HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza mrežnog prometa

CCERT-PUBDOC-2004-09-90

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, **www.lss.hr**- laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. UVOD 4

2. MREŽNI PROTOKOLI 4

2.1. PODATKOVNI SLOJ 4
2.2. MREŽNI SLOJ 5
2.3. TRANSPORTNI SLOJ 6

3. NADGLEDANJE MREŽNOG PROMETA 7

4. OTKRIVANJE MALICIOZNIH MREŽNIH AKTIVNOSTI 9

4.1. OTKRIVANJE PREGLEDAVANJA PORTOVA 9
4.2. OTKRIVANJE DOS NAPADA 10
4.3. OTKRIVANJE ICMP STRAŽNJIH ULAZA 11
4.4. RANJIVOSTI U IMPLEMENTACIJI TCP/IP STOGA 11
4.5. OTKRIVANJE POKUŠAJA NEOVLAŠTENOG PRISTUPA 12

5. ZAKLJUČAK 13

6. REFERENCE 13

1. Uvod

Mrežni promet se odnosi na ukupnu količinu mrežnih transakcija odnosno paketa koji prolaze kroz računalnu mrežu. Iz količine i vrste mrežnog prometa može se mnogo doznati o namjeni i korištenju same računalne mreže. Iz perspektive računalne sigurnosti, analiza mrežnog prometa otkriva mnogo informacija o samoj mreži, načinima njenog korištenja te podacima koji se njome prenose. U ovom dokumentu opisano je kako se *ad-hoc* analiza mrežnog prometa može iskoristiti za otkrivanje i sprječavanje malicioznih aktivnosti unutar neke računalne mreže. Opisane tehnike korisne su u okruženjima gdje IDS (*engl. **I**ntrusion **D**etection **S**ystem*) sustavi nisu dostupni, a postoji potreba za analizom mrežnog prometa. Za mogućnost analize mrežnog prometa potrebno je poznavati mrežne protokole i standarde na kojima se zasniva računalna mreža koju se analizira. U dokumentu je opisana analiza Ethernet, IP, ICMP, TCP i UDP mrežnih protokola, zbog činjenice da se na njima zasniva većina modernih računalnih mreža. *Ad-hoc* nadgledanjem mrežnog prometa moguće je otkriti npr. stražnje ulaze na računalima unutar mreže, napade na različite Internet servise i Web aplikacije, napade uskraćivanjem računalnih resursa (*engl. **D**enial **o**f **S**ervice - DoS*), pregledavati mrežne portove, i sl. Za analizu mrežnog prometa potrebno je imati odgovarajući program za nadgledanje mrežnog prometa, a trenutno najpopularniji (besplatni) *real-time* programi ove namjene su `Tcpdump`, `Ethereal` i `Ethercap`. Za nadgledanje mrežnog prometa u dokumentu je korišten `Tcpdump` program zbog njegove fleksibilnosti, jednostavnosti i dostupnosti.

2. Mrežni protokoli

Podaci koji putuju računalnom mrežom prenose se u paketima. Svaki paket ima svoje zaglavlje (*engl. header*) u kojem se nalaze podaci o samom paketu. Ti podaci uključuju izvorišnu i odredišnu mrežnu adresu, izvorišni i odredišni mrežni port (ovisno o protokolu) i mnogo drugih podataka, no da bi računalni sustavi mogli međusobno komunicirati, proces komunikacije mora biti na neki način standardiziran. U smislu standardizacije računalnih komunikacija kreiran je OSI (*engl. **O**pen **S**ystems **I**nterconnection*) referentni model. OSI referentni model definira standarde koje računalni sustavi moraju slijediti kako bi mogli komunicirati sa ostalim računalnim sustavima. OSI model se dijeli na sedam slojeva (*engl. layers*) koji su objašnjeni u nastavku:

1. **Fizički sloj** – najniži sloj koji služi isključivo za uspostavljanje fizičkog kanala između računalnih sustava koji međusobno komuniciraju.
2. **Podatkovni sloj** – omogućuje prijenos podataka između računalnih sustava koji komuniciraju, zajedno sa kontrolom prijenosa.
3. **Mrežni sloj –** služi za usmjeravanje i adresiranje mrežnih paketa.
4. **Transportni sloj –** omogućava prijenos podataka u mrežnim paketima.
5. **Sjednički sloj –** služi za ostvarivanje i održavanje sjednice između mrežnih aplikacija.
6. **Prezentacijski sloj –** mrežnoj aplikaciji predaje podatke u njoj razumljivom obliku.
7. **Aplikacijski sloj –** odnosi se na samu mrežnu aplikaciju.

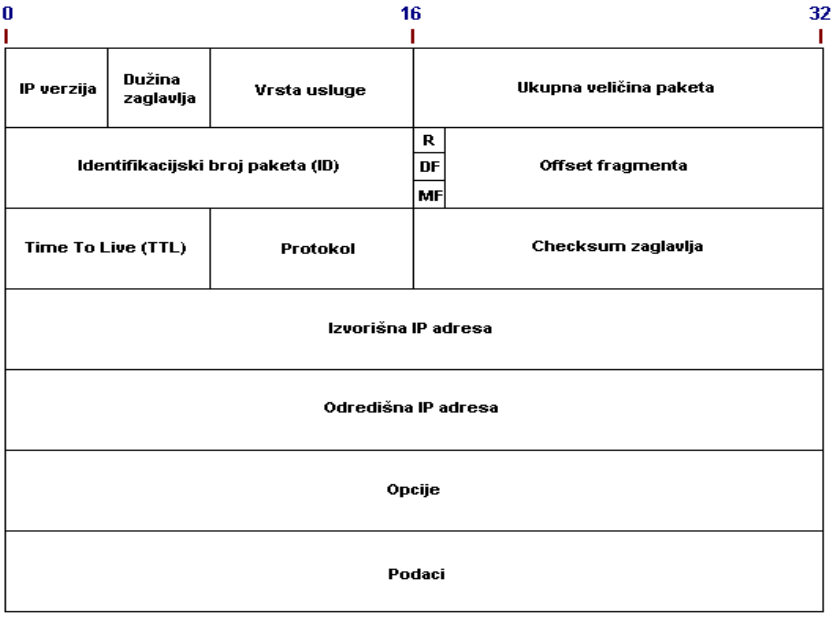
Svaki OSI sloj ima svoju funkciju i namjenu, a svi zajedno omogućavaju standardizirane mreže komunikacije. Za promatranje mrežnih transakcija, najvažniji OSI slojevi su podatkovni, mrežni i transportni koji su u nastavku detaljnije analizirani.

