HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza NMAP alata

CCERT-PUBDOC-2006-01-147

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.

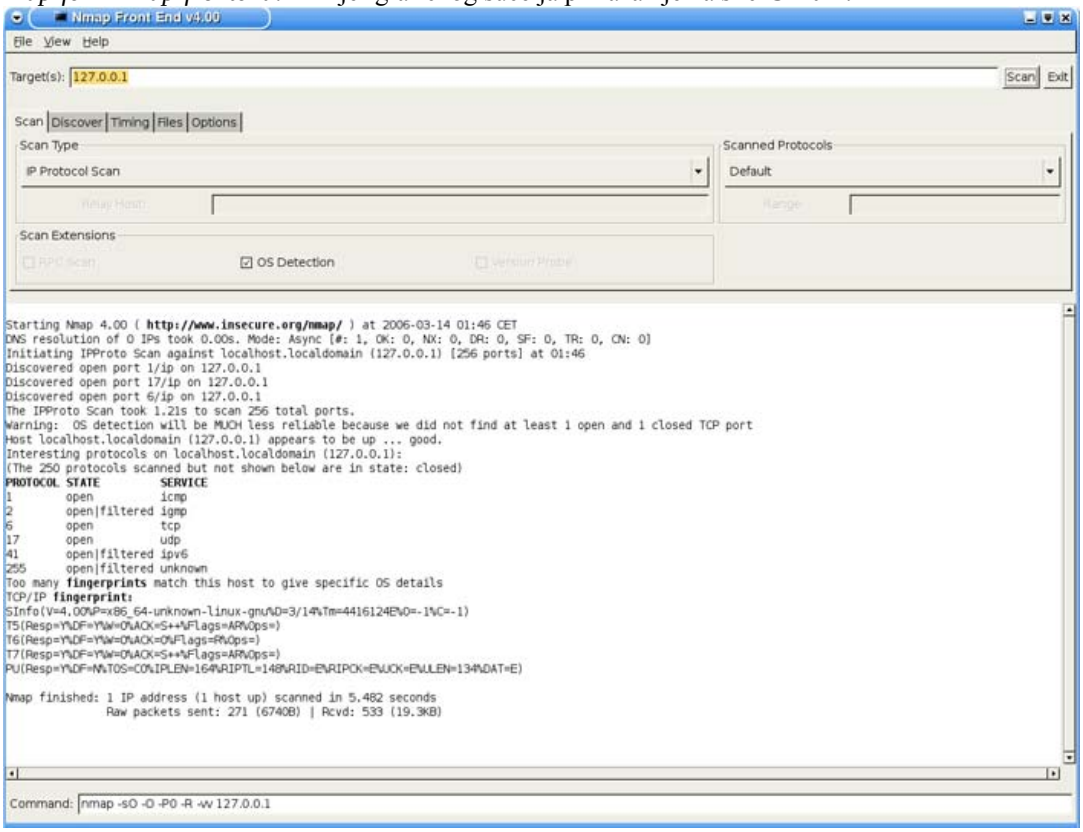
Sadržaj

1. Uvod

Nmap (*Network mapper*) je program otvorenog koda za ispitivanje mreže i provjeru sigurnosti. Program omogućuje pregledavanje TCP i UDP portova na ciljanom računalu, otkrivanje servisa koji slušaju na otvorenim portovima, prepoznavanje operacijskog sustava te još mnoge druge mogućnosti.
Trenutno je aktualna inačica 4.01. Program je podržan na velikom broju operacijskih sustava (Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga...)
Od inačice 3.5 program je doživio mnoga poboljšanja od kojih su samo neka: manja potrošnja memorije prilikom skeniranja portova, potpuno novi engine za skeniranje portova, interaktivnost za vrijeme izvršavanja i procjena vremena potrebnog za skeniranje. Također je puno truda uloženo u pisanje dokumentacije i uputa. Potpuna lista promjena može se vidjeti na službenim web stranicama alata [1].

