svi
-0,030%	-0,044%	-0,252%	-0,287%	-0,743%	-0,508%

Ovi rezultati izgledaju nešto drugačije nego što je utisak sa dijagrama krivih. Kako je ranije rečeno sistem sa većim AUC ne mora u praktičnoj upotrebi biti bolji. Bolji sistem je onaj čija je izabrana radna tačka bliža idealnoj tačci (0,1). Bitan rezultat iz ove tabele je da su razlike između krivih i po ovom kriteriju male.

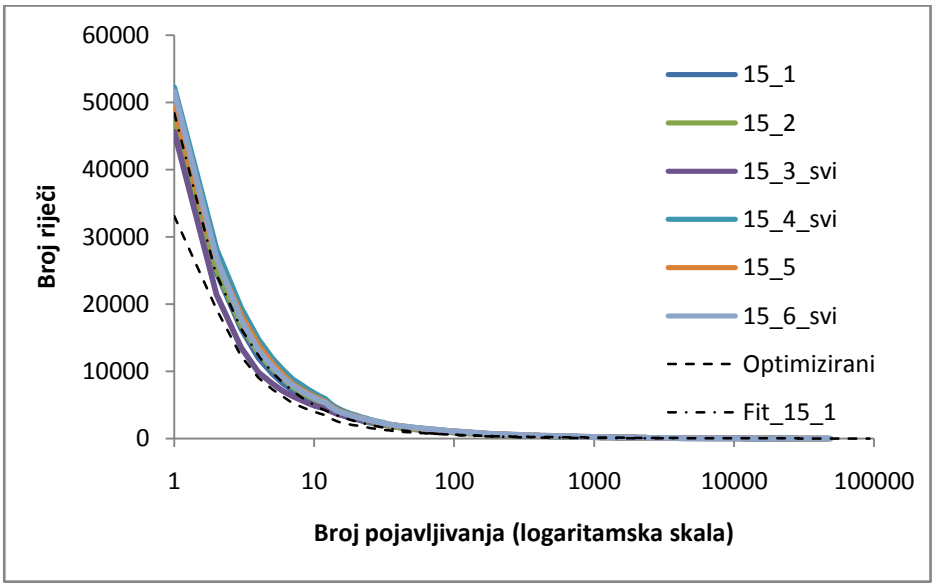
Rezultati testiranja potvrđuju mogućnost korištenja skupa od slučajno generisanih 20 separatora za generisanje skupa riječi i prelaza među njima na kojima se zasniva model normalnog ponašanje. Zapravo rezultati potvrđuju mogućnost korištenja ovog modela za uspješnu detekciju pokušaja upada. Na osnovu optimiziranog skup znakova dobiva se manji model. U ostalim pogledima može se smatrati da izbor elemenata skupa od 20 separatora ne utiče na mogućnost otkrivanja upada. Skup od 20 separatora može se koristiti kao ključ u kriptografiji.

4.4.3.2 Testovi sa 15 separatora

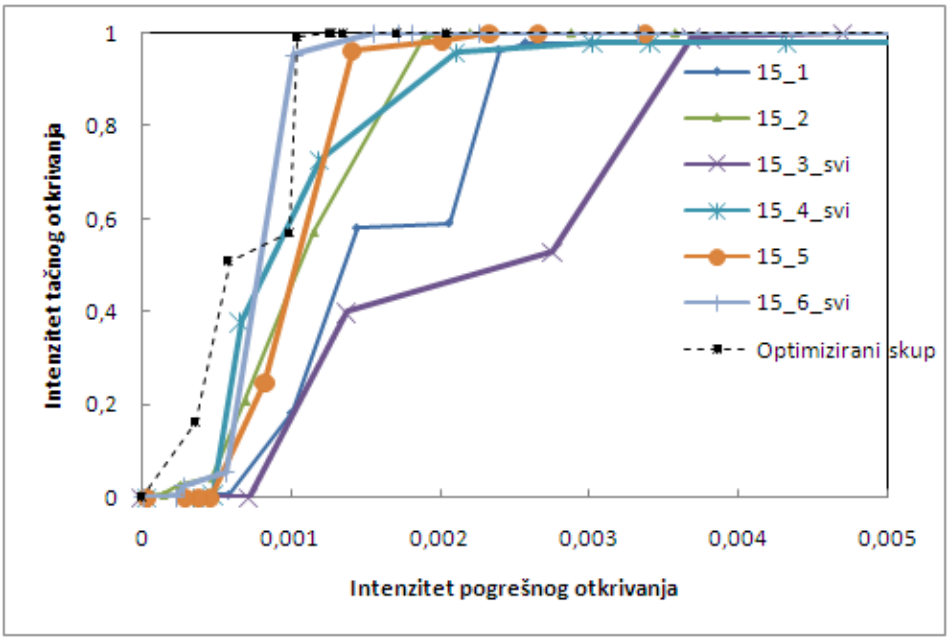
Naredna grupa testova rađena je sa skupovima od 15 separatora. Ponovo je napravljeno šest skupova. Tri skupa imala su slučajno generisane ASCII vrijednosti od 9 do 127, a druga tri u rasponu od 0 do 255. Rezultati testiranja prikazani su na slikama 34., 35. i 36. i u tabeli X.



Slika 34. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 15 elemenata



Slika 35. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 15 elemenata (logaritamska skala)



Slika 36. ROC kriva za skupove separatora od 15 elemenata

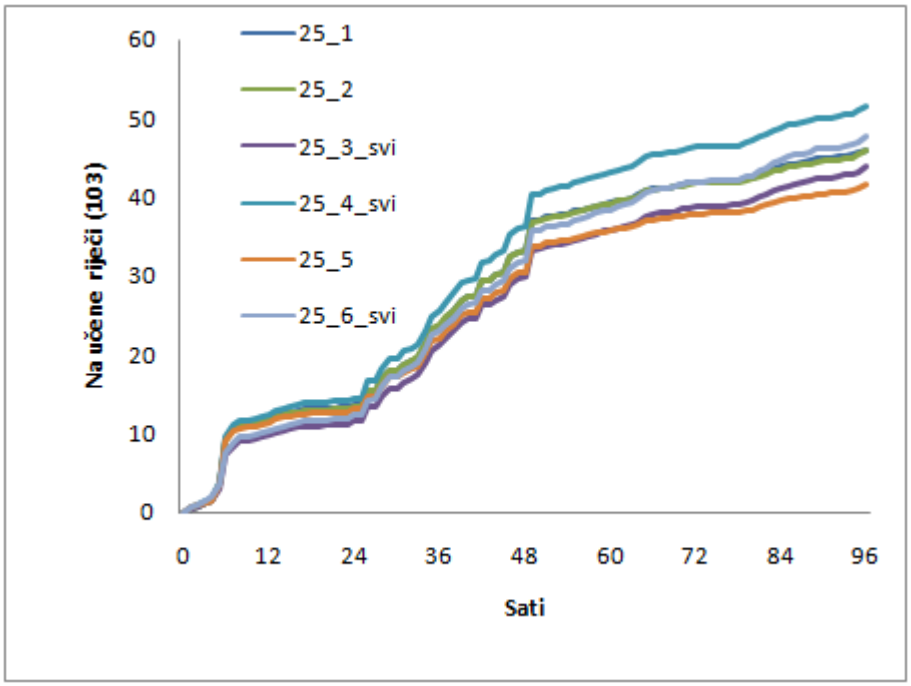
Tabela X. Razlika između AUC ROC krivih za skupove separatora od 15 elemenata i AUC ROC krive za optimizirani skup separatora

15_1	15_2	15_3_svi	15_4_svi	15_5	15_6_svi
-1,100%	-0,042%	-0,153%	-1,063%	-0,035%	-0,012%

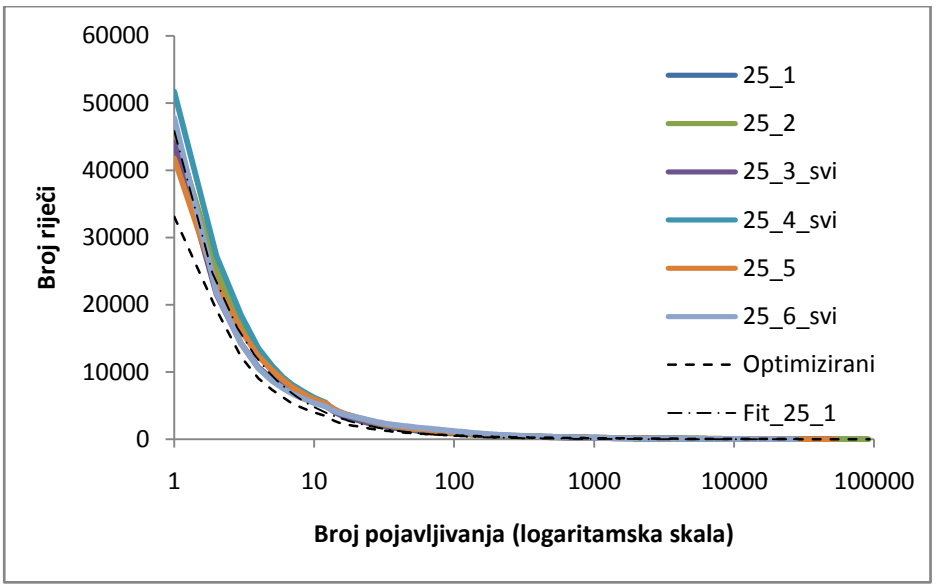
Broj naučenih riječi i njihova zavisnost od broja sati učenja slični su onoj za 20 znakova, s tim što se ovdje rezultati ni za jedan skup ne razlikuju bitno. Slično je i sa brojem pojavljivanja riječi. Zavisnosti su po vrijednosti i obliku bliske onim iz prethodnih testova. ROC krive su neznatno lošije od one za optimizirani skup, slično je i sa AUC ovih krivih. I ovaj test potvrđuje da sposobnost otkrivanja upada ne zavisi od izbora elemenata skupa od 15 separatora.