2.1. Podatkovni sloj

Kao što je već navedeno, podatkovni (*engl. data-link*) sloj služi za prijenos podataka između računalnih sustava i kontrolu prijenosa podataka između njih. Na podatkovnom sloju nalazi se i Ethernet protokol. U Ethernet mreži svako računalo ima svoju 48 bitnu MAC (*engl. **M**edia **A**ccess **C**ontrol*) adresu na osnovi koje se podaci usmjeravaju prema pojedinim računalima. Za Ethernet računalne mreže specifičan je ARP (*engl. **A**ddress **R**esolution **P**rotocol*) protokol koji služi za pretvaranje IP adrese u pripadajuću MAC adresu. ARP protokol detaljnije je objašnjen u dokumentu "*SSH Mitm Napadi*" koji je dostupan na adresi http://www.cert.hr.

2.2. Mrežni sloj

Iako mrežni sloj primarno služi za prijenos paketa i njihovo usmjeravanje, njegova funkcija mnogo je široka. Najpopularniji protokol na ovom sloju svakako je IP (*engl. **I**nternet **P**rotocol*) protokol koji spaja računalne mreže različitih tehnologija, topologija i primjene. IP familiju protokola (u koju se ubrajaju IP, TCP, UDP, ICMP) dizajnirale su vladine organizacije SAD-a 80-ih godina prošlog stoljeća u vojne svrhe. Osnovni cilj bio je dizajnirati računalnu mrežu koja će savršeno funkcionirati čak i ukoliko neki od njenih glavnih čvorova (*engl. node*) budu nedostupni. Danas je IP protokol najrašireniji u verziji 4, iako se sve više koristi verzija 6 koja će u neko dogledno vrijeme u potpunosti zamijeniti verziju 4. Glavni nedostaci IP verzije 4 su ograničeni adresni prostor (samo 32 bita) i nedostatak sigurnosnih kontrola ugrađenih u sam protokol. Na slici 1. prikazana je struktura IP paketa sa pojašnjenjem nekih važnija funkcionalnosti.



Slika 1: Struktura IP paketa

- **IP verzija** označava verziju IP protokola koja se koristi, a obično se radi o verziji 4.
- **Ukupna veličina paketa** odnosi se na ukupnu veličinu IP paketa.
- **Identifikacijski broj paketa** služi za određivanje redoslijeda paketa kako oni stižu na odredište. ID polje je vrlo važnu u postupku defragmentacije IP paketa nakon što oni stignu na odredište.
- **Time To Live (TTL)** polje određuje broj usmjerivača kroz koje paket može proći. Svaki usmjerivač umanjuje TTL vrijednost za jedan. Kad TTL polje dosegne vrijednost nula, usmjerivač odbacuje paket. TTL polje onemogućuje beskonačno kruženje mrežnih paketa na Internetu.
- **Protokol** označava tip protokola koji je enkapsuliran unutar IP paketa (npr. UDP ili TCP).
- **Izvorišna IP adresa** je adresa sa koje je IP paket poslan.
- **Odredišna IP adresa** određuje adresu na koju je IP paket poslan.
- **Podaci** se odnose na podatke koji se prenose unutar IP paketa.

Unutar mrežnog sloja koristi se i ICMP (*engl. **I**nternet **C**ontrol **M**essage **P**rotocol*) protokol. ICMP protokol uglavnom služi za otkrivanje aktivnih računala na mreži te dijagnostiku različitih grešaka na mreži. Za otkrivanje aktivnih računala koriste se ICMP `ECHO REQUEST` i `ECHO REPLY` poruke. ICMP poruke `NETWORK/HOST/PORT UNREACHABLE` označavaju nedostupnost određene računalne mreže, računala ili mrežnog porta što može biti uzrokovano fizičkom nedostupnošću sustava ili vatrozidnom zaštitom.

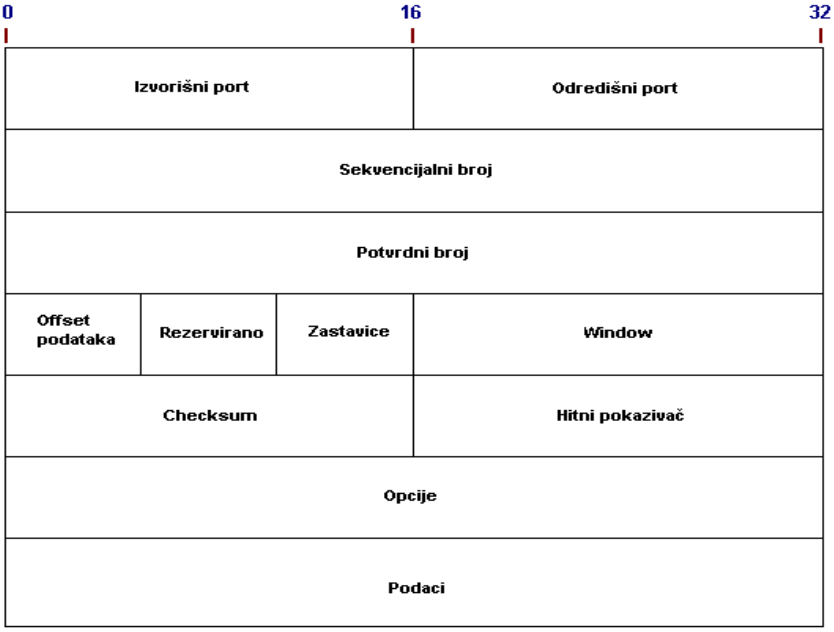
2.3. Transportni sloj

Na transportnom sloju nalaze se TCP (*engl. **T**ransport **C**ontrol **P**rotocol*) i UDP (*engl. **U**ser **D**atagram **P**rotocol*) protokoli. Pri prijenosu podataka, TCP protokol uspostavlja sjednicu između klijenta i poslužitelja, dok se UDP protokol bazira na prijenosu *datagrama* odnosno paketa, bez kontrole prijenosa na razini samog protokola. U nastavku su detaljnije opisane osnovne karakteristike TCP protokola.