2. Instalacija

Najnoviju inačicu *nmap*-a moguće je preuzeti sa službenih web stranica alata [2]. Na Windows platformi preporuča se preuzeti izvršni instalacijski paket umjesto do sada uobičajene zip arhive. Prilikom instalacije paketa, ukoliko je na računalu instaliran antivirsuni program McAffe VirusScan, javit će se poruka koja označava *nmap* kao „potencijalno neželjen program". Navedena poruka može se ignorirati, jer se radi o propustu unutar samog McAffe VirusScan-a.
S obzirom da je *nmap* nastao kao alat za Linux okruženje, na većini distribucija *nmap* je dostupan u vidu instalacijskog paketa, tako da je njegova instalacija poprilično jednostavna. Potrebno je samo skinuti paket kojeg koristi određena distribucija i nakon toga ga instalirati odgovarajućim alatom. Ukoliko se želi koristiti grafičko sučelje programa, potrebno je instalirati i paket koji se najčešće zove *nmap-fe* ili *nmap-frontend*. Primjer grafičkog sučelja prikazan je na slici *Slika 1*.



Slika 1: *nmap-fe* grafičko sučelje

Ukoliko se instalacija vrši iz izvornog koda, potrebno je u direktoriju u kojem se nalazi izvorni programski kôd pokrenuti `configure` skriptu, koja će provjeriti sve uvjete i stvoriti `Makefile` datoteku. Ukoliko se `configure` skripta uspješno izvršila program je potrebno prevesti naredbom `make` te ga zatim instalirati s `make install` naredbom. U suprotnom, potrebno je proučiti dojavljenu grešku te po potrebi instalirati pakete koji nedostaju.
Detalje vezane uz instalaciju na ostalim operacijskim sustavima moguće je pronaći na službenim web stranicama alata [3].

3. Ograničenja Windows inačice

Inačica *nmap*-a za Windows XP SP2, zbog promjena ostvarenih u SP2 kumulativnoj zakrpi, nema punu funkcionalnost koju ima UNIX/Linux inačica i još uvijek nije jednako učinkovita i stabilna. Neke od važnijih mogućnosti koje se ne mogu koristiti u Windows inačici su:

- skeniranje lokalnog računala, koje je na UNIX/Linux inačicama ostvarivo putem lokalne petlje na IP adresi 127.0.0.1,
- podržana su samo ethernet sučelja, dok RAS (eng. *Remote Access Service*) konekcije (poput PPP) nisu podržane,
- skeniranja najčešće traju dulje nego na UNIX/Linux platformama.

4. Razlikovanje portova

Funkcionalnost *nmap* alata se s vremenom poprilično proširila, no na svom početku on je bio samo program za pregledavanje otvorenih portova. To je do današnjeg dana i ostala njegova osnovna namjena. Većina programa namijenjenih pregledu portova razlikuju samo dva stanja porta: otvoreno i zatvoreno. Za razliku od njih, Nmap razlikuje čak 6 različitih stanja i to:

- `open (otvoren)` – postoji program koji aktivno sluša na portu i prima TCP ili UDP pakete. Otkrivanje ovakvih portova je najčešće i najvažniji razlog skeniranja. Osim što time otkrivamo potencijalne slabosti pregledanog sustava, također otkrivamo i koji su servisi dostupni u mreži.
- `closed (zatvoren)` – port je dostupan i on odgovara na upit poslan od strane *nmap*-a, ali nijedan program ne sluša na njemu.
- `filtered (filtrirano)` – *nmap* ne može odrediti da li je port otvoren ili zatvoren, zato što filtriranje paketa sprečava njegove upite da stignu do željenog porta. Najčešće filtri samo odbacuju pakete koji stižu na ovakve portove i ne vraćaju nikakvu informaciju izvoru paketa. S obzirom da nmap ne može znati da li je razlog ne dobivanja povratne informacije filtrirano stanje porta ili gubitak paketa, na ovakve portove se ponovno šalju upiti što značajno produžuje trajanje pregleda računala.
- `unfiltered (nefiltrirano)` – znači da je port u potpunosti dostupan, ali *nmap* nije u mogućnosti ustanoviti da li je port otvoren ili zatvoren. Jedino ACK tip pregleda dojavljuje ovakvo stanje porta. Pomoću drugih tipova pregleda (SYN, FIN) ponekad je moguće detaljnije utvrditi točno stanje ovakvih portova.
- `open|filtered (otvoren|filtriran)` – *nmap* klasificira port u ovakvo stanje kad ne može sa sigurnošću utvrditi da li je port otvoren ili filtriran. To se dešava kod otvorenih portova koji ne daju nikakav odgovor. Razlog tome može biti odbacivanje paketa od strane programa za filtriranje paketa, tako da *nmap* ne može sa sigurnošću znati da li je port otvoren ili filtriran.
- `closed|filtered (zatvoren|filtriran)` – ovo se javlja u slučaju kad *nmap* ne može ustanoviti da li je port zatvoren ili filtriran. Jedino IPID *idle* vrsta provjere vraća ovakav rezultat.