4.4.3.3 Testovi sa 25 separatora

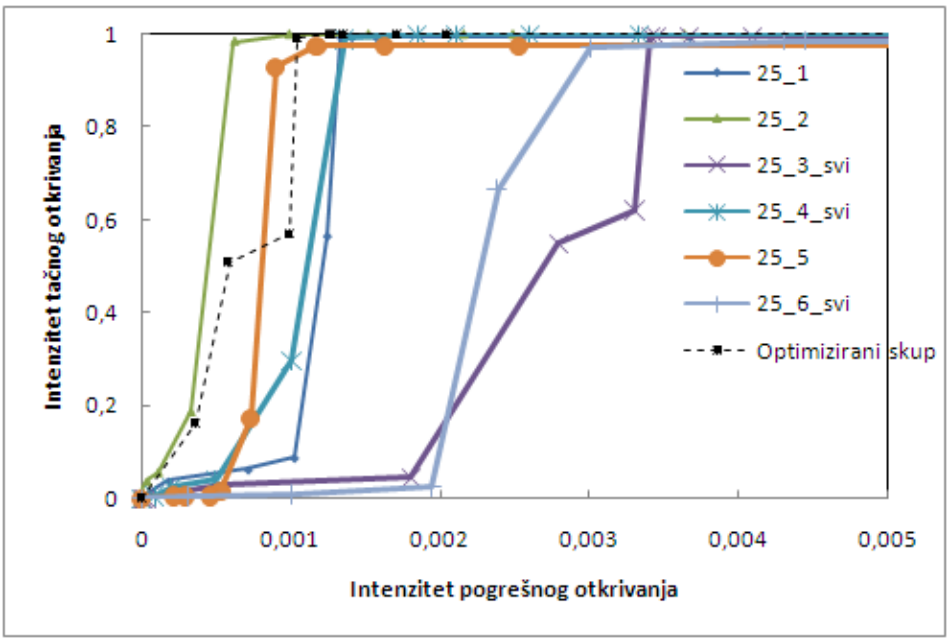
Naredna grupa testova rađena je sa skupovima od 25 separatora. I za ovaj test napravljeno je šest skupova. Ponovo su tri skupa imala su slučajno generisane ASCII vrijednosti od 9 do 127, a druga tri u rasponu od 0 do 255. Rezultati testiranja prikazani su na slikama 37., 38. i 39. i u tabeli XI.



Slika 37. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 25 elemenata



Slika 38. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 25 elemenata (logaritamska skala)



Slika 39. ROC kriva za skupove separatora od 25 elemenata

Tabela XI. Razlika između AUC ROC krivih za skupove separatora od 25 elemenata i AUC ROC krive za optimizirani skup separatora

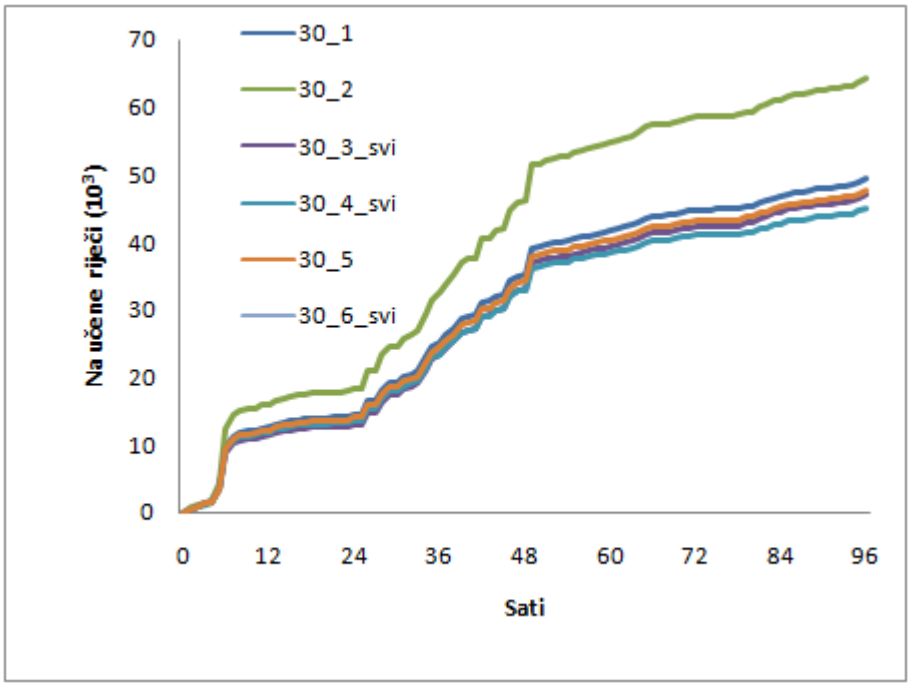
25_1	25_2	25_3_svi	25_4_svi	25_5	25_6_svi
-0,299%	0,026%	-0,452%	-0,036%	-1,279%	-0,934%

Porast broja riječi sa satima učenja, kao i broj pojavljivanja riječi slični su sa onim za prethodno testirane skupove. ROC krive su slične ROC krivim za skupove od 20 znakova. Četiri krive su bliske onoj za optimizirani skup, dok su dvije nešto lošije. Potrebno je ponovo napomenuti da su i ove lošije krive dovoljno dobre za otkrivanje upada. Zaključak je da slučajni izbor 25 separatora ne utiče negativno na sposobnost sistema da otkrije upade.

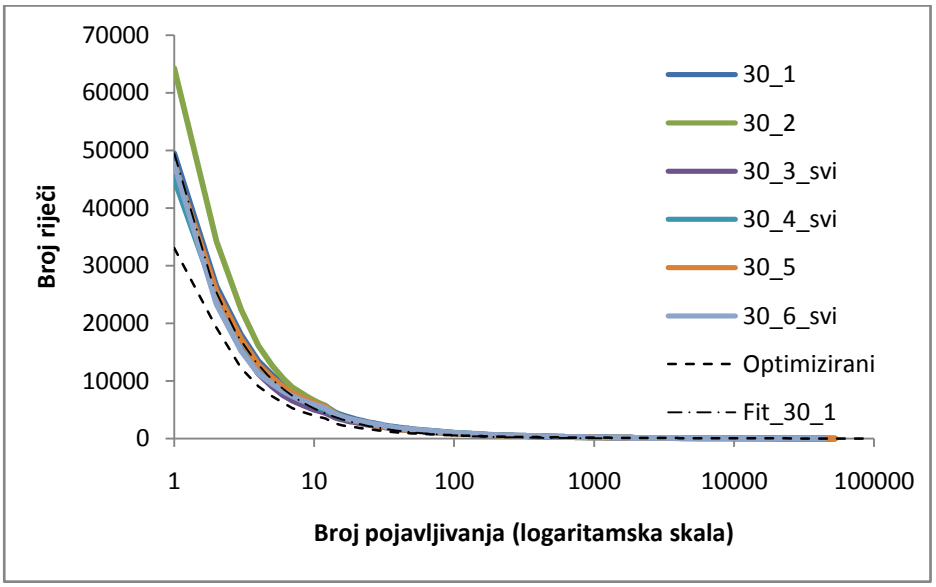
4.4.3.4 Testovi sa 30 separatora

Slijedeća grupa skupova separatora imala je po 30 elemenata. Testovi su napravljeni sa šest različitih slučajno generisanih skupova, od kojih su tri bili u opsegu od 9 do 127, a tri od 0 do 255. Krive dobivene testiranjem date su na slikama 40., 41. i 42.

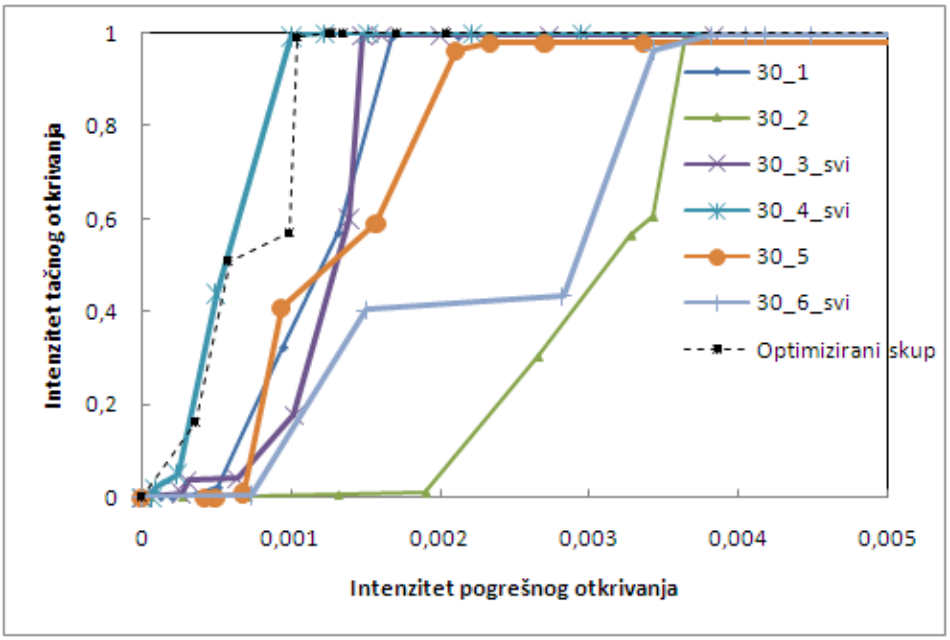
Prve dvije krive su slične onim za ranije testirane skupove. Jedan od skupova proizveo je za oko 30% veći broj riječi od ostalih. ROC krive su u prosjeku nešto lošije nego za manje skupove, ali još uvijek praktično upotrebljive. Utisak je da bi dalje povećavanje broja znakova moglo dovesti do pogoršanja sposobnosti otkrivanja upada. Naredni testovi će upravo provjeriti granične vrijednosti broja separatora.