Neke osobine TCP protokola:

- **Pouzdanost –** svaki poslani paket ima svoj potvrdni broj (*engl. Acknowledgment number*), te se paketi za koje nije dobiven potvrdni broj ponovo šalju.
- **Potpuna duplex komunikacija –** mogućnost primanja i slanja podataka istovremeno.
- **Sigurnost –** neovlašteni korisnik u osnovi ne može ubacivati lažirane pakete u otvorenu sjednicu (*engl. blind spoof*) ukoliko nema mogućnost nadgledanja te sjednice.

Na slici 2. prikazan je izgled TCP paketa i objašnjena su neka njegova važnija svojstva.

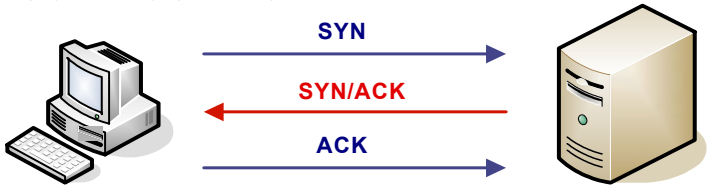


Slika 2. Struktura TCP mrežnog paketa

- **Izvorišni port** je mrežni port sa kojeg je poslan TCP paket.
- **Odredišni port** je mrežni port na koji je TCP paket poslan.
- **Sekvencijalni broj** (*engl. Sequence number)* služi za osiguravanje pouzdanosti TCP sjednice. Sekvencijalni brojevi označavaju pakete, te omogućavaju određivanje paketa koje je potrebno ponovo poslati.
- **Potvrdni broj** (*engl. Acknowledgment number*) je sekvencijalni broj inkrementiran za vrijednost 1.
- **Zastavice** (*engl. Flags*) označavaju stanje u kojem se nalazi TCP sjednica između klijenta i poslužitelja. TCP zastavice navedene su u nastavku:
  - **SYN** – sinhronizira sekvencijalne brojeve pri uspostavljanju sjednice,
  - **ACK** – potvrđuje uspostavljenu TCP sjednicu
  - **URG** – označava hitne pakete,
  - **PSH** – podatke je potrebno odmah iskoristiti (bez pohrane u međuspremnik)
  - **FIN** – označava zatvaranje TCP sjednice tako da u zatvaranju sudjeluju obje strane,
  - **RST** – prekida TCP sjednicu.
- **Window** – označava veličinu međuspremnika koju je potrebno osigurati da bi paket bio primljen.

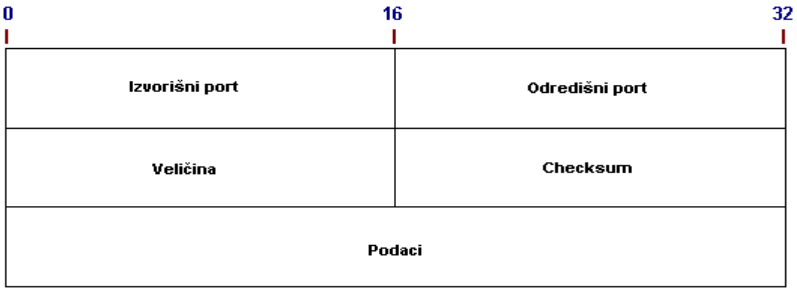
- **Checksum –** služi za ispitivanje integriteta TCP paketa koji se prenosi.

Za TCP protokol karakterističan je proces uspostavljanja sjednice tzv. *three-way handshake*. Računalo A koje želi uspostaviti TCP sjednicu sa računalom B šalje računalu B TCP paket sa SYN zastavicom i postavljenim sekvencijalnim brojem. Računalo B odgovara sa TCP paketom koji ima postavljene SYN/ACK zastavice, te sekvencijalni i potvrdni broj. Računalo A odgovara sa TCP paketom koji ima postavljenu zastavicu ACK, te sekvencijalni i potvrdni broj (Slika 2). *Three-way handshake* proces ovim završava i moguće je početi sa prijenosom podataka.



Slika 2: Three way handshake postupak uspostave TCP konekcije

UDP protokol je puno jednostavniji od TCP protokola, ali je samim time i manje pouzdan. Kao što je prije napomenuto, UDP protokol podatke prenosi u *datagramima*, bez ostvarivanja prave sjednice između klijenta i poslužitelja. Takva svojstva UDP protokola neovlaštenom korisniku pružaju puno više prostora za manipulacije nego što je to slučaj kod TCP protokola. Na slici 3. prikazan je izgled UDP paketa.



Slika 3: Struktura UDP paketa

3. Nadgledanje mrežnog prometa

Kako bi nadgledanjem mrežnog prometa bilo moguće detektirati maliciozne aktivnosti na računalnoj mreži, potrebno je znati koju je vrstu informacija potrebno tražiti. Vrsta informacije koja se u ovom slučaju traži ne uklapa se u uobičajeno funkcioniranje računalne mreže. Tu se može raditi o velikom broju TCP paketa sa SYN zastavicom koji ukazuju na DoS napad ili na pregledavanje otvorenih portova. Velik broj ICMP paketa može ukazivati na otkrivanje aktivnih računala unutar neke mreže ili na kontrolu stražnjeg ulaza. Pri nadgledanju mrežnog prometa važno je napomenuti da postoje dvije mogućnosti. Ukoliko se radi o preklapanoj računalnoj mreži (*engl. switched network*), u normalnim okolnostima, moguće je nadgledati samo promet usmjeren prema računalu na kojem je pokrenut program za nadgledanje mrežnog prometa. Postoji mogućnost aktivnog nadgledanja mrežnog prometa, no ona uključuje tehnike ARP preusmjeravanja (*engl. redirect*) i neće biti detaljnije analizirane u ovom dokumentu.

Drugi slučaj odnosi se na mreže sa mrežnim koncentratorom (*engl. hub*) gdje je moguće nadgledati kompletni mrežni promet koji prolazi kroz određeni segment računalne mreže, no potrebno je mrežno sučelje postaviti u tzv. *promiscuous* stanje u kojem mrežno sučelje prihvaća sve pakete koji stižu na njegovu MAC adresu.