5. Korištenje alata

Osnovna sintaksa korištenja *nmap*-a je:

```
# nmap [vrsta pregleda] (opcije) {meta skeniranja}
```

Nmap tumači sve što je navedeno u naredbenom retku, a nije opcija, kao metu. Najjednostavniji način za navođenje mete je navođenjem njezine IP adrese. Adrese je moguće navoditi i pomoću CIDR notacije [5]. Primjerice, ako navedemo adresu 192.168.1.0/24 *nmap* će skenirati 256 računala s adresama u rasponu od 192.168.1.0 do 192.168.1.255. Najveći broj koji možemo navesti iza adrese je 32 što će rezultirati skeniranjem samo te adrese, a najmanji broj je 1. Ovakav način notacije ne radi za Ipv6 adrese.

Nmap također podržava istovremeno navođenje više adresa u jednom retku, npr:

```
# nmap scanme.nmap.org 192.168.0.0/16 10.0.0,1,3-7.0-255
```

Osim navođenja meta u komandnom retku, alat omogućava i korištenje sljedećih opcija za kontrolu meta:

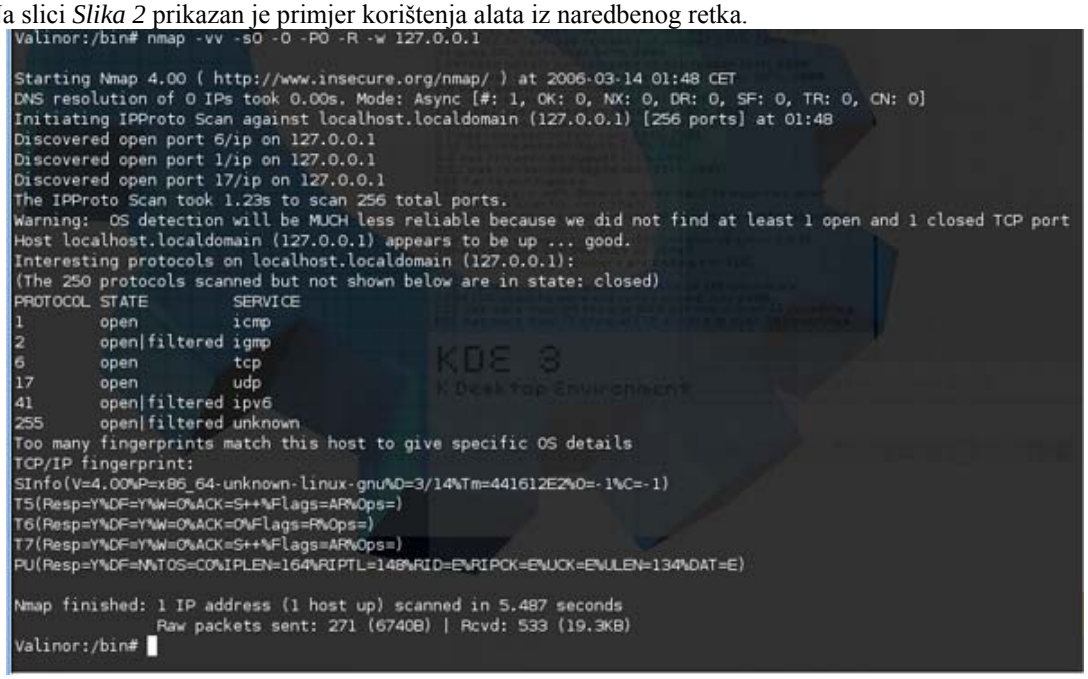
`-iL <ime ulazne datoteke>`
: Čita popis meta iz ulazne datoteke. Predavanje velikog broja adresa u naredbenom retku može biti poprilično naporno i nepregledno. Unosi u datoteci mogu biti u svim formatima koji su podržani i u naredbenom retku.