Slika 40. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 30 elemenata



Slika 41. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 30 elemenata (logaritamska skala)



Slika 42. ROC kriva za skupove separatora od 30 elemenata

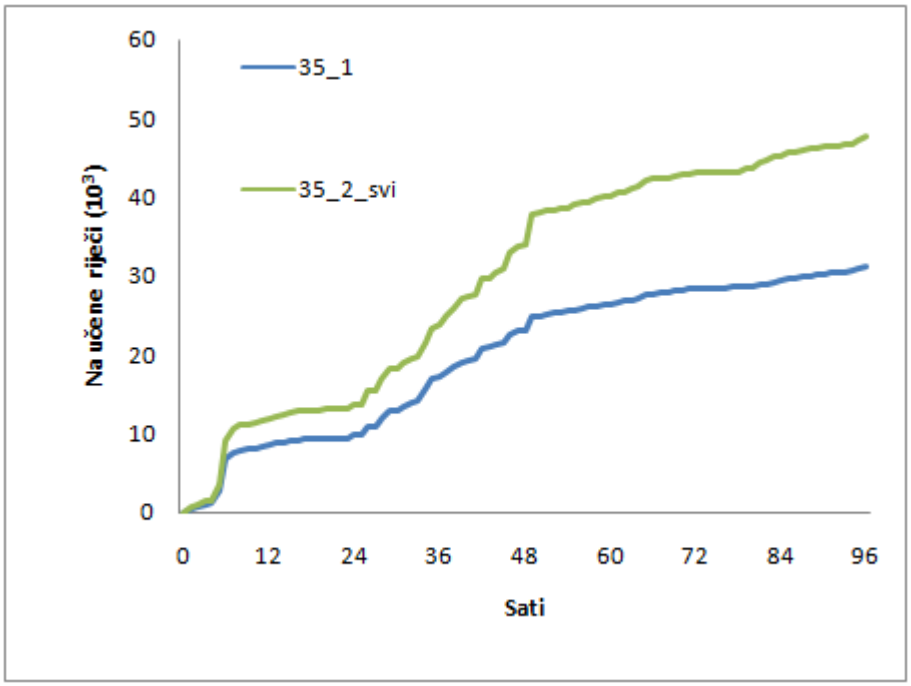
Tabela XII. Razlika između AUC ROC krivih za skupove separatora od 30 elemenata i AUC ROC krive za optimizirani skup separatora

30_1	30_2	30_3_svi	30_4_svi	30_5	30_6_svi
-0,302%	-0,485%	-0,306%	0,011%	-1,076%	-0,416%

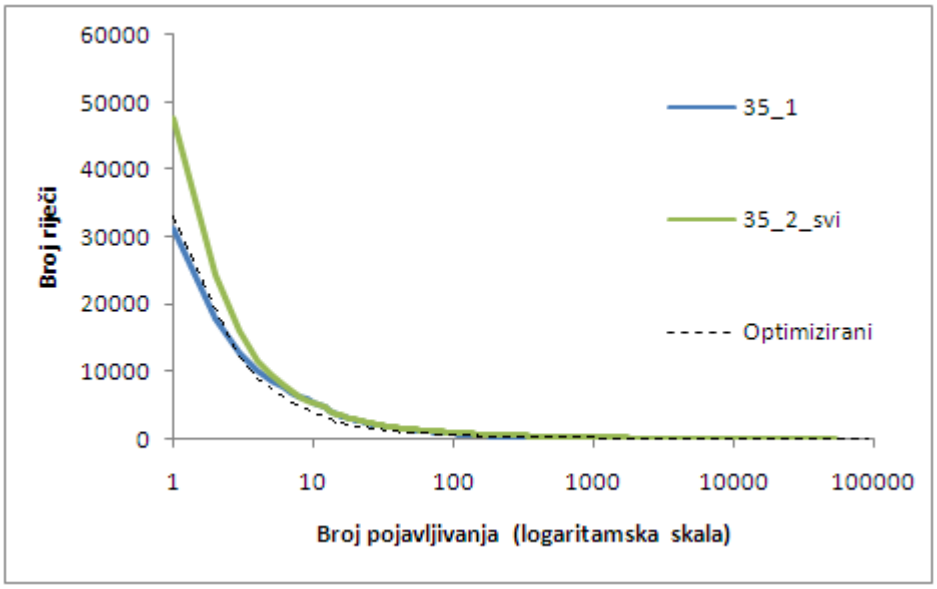
4.4.3.5 Ostali testovi

Povećavanjem broja separatora na 35 pojavila se poteškoća pri generisanju slučajnih brojeva. Bio je potreban veliki broj pokušaja da bi se generisalo 35 različitih brojeva od 9 do 127. Kako je skup od 30 znakova pokazao blagu tendenciju pogoršanja ROC krivih, smatralo se da bi ovo mogla biti praktična granica na broj separatora. Napravljena su samo dva testa sa 35 znakova. Prvi u opsegu od 9 do 127, a drugi od 0 do 255.

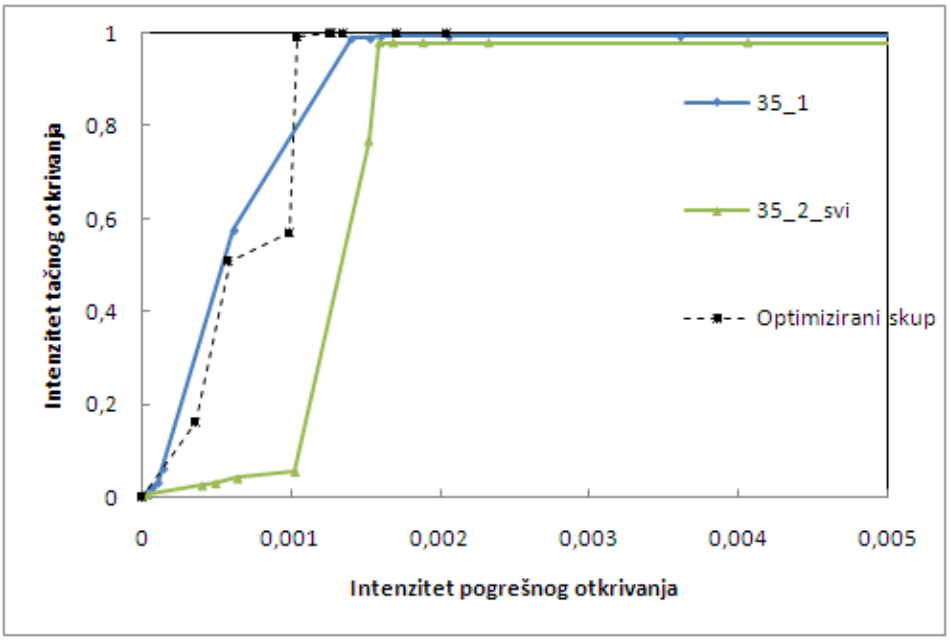
Rezultati ovih testova dati su na slikama 43., 44. i 45. i u tabeli XIII. Broj riječi za prvi skup bio je sličan onom za optimizirani. Ovo se može objasniti činjenicom da se zbog većeg broja separatora smanjuje prosječna dužina riječi, pa time i broj mogućih kombinacija. Kriva broja pojavljivanja riječi i dalje ima isti oblik. ROC su neznatno lošije od one za optimizirani skup. Povećavanje broja znakova na 35 nije umanjilo sposobnost sistema, ali zbog poteškoća sa generacijom različitih znakova odustalo se od daljeg povećavanja.



Slika 43. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 35 elemenata



Slika 44. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 35 elemenata (logaritamska skala)



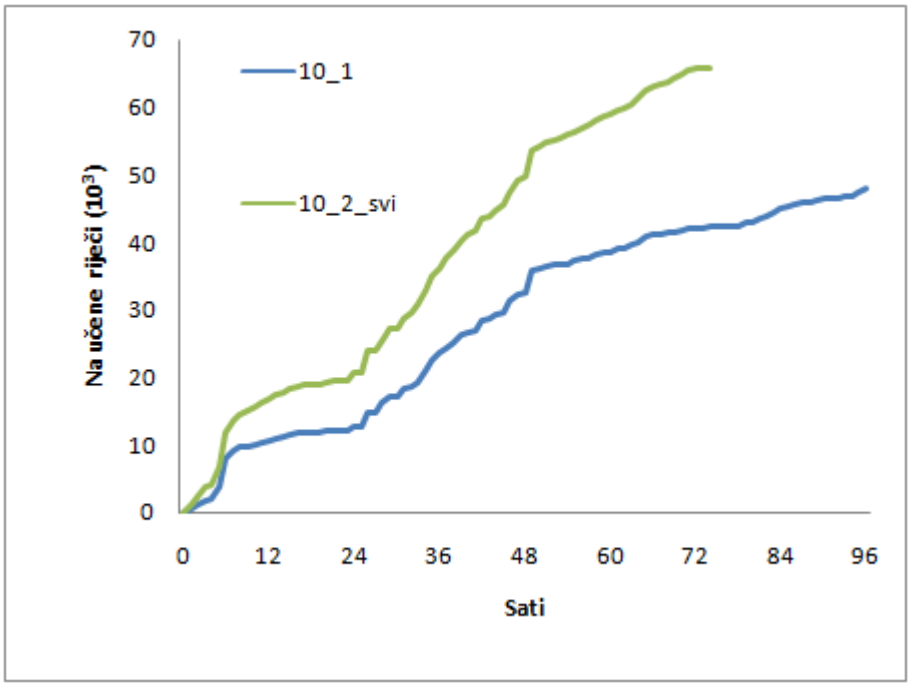
Slika 45. ROC kriva za skupove separatora od 35 elemenata

Tabela XIII. Razlika između AUC ROC krivih za skupove separatora od 35 elemenata i AUC ROC krive za optimizirani skup separatora

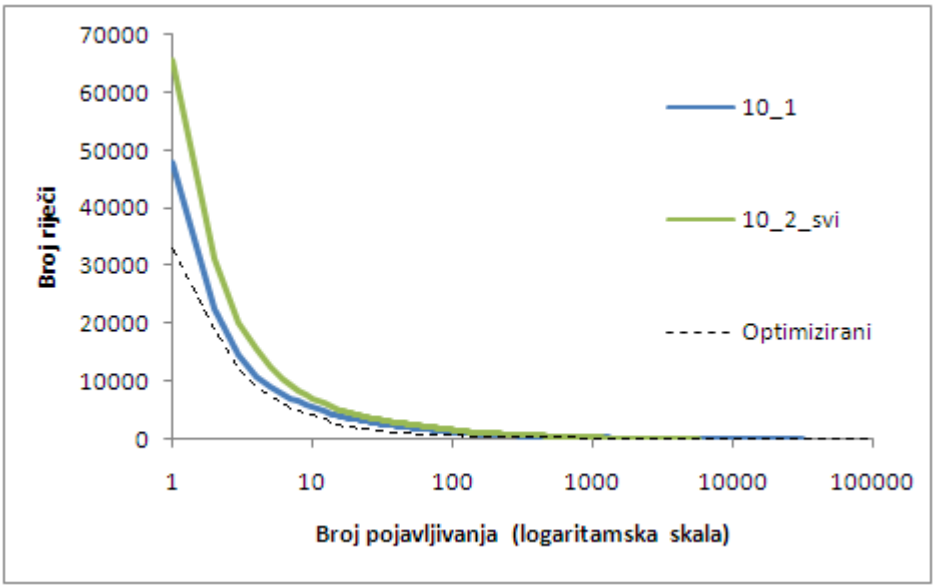
35_1	35_2_svi
-0,249%	-1,073%

Smanjivanjem broja separatora trebao bi se povećati broj riječi pogotovo kada se koristi opseg od 0 do 255 mogućih ASCII vrijednosti. Razlog za ovo je u manjem broju separatora koji rezultira prosječno dužim riječima i potencijalno većem broju kombinacija.