Za nadgledanje i analizu mrežnog prometa u dokumentu je korišten `Tcpdump` alat koji je besplatan i može se dohvatiti sa adrese http://www.tcpdump.org. Upotreba `Tcpdump` alata vrlo je jednostavna, a glavne mogućnosti su prikazane u nastavku.

```
Usage: tcpdump [-adeflnNOpqStvx] [-c count] [ -F file ]
               [ -i interface ] [ -r file ] [ -s snaplen ]
               [ -T type ] [ -w file ] [ expression ]
```

Neke od važnijih opcija TCPDUMP programa navedene su u nastavku:

- **-i** - kao argument traži ime mrežnog sučelja sa kojeg će se čitati podaci
- **-s** - sa argumentom 0 omogućava ispisivanje kompletnog paketa, dok `tcpdump` standardno ispisuje samo prvih 68 bajtova primljenih paketa.
- **-v** - *verbose* - ispisuje više podataka o primljenom paketu (npr. za IP protokol ispisuje TTL, ID i veličinu paketa). Za više detalja moguće je dodati dva znaka `v` (`-vvv`).
- **-c** - opcija kao argument traži broj paketa koje će `tcpdump` prihvatiti i nakon kojih će prestati sa izvođenjem.
- **-X** - ispisuje sadržaj primljenog paketa u ASCII obliku.
- **-x** - ispisuje sadržaj paketa u heksadecimalnom obliku.
- **-U** - kao argument prima ime korisnika koji postaje vlasnik `tcpdump` procesa. Ova opcija podiže sigurnost jer `tcpdump` ima nekoliko poznatih i kritičnih sigurnosnih propusta.
- **-nn** - portovi i protokoli se kod ispisa ne pretvaraju u njihova imena.
- **-S** - ispisivanje apsolutnih umjesto relativnih sekvencijalnih brojeva.
- **-t** - ne ispisuje vrijeme na svakoj liniji.
- **expression** – uvjet koji mora biti ispunjen da bi paket bio prikazan. Može se odnositi na mrežu, računalo, mrežni protokol i mrežni port. Npr. "*dst host 192.168.0.22*" će ispisati sve pakete kojima je odredišna IP adresa 192.168.0.2, "*src port 1024*" će ispisati sve pakete kojima je izvorišni mrežni port 1024. Ukoliko se kao uvjet odredi samo protokol, npr. UDP, `tcpdump` će ispisivati samo UDP pakete. Ova mogućnost je vrlo korisna s obzirom da je u slučaju velikog broja mrežnih paketa teško odrediti koji su zanimljivi za analizu, a koji nisu.

Ukoliko se `tcpdump` pokrene bez opcija, `tcpdump` prihvaća sve protokole i prima pakete na svim mrežnim sučeljima.

```
[root@laptop NETWORK] tcpdump
tcpdump: listening on all devices
```

U drugom primjeru `tcpdump` program "sluša" na mrežnom sučelju `eth0`, prihvaća ICMP pakete i ispisuje sadržaj paketa u ASCII obliku.

```
[root@laptop NETWORK] tcpdump -i eth0 -X icmp
tcpdump: listening on eth0
```

Moguće je ispitivati i neke specijalne postavke unutar paketa koji dolaze na mrežno sučelje. U nastavku je prikazan izraz koji ispisuje sve TCP pakete sa postavljenom SYN zastavicom. 13 okteta od početka TCP zaglavlja nalaze se zastavice, te ukoliko se AND operacijom TCP zastavica i broja 2 dobije isto broj 2, paket ima postavljenu SYN zastavicu.

```
[root@laptop NETWORK] tcpdump -i eth0 -X 'tcp[13] & 2 == 2'
tcpdump: listening on eth0
```

U nastavku je prikazano otvaranje telnet sjednice između računala IP adrese 192.168.0.5 i računala pod imenom `pingu`. `Tcpdump` je pokrenut tako ispisuje samo TCP pakete, bez vremena pristizanja, sa apsolutnim sekvencijalnim brojevima i samo sa `eth0` mrežnog sučelja.

```
[root@wbox root] tcpdump -i eth0 -t -S tcp
tcpdump: listening on eth0
< 192.168.0.5.1806 > pingu.telnet: S 2511422633:2511422633(0) win
16384 <mss 1460,nop,nop,sackOK> (DF)
> pingu.telnet > 192.168.0.5.1806: S 3722137472:3722137472(0) ack
2511422634 win 32120 <mss 1460,nop,nop,sackOK> (DF)
< 192.168.0.5.1806 > pingu.telnet: . 2511422634:2511422634(0) ack
3722137473 win 17520 (DF)
> pingu.telnet > 192.168.0.5.1806: P 3722137473:3722137485(12) ack
2511422634 win 32120 (DF)
< 192.168.0.5.1806 > pingu.telnet: P 2511422634:2511422640(6) ack
3722137485 win 17508 (DF)
> pingu.telnet > 192.168.0.5.1806: . 3722137485:3722137485(0) ack
2511422640 win 32120 (DF)
> pingu.telnet > 192.168.0.5.1806: P 3722137485:3722137488(3) ack
2511422640 win 32120 (DF)
< 192.168.0.5.1806 > pingu.telnet: P 2511422640:2511422649(9) ack
3722137485 win 17508 (DF)
> pingu.telnet > 192.168.0.5.1806: . 3722137488:3722137488(0) ack
2511422649 win 32120 (DF)
```

Prve tri linije iz prethodnog primjera označavaju već prije spomenuti TCP *three-way handshake* postupak uspostave konekcije.
Žuta linija označava inicijalni TCP paket koji je poslan sa IP adrese 192.168.0.5 sa izvorišnog porta 1806 na računalo `pingu` na telnet port (23). Paket ima postavljenu zastavicu SYN koju označava slovo **S**. Sekvencijalni broj paketa je 2511422633.
Druga linija označena svjetlo plavom bojom je paket koji je poslan sa računala `pingu` sa izvorišnog porta 23 na IP adresu 192.168.0.5 na odredišni port 1806 kao odgovor na inicijalni SYN paket. Paket ima postavljene zastavice SYN i ACK. Sekvencijalni broj paketa je 3722137472, a potvrdni broj je 2511422634.
Posljednja žuta linija označava zadnji dio *three-way handshake* procesa. Sa IP adrese 192.168.0.5 sa izvorišnog porta 1806 poslan je paket na računalo `pingu` na telnet port. Sekvencijalni broj paketa je 2511422634, a potvrdni je 3722137473. Postavljena je samo ACK zastavica.
Tu završava *three-way handshake* postupak i ostali paketi predstavljaju sam prijenos podataka. Slovo **P** pokazuje da je postavljena zastavica PSH. Zastavica ACK označena je slovima **ack**, a odmah iza nje nalazi se i pripadajući potvrdni broj.