`-iR <broj adresa>`
: Ova opcija je namijenjena testiranju. Navođenjem ove opcije *nmap* će slučajno generirati *broj adresa* računala.

`--exclude <host1[,host2][,host3],...>`
: Navodimo listu adresa računala koja ne želimo skenirati, međusobno odvojenih zarezom. Lista adresa podržava sve formate kao i ostali načini unosa.

`--excludefile <datoteka_isključi>`
: Ima istu funkcionalnost kao i `–exclude` opcija s razlikom da se adrese navode u datoteci.

Na slici *Slika 2* prikazan je primjer korištenja alata iz naredbenog retka.



Slika 2: primjer korištenja *nmap* alata iz naredbenog retka

5.1. Pregled mreže

Prije pretraživanja otvorenih portova na nekoj mreži korisno je otkriti koje su IP adrese na mreži uopće zauzete kako ne bismo trošili vrijeme na pokušaje skeniranja nekorištenih IP adresa. Ova je mogućnost također korisna ukoliko želimo saznati koji su servisi dostupni na mreži ili želimo provjeriti da li netko koristi nedozvoljenu IP adresu.

Osnovne mogućnosti koje nam *nmap* nudi za pregled mreže su slijedeće:

`-sL`
: Ovo je općenita i jedna od najjednostavnijih mogućnosti pregleda mreže *nmap*-om. Zadavanjem ove naredbe *nmap* će samo ispisati sve adrese na mreži bez slanja ikakvih pakete. Međutim *nmap* će ipak napraviti reverzne DNS upite kako bi saznao imena računala.

`-sP` (*Ping scan*)
: Prilikom ovog pregledavanja mreže *nmap* će prikazati popis svih računala koja su odgovorila na skeniranje.

5.2. Vrste skeniranja portova

S obzirom da *nmap* prilikom svog rada koristi TCP i UDP protokole, korisnici bi trebali poznavati iste. Postoje brojni izvori na Internetu, od kojih su među referencama navedeni po jedan za TCP [8] i jedan za UDP protokol [9].

Većina pregleda dostupna je samo privilegiranim korisnicima. Razlog za to je taj što *nmap* šalje "sirove" (eng. *row*) TCP pakete, za što je na UNIX/Linux platformi potrebna *root* razina ovlasti. Korištenje administratorskog računa je preporučljivo i na Windows platformi, iako ono ponekad nije potrebno, ukoliko je na sistemu već instalirana *WinPcap* programska biblioteka.

Točnost podataka koje *nmap* prikupi ovisi isključivo o povratnim informacijama koje dobije od ciljanih računala ili vatrozida iza kojih se ona nalaze. Ti podaci međutim ne moraju uvijek biti točni s obzirom da računala mogu biti podešena da namjerno šalju lažne odgovore kako bi zavarali *nmap* i slične programe, no puno je češći slučaj da operacijski sustav ili programi koji se ispituju, nisu u skladu s RFC (eng. *Request for Comment*) standardima.