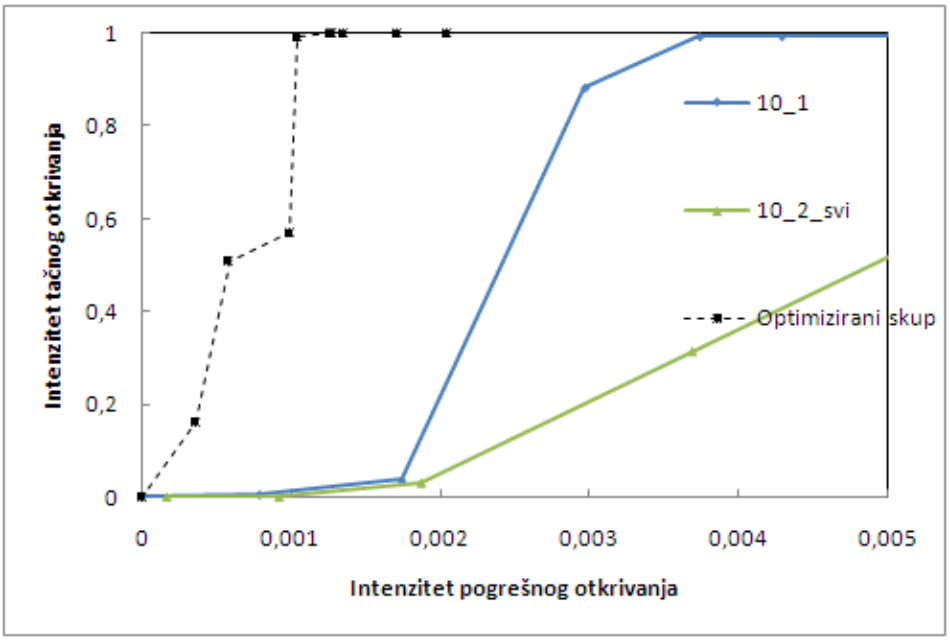
Napravljena su dva skupa od 10 separatora. Jedan u opsegu od 9 do 127, a drugi od 0 do 255. Nad ovim skupovima provedeni su testovi kao i za ranije skupove. Dobivene krive koje pokazuju karakteristike sistema za ove skupove prikazane su na slikama 46., 47. i 48. Uporedba AUC za ROC krive data je u tabeli XIV. Kako je i pretpostavljeno broj riječi se povećao. ROC krive su lošije nego za druge skupove. Ovaj broj (10) separatora utiče nepovoljno na sposobnost sistema da pravilno razlikuje pokušaje upada od normalnog saobraćaja. Broj znakova bi trebao biti veći od deset.



Slika 46. Broj naučenih riječi kao funkcija broja sati analiziranog saobraćaja za skupove separatora od 10 elemenata



Slika 47. Broj riječi u hash tabeli kao funkcija broja pojavljivanja za skupove separatora od 10 elemenata (logaritamska skala)



Slika 48. ROC kriva za skupove separatora od 10 elemenata

Tabela XIV. Razlika između AUC ROC krivih za skupove separatora od 35 elemenata i AUC ROC krive za optimizirani skup separatora

10_1	10_2_svi
-0,430%	-0,412%

4.4.3.6 Zbirni rezultati

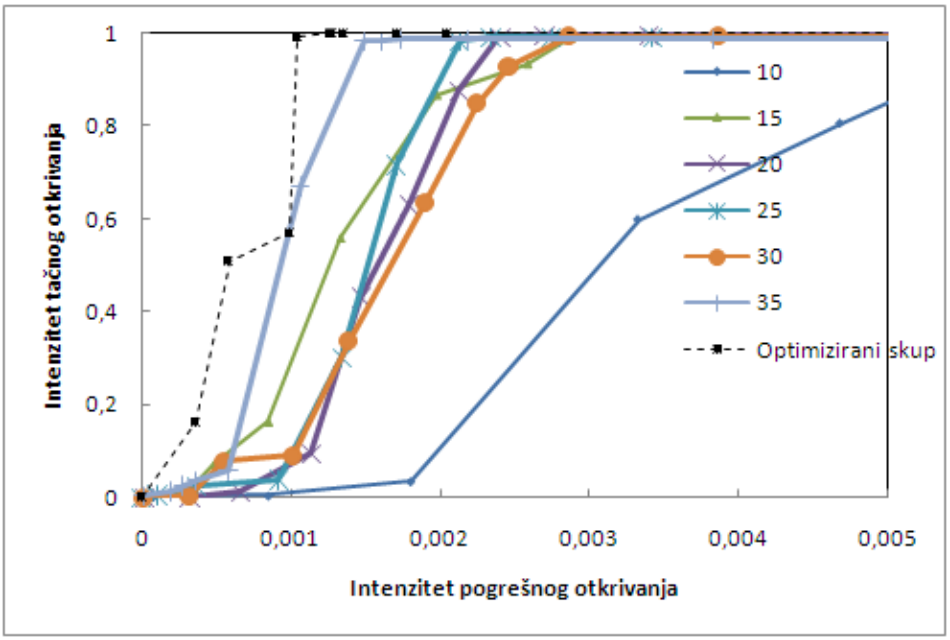
Radi ilustracije i poređenja dobivenih rezultata za različite brojeve separatora napravljena je po jedna ROC kriva za svaki od prethodnih skupova testova. Zbirne ROC krive konstruisane su računanjem prosječnih vrijednosti intenziteta pogrešnog otkrivanja i intenziteta tačnog otkrivanja za svaki skup testova sa jednakim brojem znakova. Na ovaj način bilo je moguće dobiti uvid u zavisnost uspješnosti otkrivanja upada od broja separatora.

Na slici 49. prikazane su ove prosječne ROC krive. Primjetno je da su četiri ROC krive prilično bliske, odnosno ukazuju na sisteme sa sličnim sposobnostima razlikovanja pokušaja upada od normalnog saobraćaja. Ove četiri ROC krive su za skupove koji su imali 15, 20, 25 i 30 separatora. Sve četiri su nešto lošije od ROC krive za sistem sa optimiziranim separatorima. Međutim i ove krive su dovoljno dobre za praktičnu upotrebu i uporedive ili bolje od krivih koje su dobili autori radova o sličnim sistemima navedenih u poglavlju o aktuelnoj problematici mrežnih sistema za otkrivanje upada.

Preostale dvije ROC krive se nešto razlikuju. Jedna je za sisteme sa 35 separatora i bolja je od svih drugih ROC krivih. Međutim, kriva je rezultat samo dva testa, pa je rezultat manje pouzdan. Sa druge strane ustanovljeno je da generisanje 35 separatora može predstavljati poteškoću. Iz ovog razloga se smatralo da ovakvi sistemi mogu biti nepogodni za praktičnu upotrebu.

Druga ROC kriva je za skupove od deset separatora. Ova ROC kriva je osjetno lošija od ostalih. Znači da smanjivanje broja znakova na deset nepovoljno utiče na mogućnost detekcije.

Na ovaj način dobiven je i jedan sporedni rezultat testiranja. Utvrđene su gornja i donja granica za broj separatora.



Slika 49. Zbirne ROC krive za različite skupove separatora

Trideset obavljenih testova potvrdili su tezu iznijetu na početku poglavlja. Promjena skupa separatora, u određenim granicama, ne utiče nepovoljno na mogućnost razlikovanja normalnog saobraćaja od upada. Izbor slučajnog skupa separatora ne povećava drastično broj riječi i ne kvari distribuciju. Skup separatora može se koristiti poput kriptografskog ključa. Svaka konkretna realizacija predloženog sistema može imati različit skup separatora. Sada sigurnost ovih sistema leži u ključu koji napadači ne znaju, a ne u načinu detekcije koji napadači mogu saznati. Na ovaj način ostvarena je predložena realizacija Kerckhoffs-ovog principa za sisteme za otkrivanje upada.

U nastavku će još biti data konkretna ilustracija koristi od ovakvih sistema za otkrivanje upada.

4.4.4 Otpornost na imitacijske napade

Jedan od osnovnih ciljeva predloženog uvođenja ključa u sisteme za otkrivanje upada je zaštita od imitacijskih napada. Ovdje će biti izložen jedan

mali i pojednostavljeni primjer koji pokazuje potrebu za ključem i dobit od ključa, odnosno izabranog skupa separatora. Primjer pokazuje na koji način napadač može pokušati da uklopi sadržaj napadačkog mrežnog paketa sa ovdje korištenim modelom normalnog ponašanja zasnovanom na riječima i prelazima između njih. Poznavanje algoritma učenja i detekcije pomaže napadaču da prilagodi sadržaj napadačkog paketa tako da u njemu bude što više normalnih riječi i prelaza. Ovako napravljen paket bi imao niži iznos odstupanja od modela normalnog ponašanje i napad bi mogao biti neotkriven.