4. Otkrivanje malicioznih mrežnih aktivnosti

Pažljivim nadgledanjem mrežnog prometa moguće je detaljno analizirati aktivnosti na računalnoj mreži. Kako bi otkrili maliciozne aktivnosti, potrebno je poznavati način rada alata i tehnika koje koriste neovlašteni korisnici. U nastavku je prikazano nekoliko malicioznih mrežnih aktivnosti koje se vrlo lako mogu detektirati nadgledanjem mrežnog prometa.

4.1. Otkrivanje pregledavanja portova

Pregledavanje portova obično je prvi korak kojim neovlašteni korisnik otkriva potencijalno ranjive točke pomoću kojih će penetrirati u neki računalni sustav ili mrežu. Alati za pregledavanje otvorenih TCP portova variraju od vrlo jednostavnih *connect()* programa do onih koji iskorištavaju odstupanja od standarda definiranog u RFC dokumentu.
Ukoliko administrator pretpostavlja da se radi o jednostavnom programu za pregled otvorenih TCP portova na nekom udaljenom računalu, otkrivanje takvog napada vrlo je jednostavno. Potrebno je u mrežnom prometu tražiti inicijalne TCP pakete sa postavljenim SYN zastavicama koji služe za iniciranje sjednice. Ukoliko se radi o velikom broju takvih paketa, a odredišni portovi su različiti, vrlo vjerojatno se radi o pregledavanju mrežnih portova. U nastavku je priložen primjer otkrivanja pregledavanja otvorenih TCP portova. `Tcpdump` se pokreće tako da ispisuje sve TCP pakete sa postavljenom SYN zastavicom kao što je već opisano u dokumentu.

```
[root@laptop root]# tcpdump -i eth0 'tcp[13] & 2 == 2'
tcpdump: listening on eth0
15:45:07.383284 linux.2247 > laptop.24: S 702650068:702650068(0) win 32120
<mss 1460,sackOK,timestamp 151835 0,nop,wscale 0> (DF)
15:45:07.387419 linux.2248 > laptop.smtp: S 716075689:716075689(0) win 32120
<mss 1460,sackOK,timestamp 151835 0,nop,wscale 0> (DF)
15:45:07.391433 linux.2249 > laptop.26: S 712783750:712783750(0) win 32120
<mss 1460,sackOK,timestamp 151835 0,nop,wscale 0> (DF)
15:45:07.395527 linux.2250 > laptop.27: S 719674772:719674772(0) win 32120
<mss 1460,sackOK,timestamp 151836 0,nop,wscale 0> (DF)
15:45:10.393447 linux.2250 > laptop.27: S 719674772:719674772(0) win 32120
<mss 1460,sackOK,timestamp 152136 0,nop,wscale 0> (DF)
15:45:10.397608 linux.2251 > laptop.28: S 722051343:722051343(0) win 32120
<mss 1460,sackOK,timestamp 152136 0,nop,wscale 0> (DF)
15:45:10.401596 linux.2252 > laptop.29: S 710183442:710183442(0) win 32120
<mss 1460,sackOK,timestamp 152136 0,nop,wscale 0> (DF)
15:45:10.405693 linux.2253 > laptop.30: S 718256769:718256769(0) win 32120
<mss 1460,sackOK,timestamp 152137 0,nop,wscale 0> (DF)
15:45:13.343599 linux.2253 > laptop.30: S 718256769:718256769(0) win 32120
<mss 1460,sackOK,timestamp 152437 0,nop,wscale 0> (DF)
```

Kao što je vidljivo iz primjera, paketi koji služe za pregledavanje otvorenih portova dolaze sa računala pod imenom `linux` na računalo `laptop`. Žutom bojom označeni su izvorišni portovi na `linux` računalu, a svjetlo plavom odredišni portovi na računalu `laptop`. Odredišni portovi koji se testiraju inkrementirani su za jedan što je siguran znak pregledavanja portova. Izvorišni portovi također se

inkrementiraju za jedan što vrlo vjerojatno označava jednostavnu *connect()* tehniku pregledavanja portova. Pregledavanje portova moguće je otkriti i po velikom broju TCP paketa sa RST i ACK zastavicama. Radi se o situaciji kada program za pregledavanje portova pošalje TCP paket sa SYN zastavicom na zatvoreni port. U normalnim okolnostima otvoreni port bi odgovorio TCP paketom sa postavljenim SYN i ACK zastavicama, a zatvoreni port sa paketom sa postavljenim RST i ACK zastavicama. S obzirom da računala rimaju više zatvorenih nego otvorenih portova, pri pregledavanju portova, broj TCP paketa sa RST i ACK zastavicama je prilično velik. U nastavku je prikazano otkrivanje pregledavanja portova pomoću TCP paketa sa RST i ACK zastavicama. `Tcpdump` ispisuje sve pakete koji imaju kombinaciju zastavica jednaku 20, što se odnosi na postavljene zastavice RST i ACK.

```
[root@laptop root]# tcpdump -i eth0 'tcp[13] & 20 == 20'
tcpdump: listening on eth0
17:59:11.619111 laptop.smtps > linux.52568: R 0:0(0) ack 785423663 win 0 (DF)
17:59:11.619878 laptop.719 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620043 laptop.qmtp > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620284 laptop.640 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620372 laptop.969 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620467 laptop.link > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620633 laptop.5050 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620720 laptop.783 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620808 laptop.733 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620896 laptop.1540 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.620983 laptop.2041 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.621071 laptop.1027 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.621159 laptop.4987 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.621248 laptop.260 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
17:59:11.621336 laptop.838 > linux.52567: R 0:0(0) ack 2499638273 win 0 (DF)
```

Vidljivo je da računalo `laptop` sa raznih izvorišnih portova odgovara TCP paketima sa RST i ACK zastavicama računalu `linux` na odredišni port 52567. Brojevi izvorišnih portova koji odgovaraju nisu slijedni, što ukazuje na tehniku pregledavanja portova koju koristi popularni alat `nmap`.