Napadi broj 6, 7, 8 i 9 iz tabele IV zasnovani su na istom sigurnosnom propustu u *Rewrite* modulu Apache Web servera. Svaki od ovih napada koristi različit izvršni kod koji bi trebao izazvati različit efekat na napadnutom Web serveru i napadaču omogućiti različit način pristupa napadnutom serveru, prema objašnjenjima iz tabele II. Metasploit, alat korišten za pravljenje testnih napada, pravi napadački HTTP paket za ove napade koristeći slijedeći iskaz u svom jeziku:

```
uri= "/#{rewritepath}/ldap://" +
rand_text_alphanumeric(rand(16))+"/" +
rand_text_alphanumeric(rand(32))+"%3f" +
rand_text_alphanumeric(rand(8))+"%3f" +
rand_text_alphanumeric(rand(8))+"%3f" +
rand_text_alphanumeric(rand(16))+"%3f" +
rand_text_alphanumeric(rand(8))+"%3f%90"

uri += payload.encoded
```

Ovaj iskaz znači da sadržaj HTTP paketa ima šest dijelova koji su slučajni niz alfanumeričkih znakova dužina do 16, 32, 8, 8, 16 i 8 redom po dijelovima. Na početku paketa i između slučajnih dijelova su fiksni nizovi znakova. Nakon ovih dijelova dolazi „%3f%90“ i odgovarajući izvršni kod. Izvršni kod je različit za svaki od ova četiri napada. Obično je ovaj izvršni kod ono na osnovu čega se i otkriva napad, jer takav kod je uglavnom vrlo različit od normalnih paketa. Izvršni kod uglavnom utiče na povećanje iznosa odstupanja od modela normalnog ponašanja i napadači zbog toga žele

organizovati ovaj kod tako da se što više uklapa u model, ali zadržava funkcionalnost. Ovo organizovanje izvršnog napadačkog koda da zadrži funkcionalnost, a ne bude prepoznat kao napadački je daleko od trivijalnog. Zapravo vrlo je teško i njegovo objašnjavanje izlazi van okvira ovog rada. Iz ovog razloga u ovom primjeru neće se raditi nikakve promjene izvršnog koda. Bez umanjenja opštosti pristupa, mijenja se samo početni dio napadačkog HTTP paketa sa ciljem smanjivanja iznosa odstupanja od modela normalnog ponašanja.

Konkretni napadački HTTP paketi koje je Metasploit napravio za napade 6, 7, 8 i 9, bez izvršnog koda (koji je predug i ovdje nebitan), su:

Napad 6:

„GET /1/ldap://FrMMNAV/Yr05QaIibbMSFs6rKeVdkDVuMNo6UZ%3fzPM6%3fr%3fJM%3fUhyS%3f%90“ + izvršni.kod

Napad 7:

„GET /1/ldap://2QeT9jN5nS4QA9/HTiYB1T8LhY9Az9DTR9%3feG%3fu%3fHT%3fk%3f%90“ + izvršni.kod

Napad 8:

„GET /1/ldap://gkqXy/CpaqPUgtuLUp%3fvawd%3fH%3ff3%3fb3%3f%90“ + izvršni.kod

Napad 9:

„GET /1/ldap://uGQMixf/ApTBRRKBRKXLboZE9Q%3fEW4%3f%3f83rG%3fM5%3f%90“ + izvršni.kod

Kako se i vidi, svi napadi su napravljeni po opisanom obrascu. Nizovi slučajno generisanih znakova (podvučeni) vjerovatno nisu u skupu naučenih riječi i prelaza i povećavaju iznos odstupanja od modela normalnog saobraćaja nezavisno od, i u dodatku na iznos odstupanja koji je prouzrokovao izvršni kod.

Na osnovu poznavanja metode i optimiziranog skupa separatora gornji napadi su modifikovani da bi imali što niži iznos odstupanja uz isti efekat. Ovo je postignuto zamjenom slučajnih nizova znakova sa kombinacijom dvije riječi koje su u tom obliku vrlo česte u normalnim HTTP paketima i koje su razdvojene separatorom iz optimiziranog skupa. Ta kombinacija je: „GET HTTP“. Kombinacija ima osam znakova i zadovoljava ograničenje na dužinu za sve nizove slučajnih karaktera iz napada. Gdje je bilo potrebno i moguće prije i poslije kombinacije ubačen je po jedan razmak da bi se osigurao početak i završetak riječi na početku i kraju kombinacije „GET HTTP“. Sada sva četiri napada imaju isti prvi dio uz različit izvršni kod i glase:

„GET /1/ldap://GET HTTP/GET HTTP %3fGET HTTP%3fGET HTTP%3f GET HTTP %3fGET HTTP%3f%90“ + izvršni.kod

Ovako napravljeni napadi imaju identičan efekat, ali bi trebali imati niži iznos odstupanja od modela normalnog saobraćaja nego originalni Metasploit napadi za originalni sistem sa fiksnim skupom separatora. Ovi napadi su pokrenuti ka testnom Web serveru i za njih je izračunat iznos odstupanja koristeći optimizirani skup separatora i model normalnog sadržaja paketa dobiven na osnovu tog skupa. Iznosi odstupanja za originalne i modifikovane napadačke pakete dati su u tabeli XV.

Tabela XV. Iznosi odstupanja za originalne i modifikovane napade za optimizirani skup separatora

Br.	Izvršni kod	Originalni napad	Modifikovani napad
6	shell-bind_tcp	1,60	1,48
7	shell-reverse_tcp	1,59	1,40
8	vncinject-reverse_http	1,65	1,56
9	vncinject-reverse_tcp	1,56	1,37

Iznosi odstupanja nešto su niži nego iznosi za originalne napade. Iznos umanjenja otprilike je proporcionalan odnosu broja znakova (bajta) u koji su modifikovani i ukupnog broja bajta napadačkog paketa. Izvršni kod napada 7 i 9 je nešto kraći pa je za ove napade ostvareno najveće procentualno smanjenje. Izvršni kod napada 8 je najduži, pa je za ovaj napad ostvareno najmanje umanjenje iznosa odstupanja od modela normalnog ponašanja. Iznosi odstupanja za modifikovane napade vjerovatno nisu umanjeni dovoljno da bi se izbjeglo njihovo otkrivanje, ali pokazuju tendenciju i ilustruju ideju na osnovu koje bi, makar teoretski, bilo moguće dovoljno smanjiti ove iznose. Pogodno modifikovanje izvršnog koda, da se i on uklapa u model normalnog ponašanja, bi moglo ostvariti cilj.

Za iste modifikovane napade izračunat je iznos odstupanja koristeći drugi, pa treći testni skup separatora i model normalnog sadržaja paketa dobiven na osnovu tih skupova. Iznosi odstupanja za originalne i modifikovane napadačke pakete za drugi testni skup separatora dati su u tabeli XVI, a za treći u tabeli XVII.

Tabela XVI. Iznosi odstupanja za originalne i modifikovane napade za drugi testni skup separatora

Br.	Izvršni kod	Originalni napad	Modifikovani napad
6	shell-bind_tcp	1,21	1,39
7	shell-reverse_tcp	1,20	1,35
8	vncinject-reverse_http	1,31	1,32
9	vncinject-reverse_tcp	1,19	1,26

Tabela XVII. Iznosi odstupanja za originalne i modifikovane napade za treći testni skup separatora

Br.	Izvršni kod	Originalni napad	Modifikovani napad
6	shell-bind_tcp	1,35	1,35
7	shell-reverse_tcp	1,34	1,22
8	vncinject-reverse_http	1,23	1,35
9	vncinject-reverse_tcp	1,29	1,36

Za razliku od sistema sa optimiziranim skupom separatora, sistemi sa ključevima, slučajno generisanim skupovima separatora nisu bili nimalo prevareni modifikacijom napadačkih HTTP paketa. Iznosi odstupanja su, osim u dva slučaja, čak bili veći nego za originalne napade. Razlog za ovo je očigledan. Pošto su drugi i treći skup separatora bili „nepoznati“ prilikom izmjena napada nije se moglo znati na koji način modifikovati napad da se ostvari veći broj normalnih riječi i prelaza, te smanji iznos odstupanja i izbjegne otkrivanje napada.

Predloženi sistem za otkrivanje upada sa ključevima izrazito otežava ubacivanje zloćudnih niza bajtova u sadržaj paketa koji neće biti otkriveni. Poznavanje metode kreiranja modela sadržaja normalnih paketa i načina računanja iznosa odstupanja od ovog modela je od male pomoći, pošto je nepoznat skup separatora koji se koristi na lokaciji koja se želi napasti.

Neophodno je naglasiti da je uspješna modifikacija izvršnog napadačkog koda (da se uklopi u potrebna ograničenja koja nameće model normalnog ponašanja, a pri tome i dalje ostvari željeni efekat) uglavnom prilično teška, a ponekad i nemoguća. Ručne modifikacije paketa koje su ovdje napravljene iskoristile su veoma pogodan sigurnosni propust kao dokaz koncepta i za podršku predloženom sistemu za otkrivanje upada sa ključevima. U opštem slučaju predložena metoda sa optimiziranim skupom separatora može otkriti većinu automatizovanih napada. Dodavanje ključeva je bitan korak za dodatnu sigurnost i dobivanje prednosti u odnosu na napadače koji pokušavaju sakriti svoje pokušaje upada.

4.5 Zaključak

Uvođenje ključeva u sisteme za otkrivanje upada znatno povećava njihovu otpornost na pokušaje izbjegavanja otkrivanja napada. Ključ je tajna informacija, koja za isti algoritam otkrivanja upada omogućava različitu realizaciju sistema za otkrivanje upada. Na ovaj način sigurnost sistema nije u tajnosti algoritma detekcije već u tajnosti ključa. Ovakav pristup je posuđen iz kriptografije, gdje predstavlja fundamentalni princip. Princip se naziva Kerckhoffs-ov, po jednom od šest principa za dizajn praktičnih kriptografskih šifratora [148] [149]. Primjena ovog principa na sisteme za otkrivanje napada čini pravljenje imitacijskih napada izuzetno teškim ili čak i nemogućim, bez poznavanja ključa.

Praktična primjena principa na sisteme za otkrivanje upada predstavljena je kroz realizaciju. Realizacija je zasnovana na sistemu predstavljenom u prvom

dijelu rada. Taj sistem otkriva pokušaje upada pronalazeći anomalije u sadržajima mrežnih paketa. Analiza sadržaja paketa zasnovana je na podjeli sadržaja na riječi, uzastopne nizove bajtova ograničene separatorima. U tom sistemu skup separatora je fiksan. Promjenom skupa separatora mijenja se rezultujući model normalnog saobraćaja, te na njemu zasnovana detekcija pokušaja upada. Skup separatora iskorišten je kao ključ.

Testiranja sa različitim skupovima separatora dala su veoma obećavajuće rezultate. Predstavljeni su detaljni rezultati za dva skupa separatora i zbirni rezultati za još 28 skupova. Utvrđeno je da bi skup znakova trebao imati od 15 do 30 elemenata. Uvođenje skupa znakova kao ključa nije umanjilo sposobnost sistema da razlikuje normalne od zlоćudnih paketa. Sa druge strane uvođenje ključa otežalo je pravljenje napada koji sistem neće otkriti. Otpornost na glavnu klasu takvih napada, takozvane, imitacijske napade pokazana je na primjeru. Promjena ključa, po potrebi može se brzo obaviti. Potrebno je napraviti novi model normalnog ponašanja što je kratak proces jer je sistem brz u procesiranju paketa i učenju. U konkretnim testovima proces učenja trajao je oko deset minuta u prosjeku.

Proces promjene ključa može se i automatizovati. Sistem može sam generisati slučajni skup separatora. Sa ovim skupom sistem analizira saobraćaj onoliko vremena koliko je potrebno da napravi novi model. Nakon što ustanovi da je dovoljno naučio sistem automatski prelazi na novi model i detektuje pokušaje upada koristeći novi skup separatora. Ovaj proces se može dešavati u zadanim vremenskim intervalima ili može biti iniciran po potrebi. Ovakvim pristupom čak ni administrator sistema ne zna koji su separatori i čak ni on ne može napraviti imitacijski napad.

Druga korist od ovog pristupa, automatske promjene ključa, je ostvarivanje dinamičnosti modela. Snimanjem tekućeg saobraćaja i pravljenjem modela normalnog ponašanja na osnovu tog saobraćaja omogućava da se uvijek ima ažuran model stvarnog saobraćaja u mreži.

Još jedna ideja za poboljšanje mogućnosti razlikovanja normalnih i zloćudnih pakete je paralelno korištenje više ključeva. Moguće je napraviti nekoliko modela normalnog saobraćaja na osnovu istog saobraćaja za učenja samo sa različitim ključevima, skupovima separatora. Analiza novog saobraćaja može se vršiti paralelnim poređenjem sa svakim od modela. Na ovaj način postiže se veća pouzdanost pravilne detekcije, jer se odluka donosi na osnovu rezultata više paralelnih analiza. U spornim situacijama može se napraviti neka vrsta glasanja na osnovu koje će se odlučiti da li se radi o upadu ili ne. Pošto je proces učenja kratak, model nije prevelik, a detekcija ne zahtjeva previše resursa navedena ideja bi bila praktično izvodiva.

ZAKLJUČAK

Ovaj rad predlaže jednu metodu i novi pristup dizajnu mrežnih sistema za otkrivanje upada zasnovanih na otkrivanju anomalija.

Kao uvod u problematiku na početku su objašnjene osnovne komponenete sigurnosti informacionih sistema. Definisane su prijetnje koje ugrožavaju sigurnost. Navedene su kontrole kojima se realizuje sigurnost. Razjašnjeni su pojmovi napada i upada. Napravljena je sistematizacija upada.

Nakon ovog uvoda posebno su obrađeni sistemi za otkrivanje upada. Navedena je sistematizacija ovih sistema. Poseban osvrt napravljen je na načine upoređivanja i vrednovanja različitih sistema. Različita otvorena pitanja iz ove oblasti su razmotrena. Aktuelna problematika i savremeni akademski radovi koji je obrađuju su analizirani.

Metoda predložena u radu zasnovana je na analizi sadržaja mrežnih paketa. Analiza sadržaja vrši se razdvajanjem na riječi. Riječi su nizovi uzastopnih bajta između znakova za razdvajanje - separatora. Utvrđen je skup separatora koji rezultira sa najvećim procentom smisaonih riječi iz sadržaja paketa. Model normalnog ponašanja sastoji se od dvije komponente. Jedna komponenta su frekvencije pojavljivanja riječi u normalnim paketima. Druga komponenta su frekvencije prelaza između dvije riječi. Formula za računanje iznosa odstupanja paketa koji se analiziraju od modela normalnog ponašanja uključuje obje komponente modela. Ova formula koja se koristi za razlikovanje pokušaja upada od normalnog saobraćaja napravljena je da bude otporna na određen broj napada u saobraćaju za učenje normalnog ponašanja. Formula je jednostavna. Jednostavnost formule omogućuje detekciju napada u realnom vremenu što je potvrđeno testiranjem na realnom sistemu.

Radi provjere uspješnosti otkrivanja upada obavljeno je testiranje. Kao testni protokol izabran je HTTP i razlozi za to su posebno objašnjeni. Za testiranje

je korišten stvarni mrežni saobraćaj i savremeni napadi napravljeni aktuelnim alatima za ovu namjenu. Svi testirani napadi su bili otkriveni pri čemu je broj lažnih uzbuna bio vrlo mali. Svi rezultati su u rangu ili bolji od rezultata drugih istraživača, što je pokazano upoređivanjem ROC krivih i površina ispod njih. Testirana je i potvrđena otpornost metoda na prisustvo malog broja napada u saobraćaju za učenje normalnog ponašanja. Ovo je bitna osobina jer nije realno očekivati da je za učenje dostupan realan saobraćaj koji je sigurno bez napada. Prvi očekivani ključni originalni doprinos je ostvaren.

Nakon metode predložen je i sasvim novi pristup dizajnu sistema za otkrivanje upada. Primjena principa, poznatog iz kriptografije, da sigurnost sistema ne treba da zavisi od tajnosti dizajna već od tajnosti ključa na sisteme za otkrivanje upada povećava njihovu otpornost na takozvane imitacijske napade. Ovi napadi, koji su posebno objašnjeni, su jedna od najvećih prijetnji sistemima zasnovanim na otkrivanju anomalija.

Realizacija novopredloženog pristupa pokazana je na sistemu iz prvog dijela rada. Skup separatora sadržaja paketa na riječi korišten je kao ključ. Iznesena je pretpostavka da bi metod trebao biti u stanju razlikovati normalni saobraćaj od napada i za neki skup separatora različit od optimiziranog skupa korištenog u prvom dijelu rada. Obavljen je veliki broj testova sa različitim skupovima znakova koji su potvrdili da sposobnost sistema da otkriva pokušaje upada ne zavisi od skupa separatora. Testovi su potvrdili i da promjena skupa separatore ne dovodi do višestrukog povećanja riječi niti kvari njihovu distribuciju. Utvrđeno je i koliko bi minimalno i maksimalno elementa trebao imati skup separatora za praktičnu upotrebu. Uspješno je testirana otpornost predloženog pristupa na imitacijske napade.

Predloženi pristup omogućava da svaka realizacija sistema koristi različit skup separatora. Na ovaj način realizovan je princip da je sigurnost sistema za otkrivanje upada u tajnosti skupa separatora, a ne u tajnosti metode

otkrivanja. Ovim su realizovana i preostala tri očekivana doprinosa rada iznesena u uvodu.

Dalja istraživanja mogu biti usmjerena u različitim pravcima. Sa jedne strane može se istraživati osnovni metod predložen u prvom dijelu rada. Ovaj metod mogao bi se testirati i za druge protokole. Tu su prvenstveno drugi tekstualni protokoli kao SMTP, IMAP, POP i FTP. Metod bi mogao biti korišten i za DNS protokol. Mogućnost korištenja metoda za šifrirane protokole SSL, TLS i SSH je otvoreno pitanje koje treba provjeriti.

Kako se metod bavi analizom sadržaja koji ima određeno značenje, iskazano riječima protokola, mogao bi se povezati sa metodama analize teksta, prepoznavanja jezika i semantičke analize. Otkrivanje anomalija je poput traženja riječi ili izraza koji ne pripadaju nekom tekstu. Bliska oblast bi mogla biti analiza teksta usmjerena na otkrivanje autora, odnosno utvrđivanja da li su različiti tekstovi djelo istog autora.

Drugi pravac istraživanja bi mogao biti ka sistemima za otkrivanje upada sa ključevima. Izbor skupa separatora može se detaljnije razmotriti. Kriptografija ima posebnu oblast istraživanja koja se bavi ključevima. Kako ovaj skup ima ulogu ključa potrebno je ispitati da li postoje neki slabi ključevi ili ključevi koji su bolji ili lošiji od drugih.

Sistemi za otkrivanje upada sa ključevima bi možda mogli biti napravljeni i sa osnovnim metodom različitim od onog korištenog u ovom radu. Ideja je sasvim nova i moguće je da postoje njene bolje realizacije od ove.

BIBLIOGRAFIJA

[1] European Commission Eurobarometer, "E-Communications Household Survey," 2007.

[2] International Telecommunications Union (ITU), "World Information Society Report 2007," 2007.

[3] Organisation for Economic Co-operation and Development (OECD), "Malicious Software (Malware): A Security Threat to the Internet Economy," *Ministerial Background Report DSTI/ICCP/REG(2007)5/FINAL*, 2007.

[4] US Government Accountability Office, "Cybercrime: Public and Private Entities Face Challenges in Addressing Cyber Threats," 2007.

[5] D. Bell and L. La Padula, *Secure Computer Systems: Mathematical Foundations (Volume 1)*, Technical Report ESD-TR-73-278, Mitre Corporation, 1973.

[6] D.E. Bell and L.J. LaPadula, *Secure computer system: Unified exposition and MULTICS interpretation.* , Tech. Rep.: MTR-2997 Revision 1, The MITRE Corporation, 1976.

[7] Department of Defense, "Trusted Computer System Evaluation Criteria, DOD 5200.28-STD."

[8] K.J. Biba, *Integrity Considerations for Secure Computer Systems*, Bedford, MA : MITRE Corporation, 1977.

[9] D.D. Clark and D.R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, 1987, pp. 184-194.

[10] M. Bishop, *Introduction to Computer Security*, Addison-Wesley Professional, 2004.

[11] D.E. Denning and P.F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud & Security*, vol. 1996, 1996, pp. 12-16.

[12] J. Brainard, A. Juels, R.L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: somebody you know," *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 168-178.