Uz pregledavanje TCP portova moguće je i pregledavanje UDP portova. Pregledavanje UDP portova bazira se na slanju UDP paketa veličine 0 okteta na željeni odredišni port. Ukoliko program za pregledavanje portova dobije ICMP `PORT UNREACHABLE` poruku, testirani UDP port je zatvoren, a ukoliko nema odgovora, port je otvoren.

```
[root@laptop root]# tcpdump -i eth0
tcpdump: listening on eth0
16:08:57.394350 linux > laptop: icmp: echo request
16:08:57.394528 laptop > linux: icmp: echo reply
16:08:57.394500 linux.45550 > laptop.http: . ack 1811979525 win 3072
16:08:57.394636 laptop.http > linux.45550: R 1811979525:1811979525(0) win 0
(DF)
16:08:57.713151 linux.45530 > laptop.31337:  udp 0
16:08:57.713246 laptop > linux: icmp: laptop udp port 31337 unreachable [tos
0xc0]
```

Iz prethodnog primjera vidljivo je da se radi o pregledavanju UDP porta 31337 (popularni trojanski konj za Windows operacijke sustave) korištenjem `nmap` alata. Prije samog pregledavanja mrežnih portova, `nmap` šalje ICMP `ECHO REQUEST` i TCP paket sa ACK zastavicom na port 80 na odredišno računalo kako bi se ustanovilo da li je odredišno računalo aktivno. Nakon toga šalje se UDP paket veličine 0 okteta sa računala pod imenom `linux` sa porta 45530 na računalo pod imenom `laptop` na odredišni port 31337. Port nije otvoren, pa računalo `linux` dobiva ICMP `PORT UNREACHABLE` poruku.

4.2. Otkrivanje DoS napada

Napade uskraćivanjem resursa (*engl. Denial of Service*) lako je izvesti i vrlo su destruktivni. Popularan napad uskraćivanjem računalnih resursa odnosno tzv. *SYN flood* napad moguće je jednostavno otkriti nadgledanjem mrežnog prometa. Napad se bazira na velikom broju TCP paketa sa postavljenom SYN zastavicom koji su usmjereni na jedan ili više TCP portova na nekom računalu.

```
[root@laptop root]# tcpdump -i eth0 'tcp[13] & 2 == 2'
tcpdump: listening on eth0
18:50:53.223225  70.64.69.64.38663  >  laptop.http:  S  1769603072:1769603072(0)
win 512
18:50:53.755837  70.64.69.64.38663  >  laptop.http:  S  1769603072:1769603072(0)
win 512
```

```
18:50:54.601344 109.103.108.103.9737 > laptop.http: S 1769603072:1769603072(0)
win 512
18:50:54.761343 109.103.108.103.9737 > laptop.http: S 1769603072:1769603072(0)
win 512
18:50:55.938306 22.16.21.16.2315 > laptop.http: S 1769603072:1769603072(0) win
512
18:50:56.748821 22.16.21.16.2315 > laptop.http: S 1769603072:1769603072(0) win
512
18:50:57.592714 90.84.89.84.8968 > laptop.http: S 1769603072:1769603072(0) win
512
18:50:57.777047 90.84.89.84.8968 > laptop.http: S 1769603072:1769603072(0) win
512
18:50:57.936267  97.91.96.91.57350  >  laptop.http:  S  1769603072:1769603072(0)
win 512
18:50:58.500923  97.91.96.91.57350  >  laptop.http:  S  1769603072:1769603072(0)
win 512
```

Iz priloženog primjera vidljivo je da se radi o više IP adresa sa kojih pristižu TCP paketi sa postavljenim SYN zastavicama na računalo pod imenom `laptop` na port 80 (HTTP). U slučaju napada uskraćivanjem računalnih resursa uglavnom se radi o lažiranim izvorišnim IP adresama. Svi TCP paketi iz prethodnog primjera imaju isti sekvencijalni broj (označen žutom bojom), što je u normalnim okolnostima nemoguće, pa je vrlo lako zaključiti da se radi o lažiranim (*engl. spoof*) paketima koje je generirao isti program.

4.3. Otkrivanje ICMP stražnjih ulaza

Stražnji ulazi kontrolirani preko ICMP paketa vrlo su popularni kod neovlaštenih korisnika u posljednje vrijeme. Radi se o mogućnosti ICMP protokola da osim poruka prenosi i podatke. Ubacivanjem podataka u ICMP pakete moguće je ostvariti dobro skriveni komunikacijski kanal sa računalom na kojem je postavljen stražnji ulaz. Otkrivanje ovakvih stražnjih ulaza moguće je pregledavanjem kompletnih ICMP paketa koji dolaze na određenu mrežu ili računalo. U nastavku je prikazan sadržaj ICMP paketa koji upućuje na stražnji ulaz.

```
[root@laptop root]# tcpdump -i eth0 icmp -X
tcpdump: listening on eth0
18:16:53.057767 linux > laptop: icmp: echo request
0x0000   4500 0414 d053 0000 4001 2540 c0a8 0002     E....S..@.%@....
0x0010   c0a8 0003 0830 0140 5c41 0000 6361 7420     .....0.@\A..cat.
0x0020   2f65 7463 2f73 6861 646f 7720 3b20 6c73     /etc/shadow.;.ls
0x0030   202d 616c 202f 6574 632f 203b 2070 7320     .-al./etc/.;.ps.
0x0040   2d61 7578 0000 0000 0000 0000 0000 0000     -aux............
0x0050   0000                                        ..
```

Žuto označen dio predstavlja naredbe za ljusku (*engl. shell*) koje unutar ICMP paketa dokazuju postojanje stražnjeg ulaza na sustavu. Važno je napomenuti da kod ICMP stražnih ulaza IP adresa sa koje je paket pristigao također može biti lažirana. Na ovaj način moguće je otkriti stražnje ulaze bazirane na TCP i UDP protokolima, no ICMP je ovdje posebno napomenut jer se na njega često zaboravlja.