[13] D. Ferraiolo, J. Cugini, and D.R. Kuhn, "Role-Based Access Control (RBAC): Features and Motivations," *Proceedings of 11th Annual Computer Security Application Conference*, 1995, pp. 11–15.

[14] B. Lampson, "Protection," *Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems*, 1971, pp. 437-443.

[15] P.J. Denning, "Third Generation Computer Systems," *ACM Comput. Surv.*, vol. 3, 1971, pp. 175-216.

[16] G.S. Graham and P.J. Denning, "Protection-principles and practice," *Proceedings of the 1972 AFIPS Spring Joint Computer Conference*, pp. 417-429.

[17] J. Saltzer and M. Schroeder, "The protection of information in computer systems," *IEEE, Proceedings*, vol. 63, 1975, pp. 1278-1308.

[18] ISO, "Information security management system – Requirements, ISO/IEC 27001," 2005.
[19] S. Zanero, "Unsupervised Learning Algorithms for Intrusion Detection - Ph.D. Thesis," Ph.D., DEI Politecnico di Milano, 2006.
[20] C.V. Lundestad and A. Hommels, "Software vulnerability due to practical drift," *Ethics and Information Technology*, vol. 9, 2007, pp. 89-100.
[21] R.J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2001.
[22] E.W. Dijkstra, "Chapter I: Notes on structured programming," *Structured programming*, Academic Press Ltd., 1972, pp. 1-82.
[23] *Internet Security Threat Report, vol. XII*, Symantec Corporation, 2007.
[24] US –National Institute of Standards and Technology, "National Vulnerability Database Home."; http://nvd.nist.gov/ (1.X 2008.)
[25] "CERT Statistics: Vulnerability Remediation."; http://www.cert.org/stats/vulnerability_remediation.html (1.X 2008.)
[26] R.W. Shirey, *Security Architecture for Internet Protocols. A Guide for Protocol Designs and Standards*, Internet Draft, 1994.
[27] A. The Institute of Risk Management, "A Risk Management Standard," 2002.
[28] ISO, *Risk management — Vocabulary — Guidelines for use in standards, ISO/IEC Guide 73:2002*, International Standards Organisation, 2002.
[29] A.K. Jones and W.A. Wulf, "Towards the Design of Secure Systems," *Software - Practice and Experience*, vol. 5, 1975, pp. 321-336.
[30] S. Lodin, *Intrusion Detection Product Evaluation Criteria*, Ernst & Young LLP, 1998.
[31] P. Mell and R. Bace, "NIST Special Publication 800- 31: Intrusion Detection Systems," *National Institute of Standards and Technology (NIST)*, vol. 31, 2001.
[32] P.G. Neumann and D.B. Parker, "A Summary of Computer Misuse Techniques," *Proceedings of the 12th National Computer Security Conference*, 1989, pp. 396-407.
[33] U.L. Lindqvist and E. Jonsson, "How to systematically classify computer security intrusions," *IEEE Symposium on Security and Privacy*, 1997.
[34] D.L. Lough, "A Taxonomy of Computer Attacks with Applications to Wireless Networks," Ph.D., Virginia Polytechnic Institute and State University, 2001.
[35] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: a taxonomy," *Network, IEEE*, vol. 16, 2002, pp. 13-21.
[36] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems," Master, Massachusetts Institute of Technology, 1999.
[37] D.J. Weber, "A taxonomy of computer intrusions," Master, Massachusetts Institute of Technology, 1998.
[38] J.D. Howard, "An Analysis of Security Incidents on the Internet, 1989-1995," Ph.D., Carnegie Mellon University, 1997.

[39] D.E. Denning, "An intrusion-detection model.," *IEEE Transactions on Software Engineering*, vol. 13, 1987, pp. 222-232.
[40] J.P. Anderson, *Computer Security Threat Monitoring and Surveillance, James P*, Fort Washington, PA: Anderson Co., 1980.
[41] D. Denning and P.G. Neumann, *Requirements and Model for IDES-a Real-time Intrusion-detection Expert System: Final Report*, Menlo Park, CA: SRI International, 1985.
[42] L.T. Heberlein, G.V. Dias, K.N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A network security monitor," *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, 1990, pp. 296-304.
[43] R. Heady, G. Luger, A. Maccabe, and M. Servilla, *The architecture of a network level intrusion detection system*, LA-SUB--93-219, Los Alamos National Lab., NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science, 1990.
[44] S. Snapp, J. Brentano, G. Dias, T. Goan, T. Grance, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, D.A.-.M. Mansur, K.A.-.P. Pon, and S.A.-.S. Smaha, "A system for distributed intrusion detection," *Compcon Spring '91. Digest of Papers*, 1991, pp. 170-176.
[45] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, and J.D. Tygar, "Can machine learning be secure?," *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006, pp. 16-25.
[46] S. Kumar and E. Spafford, *An Application of Pattern Matching in Intrusion Detection*, Department of Computer Sciences, Purdue University: 1994.
[47] S. Zanero and S.M. Savaresi, "Unsupervised learning techniques for an intrusion detection system," *Proceedings of the 2004 ACM symposium on Applied computing*, Nicosia, Cyprus: ACM, 2004, pp. 412-419.
[48] C. Kreibich and J. Crowcroft, "Honeycomb: creating intrusion detection signatures using honeypots," *ACM SIGCOMM Computer Communication Review*, vol. 34, 2004, pp. 51-56.
[49] H.A. Kim and B. Karp, "Autograph: toward automated, distributed worm signature detection," *Proceedings of the 13th conference on USENIX Security Symposium*, 2004.
[50] P. Mell, V. Hu, and R. Lippmann, "An overview of issues in testing intrusion detection systems," *National Institute of Standards and Technology ITL, July*, 2003.
[51] J. Neyman and E.S. Pearson, "The testing of statistical hypotheses in relation to probabilities a priori," *Proceedings of the Cambridge Philosophical Society*, vol. 29, 1933, pp. 492-510.
[52] J.P. Egan, *Signal detection theory and ROC analysis*, Academic Press New York, 1975.
[53] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers," *Machine Learning*, vol. 31, 2004.
[54] J.A. Hanley and B.J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, 1982, pp. 29-36.

[55] C. Cleverdon, "Evaluation Tests of Information Retrieval Systems," *Journal of Documentation*, vol. 26, 1970, pp. 55 - 67.
[56] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *Proceedings of the 23rd international conference on Machine learning*, Pittsburgh, Pennsylvania: ACM, 2006, pp. 233-240.
[57] R. Bunescu, R. Ge, R.J. Kate, E.M. Marcotte, A.K. Ramani, and Y.W. Wong, "Comparative experiments on learning information extractors for proteins and their interactions," *Artificial Intelligence in Medicine*, vol. 33, Feb. 2005, pp. 139-155.
[58] M. Goadrich, L. Oliphant, and J. Shavlik, "Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction," *Inductive Logic Programming*, 2004, pp. 98-115.
[59] R. Yasin, "High-Tech Burglar Alarms Expose Intruders," *InternetWeek*, Sep. 1997.
[60] F.B. Cohen, *A Short Course on Computer Viruses*, Wiley, 1994.
[61] A.M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. 2, 1937, pp. 230-265.
[62] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Trans. Inf. Syst. Secur.*, vol. 3, 2000, pp. 186-205.
[63] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Proceedings of the 7th conference on USENIX Security Symposium*, San Antonio, Texas: USENIX Association, 1998, pp. 3-3.
[64] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, Affordable, High Performance Statistics Collection," *Proceedings of INET*, vol. 97, 1997.
[65] N.S. Artan and H.J. Chao, "TriBiCa: Trie Bitmap Content Analyzer for High-Speed Network Intrusion Detection," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 125-133.
[66] A. Mahajan, B. Soewito, S.K. Parsi, N. Weng, and H. Wang, "Implementing high-speed string matching hardware for network intrusion detection systems," *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*, 2008, pp. 264-264.
[67] J. O'Reilley, *Tools for Intrusion Detection Beyond the Firewall*, Gartner Inc., 1997.
[68] R. Stiennon and M. Easley, *Intrusion Prevention Will Replace Intrusion Detection*, Gartner Inc., 2002.
[69] S. Northcutt, *Computer Security Incident Handling: Step-by-Step*, SANS Institute, 2003.
[70] T.H. Ptacek and T.N. Newsham, *Insertion, evasion, and denial of service: Eluding network intrusion detection*, Secure networks Inc., 1998.
[71] S.M. Bellovin, "Packets found on an internet," *SIGCOMM Comput. Commun. Rev.*, vol. 23, 1993, pp. 26-31.

[72] R. Chinchani and E. van den Berg, "A Fast Static Analysis Approach to Detect Exploit Code Inside Network Flows," *In 8th International Symposium on Recent Advances in Intrusion Detection (RAID)* , 2005.

[73] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic blending attacks," *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, Vancouver, B.C., Canada: USENIX Association, 2006, p. 17.

[74] C. Krügel, T. Toth, and E. Kirda, "Service specific anomaly detection for network intrusion detection," *Proceedings of the 2002 ACM symposium on Applied computing*, Madrid, Spain: ACM, 2002, pp. 201-208.

[75] T. Toth and C. Kruegel, "Accurate Buffer Overflow Detection via Abstract Pay load Execution," *In Recent Advances in Intrusion Detection (RAID)*, 2002, pp. 274-291.

[76] M.V. Mahoney, "Network traffic anomaly detection based on packet bytes," *Proceedings of the 2003 ACM symposium on Applied computing*, Melbourne, Florida: ACM, 2003, pp. 346-350.

[77] K. Wang and S.J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," *In 7th International Symposium on Recent Advances in Intrusion Detection (RAID)* , 2004.

[78] M. Rash, A.D. Orebaugh, G. Clark, B. Pinkard, and J. Babbin, *Intrusion Prevention and Active Response: Deploying Network and Host IPS*, Syngress, 2005.

[79] SANS Institute, "SANS Top-20 2007 Security Risks (2007 Annual Update)."; http://www.sans.org/top20/ (1.X 2008.)

[80] M. Roesch, "Snort-Lightweight Intrusion Detection for Networks," *Proceedings of the 1999 USENIX LISA Systems Administration Conference*, 1999, pp. 229–238.