4.4. Ranjivosti u implementaciji TCP/IP stoga

TCP/IP stog označava dio jezgre operacijskog sustava koji je odgovoran za mrežnu funkcionalnost. Ponekad je nadgledanjem mrežnog prometa moguće otkriti i određene ranjivosti unutar TCP/IP stoga kod računala koja komuniciraju mrežom. Ranjivosti TCP/IP stoga odnose se npr. na TCP sekvencijalne brojeve koje je moguće predvidjeti ili na neotpornost TCP/IP stoga na neuobičajene mrežne pakete. U nastavku je prikazana mogućnost `tcpdump` alata da ispisuje različite tipove ICMP paketa. Tip paketa odnosno tip ICMP poruke određen je na samom početku ICMP paketa. Broj 8 označava tip paketa `ECHO REQUEST`, a broj 0 označava tip paketa `ECHO REPLY`. Pri pokretanju `tcpdump` programa dodana je i *verbose* opcija koja daje više informacija o samom paketu.

```
[root@laptop root]# tcpdump -i eth0 -v 'icmp[0] == 8'
tcpdump: listening on eth0
19:25:23.618367 winbox > laptop: icmp: echo request (ttl 32, id 18185, len 60)
19:25:24.623265 winbox > laptop: icmp: echo request (ttl 32, id 18441, len 60)
19:25:25.628202 winbox > laptop: icmp: echo request (ttl 32, id 18697, len 60)
19:25:26.633072 winbox > laptop: icmp: echo request (ttl 32, id 18953, len 60)
19:25:27.637996 winbox > laptop: icmp: echo request (ttl 32, id 19209, len 60)
19:25:28.642894 winbox > laptop: icmp: echo request (ttl 32, id 19465, len 60)
```

```
19:25:29.647791 winbox > laptop: icmp: echo request (ttl 32, id 19721, len 60)

[root@laptop root]# tcpdump -i eth0 -v 'icmp[0] == 0'
tcpdump: listening on eth0
19:25:49.656820 laptop > winbox: icmp: echo reply (ttl 255, id 1198, len 60)
19:25:50.661029 laptop > winbox: icmp: echo reply (ttl 255, id 1199, len 60)
19:25:51.665966 laptop > winbox: icmp: echo reply (ttl 255, id 1200, len 60)
19:25:52.670839 laptop > winbox: icmp: echo reply (ttl 255, id 1201, len 60)
19:25:53.675757 laptop > winbox: icmp: echo reply (ttl 255, id 1202, len 60)
19:25:54.630684 laptop > winbox: icmp: echo reply (ttl 255, id 1203, len 60)
19:25:55.635590 laptop > winbox: icmp: echo reply (ttl 255, id 1204, len 60)
```

ECHO REQUEST pakete generiralo je računalo pod imenom winbox, a ECHO REPLY pakete generiralo je računalo pod imenom laptop. *Verbose* opcija rezultirala je i ispisom ID vrijednosti IP zaglavlja. Paketi koje je generiralo winbox računalo imaju ID vrijednost koja se za svaki poslani paket inkrementira za 256, a paketi koje je generiralo računalo laptop imaju ID vrijednost koja se za svaki paket inkrementira za 1. Ovakvo statično ponašanje TCP/IP stoga kod generiranja ID brojeva predstavlja ranjivost koja može biti iskorištena u svrhu specijalne tehnike pregledavanja portova koja se naziva zombi tehnika. Zombi tehnika pregledavanja portova detaljnije je objašnjena na web adresi http://www.insecure.org/nmap. Statični ID brojevi mogu neovlaštenom korisniku u određenoj mjeri omogućiti i otkrivanje količine mrežnog prometa koji dolazi odnosno odlazi sa sustava koji ima ranjivu implementaciju TCP stoga u smislu generiranja ID brojeva.