[81] L. Zhang and G.B. White, "Analysis of Payload Based Application level Network Anomaly Detection," *Proceedings of the 40th Hawaii International Conference on System Sciences*, IEEE Computer Society, 2007, p. 99.

[82] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting network intrusions," *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA: ACM, 2002, pp. 265-274.

[83] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," *Proceedings of the 10th ACM conference on Computer and communications security*, Washington D.C., USA: ACM, 2003, pp. 251-261.

[84] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," *Computer Networks*, vol. 48, Aug. 2005, pp. 717-738.

[85] W. Robertson, G. Vigna, C. Kruegel, and R.A. Kemmerer, "Using Generalization and Characterization Techniques in the Anomaly-based Detection of Web Attacks," *Proceedings of the 13th Symposium on Network and Distributed System Security (NDSS)*, 2006.

[86] K. Ingham, A. Somayaji, J. Burge, and S. Forrest, "Learning DFA representations of HTTP for protecting web applications," *Computer Networks*, vol. 51, Apr. 2007, pp. 1239-1255.

[87] P. Akritidis, E.P. Markatos, M. Polychronakis, and K. Anagnostakis, "Stride: Polymorphic sled detection through instruction sequence analysis," *Proceedings of the 20th IFIP International Information Security Conference (IFIP/SEC 2005)*, 2005.

[88] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, "Polymorphic Worm Detection Using Structural Information of Executables," *In 8th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2005.

[89] X. Wang, C. Pan, P. Liu, and S. Zhu, "SigFree: a signature-free buffer overflow attack blocker," *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, Vancouver, B.C., Canada: USENIX Association, 2006, p. 16.

[90] M. Polychronakis, K. Anagnostakis, and E. Markatos, "Emulation-Based Detection of Non-self-contained Polymorphic Shellcode," *Recent Advances in Intrusion Detection*, 2007, pp. 87-106.

[91] Y. Song, M.E. Locasto, A. Stavrou, A.D. Keromytis, and S.J. Stolfo, "On the infeasibility of modeling polymorphic shellcode," *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Virginia, USA: ACM, 2007, pp. 541-551.

[92] K. Wang, G. Cretu, and S.J. Stolfo, "Anomalous Payload-Based Worm Detection and Signature Generation," *In 8th International Symposium on Recent Advances in Intrusion Detection (RAID)* , 2005.

[93] P.C. Mahalanobis, "On the generalized distance in statistics," *Proc Natl Inst Sci India*, vol. 2, 1936, pp. 49-55.

[94] K. Wang, J. Parekh, and S. Stolfo, "Anagram: A Content Anomaly Detector Resistant to Mimicry Attack," *In Recent Advances in Intrusion Detection (RAID)*, 2006, pp. 226-248.

[95] R. Vargiya and P. Chan, "Boundary Detection in Tokenizing Network Application Payload for Anomaly Detection," *Workshop on Data Mining for Computer Security*, 2003.

[96] K. Rieck and P. Laskov, "Language models for detection of unknown attacks in network traffic," *Journal in Computer Virology*, vol. 2, 2007, pp. 243-256.

[97] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, "A sense of self for Unix processes," *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, 1996, pp. 120-128.

[98] S. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, pp. 151-180.

[99] C. Marceau, "Characterizing the behavior of a program using multiple-length N-grams," *Proceedings of the 2000 workshop on New security paradigms*, Ballycotton, County Cork, Ireland: ACM, 2000, pp. 101-110.

[100] E. Eskin, Wenke Lee, and S. Stolfo, "Modeling system calls for intrusion detection with dynamic window sizes," *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings*, 2001, pp. 165-175 vol.1.
[101] Wei, Xiaohong, Xiangliang, and Liwei, "Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data," *Computers & Security*, vol. 25, Oct. 2006, pp. 539-550.
[102] W.R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Professional, 1994.
[103] B.A. Forouzan, *TCP/IP Protocol Suite*, McGraw-Hill Science/Engineering/Math, 2005.
[104] IEEE, "802.3-2005 LAN/MAN CSMA/CD Access Method," 2005.
[105] IETF, "RFC 791 - Internet Protocol," 1981.
[106] IETF, "RFC 793 - Transmission Control Protocol," 1981.
[107] R. Fielding , J. Gettys , J. Mogul , H. Frystyk , L. Masinter , P. Leach , and T. Berners-Lee , "RFC2616 - Hypertext Transfer Protocol -- HTTP/1.1," Jun. 1999.
[108] B. Krishnamurthy and J. Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*, Addison-Wesley Professional, 2001.
[109] D. Gourley and B. Totty, *HTTP: The Definitive Guide*, O'Reilly Media, Inc., 2002.
[110] *Internet Security Threat Report*, Symantec Corporation, 2008.
[111] *2008 INTERNET SECURITY TRENDS*, IronPort and Cisco, 2008.
[112] T. Hastie, R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning*, Springer, 2003.
[113] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.
[114] K. Ingham and H. Inoue, "Comparing Anomaly Detection Techniques for HTTP," *Recent Advances in Intrusion Detection*, 2007, pp. 42-62.
[115] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, 2000, pp. 12-26 vol.2.
[116] L. Richard, J.W. Haines, D.J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Computer Networks*, vol. 34, Oct. 2000, pp. 579-595.
[117] J. McHugh, "The 1998 Lincoln Lab IDS Evaluation—A Critique," *Recent Advances in Intrusion Detection. Third International Workshop, RAID 2000*, 2000.
[118] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, 2000, pp. 262-294.

[119] C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: a provocative discussion," *Proceedings of the 2006 workshop on New security paradigms*, Germany: ACM, 2006, pp. 21-29.
[120] Tenable Network Security, "Nessus 3, the Network Vulnerability Scanner."; http://www.nessus.org/nessus/ (1.X 2008.)
[121] "Nikto 2, Web server scanner."; http://www.cirt.net/nikto2 (1.X 2008.)
[122] "The Metasploit Project."; http://www.metasploit.com/ (1.X 2008.)
[123] W.B. Cavnar and J.M. Trenkle, "N-Gram-Based Text Categorization," *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US: 1994, p. 161—175.
[124] M. Damashek, "Gauging Similarity with n-Grams: Language-Independent Categorization of Text," *Science*, vol. 267, 1995, pp. 843-8
[125] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2001.
[126] R. Jenkins, "A Hash Function for Hash Table Lookup."; http://www.burtleburtle.net/bob/hash/doobs.html (1.X 2008.)
[127] A.A. Markov, "Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga," *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, vol. 15, 1906, pp. 135-156.
[128] G. Zipf, *The Psycho-biology of Language: An Introduction to Dynamic Philology*, Houghton Mifflin Company, 1935.
[129] C.Y. Suen, "N-gram statistics for natural language understanding and text processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, 1979, pp. 164–172.
[130] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *J. Mach. Learn. Res.*, vol. 2, 2002, pp. 419-444.
[131] C. Watkins, "Dynamic alignment kernels," *Advances in large margin classifiers*, 1999.
[132] C. Leslie, E. Eskin, and W.S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," *Proceedings of the Pacific Symposium on Biocomputing*, 2002, pp. 566–575.
[133] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models ," *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, 1999, pp. 133-145.
[134] A.K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior proles for intrusion detection," *1st USENIX Workshop on Intrusion Detection and Network Monitoring*, 1999, pp. 11-12.
[135] W. Lee, S.J. Stolfo, and P.K. Chan, "Learning patterns from unix process execution traces for intrusion detection," *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, AAAI Press, 1997, pp. 50–56.
[136] M. Karim, A. Walenstein, A. Lakhotia, and L. Parida, "Malware phylogeny generation using permutations of code," *Journal in Computer Virology*, vol. 1, Nov. 2005, pp. 13-23.

[137] J.Z. Kolter and M.A. Maloof, "Learning to Detect and Classify Malicious Executables in the Wild," *J. Mach. Learn. Res.*, vol. 7, 2006, pp. 2721-2744.
[138] AACS Licensing Administrator, "Advanced Access Content System (AACS): Introduction and Common Cryptographic Elements," 2006.
[139] P. Gutman, "A Cost Analysis of Windows Vista Content Protection," Dec. 2006.; http://www.cs.auckland.ac.nz/~pgut001/pubs/vista_cost.html (1.X 2008.)
[140] Playfuls Staff, "HD DVD's AACS Protection Bypassed. In Only 8 Days?!," *Playfuls.com*, Dec. 2006.
[141] D. Wagner and R. Dean, "Intrusion detection via static analysis," *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, 2001, pp. 156-168.
[142] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA: ACM, 2002, pp. 255-64.
[143] K. Tan, K. Killourhy, and R. Maxion, "Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits," *Recent Advances in Intrusion Detection*, 2002, pp. 54-73.
[144] O. Kolesnikov and W. Lee, *Advanced Polymorphic Worms: Evading IDS by Blending in with Normal Traffic*, College of Computing, Georgia Tech, 2005.
[145] P. Fogla and W. Lee, "Evading network anomaly detection systems: formal reasoning and practical techniques," *Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria, Virginia, USA: ACM, 2006, pp. 59-68.
[146] "FIPS 46-2 - (DES), Data Encryption Standard."; http://www.itl.nist.gov/fipspubs/fip46-2.htm (1. X 2008.)
[147] "FIPS 197 - (AES), Advanced Encryption Standard," Nov. 2001.; http://www.itl.nist.gov/fipspubs/fip197.htm (1. X 2008.)
[148] A. Kerckhoffs, "La cryptographie militaire - Partie I," *Journal des sciences militaires*, vol. IX, Jan. 1883, pp. 5-83.
[149] A. Kerckhoffs, "La cryptographie militaire - Partie II," *Journal des sciences militaires*, vol. IX, Feb. 1883, pp. 161-191.
[150] C.E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, 1949, pp. 656-715.
[151] G. Miller, "Introduction," *The Psycho-biology of Language: An Introduction to Dynamic Philology*, MIT Press, 1965.
[152] W. Li, "Random texts exhibit Zipf's-law-like word frequency distribution ," *Information Theory, IEEE Transactions on*, vol. 38, 1992, pp. 1842-1845.
[153] B. Mandelbrot, "An informational theory of the statistical structure of language," *Communication theory*, 1953, pp. 486–502.