4.5. Otkrivanje pokušaja neovlaštenog pristupa

Nadgledanje i analiza mrežnog prometa mogu otkriti i pokušaje penetracije u računalni sustav. Ukoliko neovlašteni korisnik želi iskoristiti određenu ranjivost sustava u koji želi penetrirati, on mora na neki način provjeriti da li je ranjivost prisutna. S obzirom da se ispitivanje vrši putem računalne mreže, analiza mrežnog prometa može otkriti pokušaje penetracije. U nastavku je dan primjer mrežnog prometa u slučaju kada neovlašteni korisnik na Web poslužitelju želi iskoristiti ranjivost . Za ovaj primjer, ime ranjive skripte je /cgi-bin/ranjiva_skripta.cgi.

```
[root@laptop root]# tcpdump -i eth0 -X -s 0
tcpdump: listening on eth0
17:24:13.167850 linux.4375 > laptop.http: S 2715921196:2715921196(0) win 32120
<mss 1460,sackOK,timestamp 749010 0,nop,wscale 0> (DF)
0x0000   4500 003c cdca 4000 4006 eb9b c0a8 0002     E..<..@.@.......
0x0010   c0a8 0003 1117 0050 a1e1 ab2c 0000 0000     .......P...,....
0x0020   a002 7d78 7ce6 0000 0204 05b4 0402 080a     ..}x|...........
0x0030   000b 6dd2 0000 0000 0103 0300 ce5f 5194     ..m.........._Q.
17:24:13.167999  laptop.http  >  linux.4375:  S  1400740319:1400740319(0)  ack
2715921197  win  5792  <mss  1460,sackOK,timestamp  1637408  749010,nop,wscale  0>
(DF)
0x0000   4500 003c 0000 4000 4006 b966 c0a8 0003     E..<..@.@..f....
0x0010   c0a8 0002 0050 1117 537d 99df a1e1 ab2d     .....P..S}.....-
0x0020   a012 16a0 fa17 0000 0204 05b4 0402 080a     ................
0x0030   0018 fc20 000b 6dd2 0103 0300               ......m.....
17:24:13.168535 linux.4375 > laptop.http: . ack 1 win 32120 <nop,nop,timestamp
749010 1637408> (DF)
0x0000   4500 0034 cdcb 4000 4006 eba2 c0a8 0002     E..4..@.@.......
0x0010   c0a8 0003 1117 0050 a1e1 ab2d 537d 99e0     .......P...-S}..
0x0020   8010 7d78 c204 0000 0101 080a 000b 6dd2     ..}x..........m.
0x0030   0018 fc20 6f2e fda9                         ....o...
17:24:31.429877  linux.4375  >  laptop.http:  P  1:44(43)  ack  1  win  32120
<nop,nop,timestamp 750841 1637408> (DF)
0x0000   4500 005f cdcc 4000 4006 eb76 c0a8 0002     E.._..@.@..v....
0x0010   c0a8 0003 1117 0050 a1e1 ab2d 537d 99e0     .......P...-S}..
0x0020   8018 7d78 401d 0000 0101 080a 000b 74f9     ..}x@.........t.
0x0030   0018 fc20 4845 4144 202f 6367 692d 6269     ....HEAD./cgi-bi
0x0040   6e2f 7261 6e6a 6976 615f 736b 7269 7074     n/ranjiva_skript
0x0050   612e 6367 6920 4854 5450 2f31 2e30 0ade     a.cgi.HTTP/1.0..
0x0060   9262 19                                     .b.
17:24:31.430533 laptop.http > linux.4375: . ack 44 win 5792 <nop,nop,timestamp
1639234 750841> (DF)
0x0000   4500 0034 65c0 4000 4006 53ae c0a8 0003     E..4e.@.@.S.....
0x0010   c0a8 0002 0050 1117 537d 99e0 a1e1 ab58     .....P..S}.....X
0x0020   8010 16a0 1a69 0000 0101 080a 0019 0342     .....i.........B
0x0030   000b 74f9                                   ..t.
17:24:31.970869  linux.4375  >  laptop.http:  P  44:45(1)  ack  1  win  32120
```

```
<nop,nop,timestamp 750895 1639234> (DF)
0x0000   4500 0035 cdcd 4000 4006 eb9f c0a8 0002     E..5..@.@.......
0x0010   c0a8 0003 1117 0050 a1e1 ab58 537d 99e0     .......P...XS}..
0x0020   8018 7d78 a951 0000 0101 080a 000b 752f     ..}x.Q........u/
0x0030   0019 0342 0a69 58d9 fb                      ...B.iX..
17:24:31.971345 laptop.http > linux.4375: . ack 45 win 5792 <nop,nop,timestamp
1639288 750895> (DF)
0x0000   4500 0034 65c1 4000 4006 53ad c0a8 0003     E..4e.@.@.S.....
0x0010   c0a8 0002 0050 1117 537d 99e0 a1e1 ab59     .....P..S}.....Y
0x0020   8010 16a0 19fc 0000 0101 080a 0019 0378     ...............x
0x0030   000b 752f                                   ..u/
17:24:31.972464  laptop.http  >  linux.4375:  P  1:238(237)  ack  45  win  5792
<nop,nop,timestamp 1639288 750895> (DF)
0x0000   4500 0121 65c2 4000 4006 52bf c0a8 0003     E..!e.@.@.R.....
0x0010   c0a8 0002 0050 1117 537d 99e0 a1e1 ab59     .....P..S}.....Y
0x0020   8018 16a0 d473 0000 0101 080a 0019 0378     .....s.........x
0x0030   000b 752f 4854 5450 2f31 2e31 2034 3034     ..u/HTTP/1.1.404
0x0040   204e 6f74 2046 6f75 6e64 0d0a 4461 7465     .Not.Found..Date
0x0050   3a20 5375 6e2c 2030 3520 5365 7020 3230     :.Sun,.05.Sep.20
0x0060   3034 2031 353a 3234 3a33 3120 474d 540d     04.15:24:31.GMT.
0x0070   0a53 6572 7665 723a 2041 7061 6368 652f     .Server:.Apache/
0x0080   312e 332e 3233 2028 556e 6978 2920 2028     1.3.23.(Unix)..(
0x0090   5265 642d 4861 742f 4c69 6e75 7829 206d     Red-Hat/Linux).m
0x00a0   6f64 5f73 736c 2f32 2e38 2e37 204f 7065     od_ssl/2.8.7.Ope
0x00b0   6e53 534c 2f30 2e39 2e36 6220 4441 562f     nSSL/0.9.6b.DAV/
0x00c0   312e 302e 3320 5048 502f 342e 312e 3220     1.0.3.PHP/4.1.2.
0x00d0   6d6f 645f 7065 726c 2f31 2e32 360d 0a43     mod_perl/1.26..C
0x00e0   6f6e 6e65 6374 696f 6e3a 2063 6c6f 7365     onnection:.close
0x00f0   0d0a 436f 6e74 656e 742d 5479 7065 3a20     ..Content-Type:.
0x0100   7465 7874 2f68 746d 6c3b 2063 6861 7273     text/html;.chars
0x0110   6574 3d69 736f 2d38 3835 392d 310d 0a0d     et=iso-8859-1...
0x0120   0a                                          .
```

Kao što je vidljivo iz primjera, neovlašteni korisnik se spaja sa računala pod imenom `linux` na računalo pod imenom `laptop` na port 80 (HTTP). Nakon *three-way handshake* procesa, na Web poslužitelj je poslan zahtjev (prvi žuto označen sadržaj paketa u primjeru) "`HEAD /cgi-bin/ranjiva_skripta.cgi HTTP/1.0`", koji ispituje da li se ranjiva Web skripta nalazi na polsužitelju. Web poslužitelj odgovara (drugi žuto označen sadržaj paketa u primjeru) sa "`HTTP/1.0 404 Not Found`" što znači da zatražena skripta nije pronađena. Osoba koja nadgleda mrežni promet sada je u mogućnosti definirati odgovarajuća pravila na vatrozidu koja će spriječiti neovlaštenog korisnika u daljnjim pokušajima penetracije.

Ovakvo nadgledanje mrežnog prometa može otkriti mnoge druge pokušaje penetracije kao što su napadi primjenom sile (*engl. brute force*) na proceduru za prijavljivanje npr. FTP, telnet ili POP servisa, iskorištavanje preljeva spremnika (*engl. buffer overflow*), posebne napade uskraćivanjem računalnih resursa itd.

5. Zaključak

Nadgledanjem mrežnog prometa moguće je saznati mnogo informacija o računalnoj mreži i aktivnostima na njoj. U dokumentu su prikazane različite tehnike i mogućnosti analize mrežnog prometa pomoću kojih legitimni korisnik (ali i neovlašteni korisnik) može detaljnije analizirati aktivnosti na računalnoj mreži za koju je nadležan. Važno je napomenuti da neovlašteni korisnici također koriste alate za nadgledanje mrežnog prometa uglavnom u svrhu skupljanja korisničkih imena i zaporki koje putuju nezaštićene računalnom mrežom. Nadgledanje i analiza mrežnog prometa aktivnost je koju bi svaki odgovorni mrežni administrator trebao s vremena na vrijeme provesti na svojoj mreži.

6. Reference

[1] Richard Stevenson, *"TCP/IP Illustrated, Volume 1"*
[2] Jon Erickson, "Hacking: The Art of Exploitation"
[3] Simson Garfinel, Gene Spafford & Alan Schwartz, *"Practical Unix & Internet Security"*
[4] Tcpdump manual, tcpdump(8)
[5] Nmap, http://www.insecure.org/nmap