Istovremeno je moguće izvršavati samo jednu vrstu pregleda. Jedini izuzetak je mogućnost kombiniranja TCP i UDP pregleda. Osnovni tip pregleda je SYN pregled. U nastavku je dan kratki opis vrsta pregleda:

`-sS` (TCP SYN pregled)

SYN pregled je jedan od najčešće korištenih tipova pregleda. Za to postoji više razloga, od kojih je jedan i brzina pregleda, s obzirom da je na današnjim brzim mrežama koje su nesputane vatrozidima moguće pregledavati tisuće portova u sekundi. SYN pregled je teško opaziti jer nikad u potpunosti ne uspostavlja TCP vezu. Za ovu vrstu pregleda se često kaže da je poluotvoreni pregled (eng. *half-open scanning*) jer se potpuna veza nikada ne uspostavlja. *Nmap* šalje SYN paket na ciljano računalo, kao da želi uspostaviti vezu. Ukoliko je port otvoren računalo će odgovoriti sa SYN/ACK paketom, a u slučaju da nije, računalo će odgovoriti RST (*reset*) paketom. Ako nakon određenog vremena *nmap* ne primi nikakav odgovor ili primi neki od određenih ICMP poruka o nedostupnosti, označit će takav port filtriranim.

`-sT` (*TCP connect* pregled)

Ovo je prvi izbor pregleda ukoliko korisniku nije dostupan SYN pregled. Za razliku od SYN pregleda *connect* pregled koristi `connect()` funkciju operacijskog sustava. To je ista funkcija koju pozivaju i druge aplikacije (web preglednici, P2P programi i većina drugih mrežnih programa) kada žele uspostaviti vezu. S obzirom da *nmap* ima manji nadzor nad `connect()` funkcijom nego nad „sirovim" TCP paketima ovaj tip pregleda je nešto manje učinkovit nego SYN pregled. Sistemska funkcija `connect()`, ukoliko je port otvoren, uvijek uspostavi vezu u potpunosti. Posljedica takvog ponašanja je duže trajanje pregleda zbog većeg broja poslanih paketa, ali i zbog povećane vjerojatnost da ciljana računala zabilježe sve uspostavljene veze. Kako *nmap* zatvara vezu bez da slanja ikakvih podataka, neki loše napisani programi mogu se čak i srušiti prilikom ovakvog pregleda.

`-sU` (UDP pregled)

Iako većina najučestalijih mrežnih servisa danas koristi TCP protokol, UDP servisi su još uvijek poprilično rasprostranjeni. Primjeri najčešćih UDP servisa su DNS, SNMP i DHCP.

Kod UDP pregleda *nmap* šalje UDP paket koji sadrži samo zaglavlje (nema nikakvih ostalih podataka). Ukoliko *nmap* kao odgovor primi ICMP „*port unreachable error*" (tip 3, kod 3) odgovor, znači da je port zatvoren. Ostali tipovi ICMP „*port unreachable*" grešaka označavaju da je port filtriran. Ukoliko port odgovori UDP paketom znači da je otvoren. Ako nema nikakvog odgovora znači da bi port mogao biti otvoren, ali isto tako je moguće i da program za filtriranje paketa onemogućava slanje odgovora. Takvi portovi se označavaju kao *open|filtered*. Za preciznije otkrivanje stanja takvih portova moguće je koristiti pregled inačice (`-sV`). Jedan od najvećih problema prilikom UDP pregleda je pitanje brzine. Naime otvoreni i filtrirani portovi najčešće ne šalju nikakav odgovor što rezultira time da *nmap* ponovno šalje pakete na te portove, kako bi se eliminirala mogućnost gubljenja paketa na mreži. Još veći problem su zatvoreni portovi. Oni najčešće šalju ICMP „*port unreachable error*" paket kao odgovor. Međutim većina operacijskih sustava ograničava broj takvih paketa koje je moguće poslati.

`-sN, -sF, -sX` (TCP null, FIN i Xmas pregled)

Ove tri vrste pregleda koriste nedostatak u RFC standardu koji definira TCP protokol. Prilikom skeniranja sustava čija je implementacija TCP protokola u skladu sa odgovarajućim RFC-om,

na svaki paket koji nema postavljenu SYN, RST ili ACK zastavicu u zaglavlju paketa, mora se odgovoriti RST paketom. *Nmap* iskorištava ovaj propust slanjem FIN, PSH i URG paketa.
Null pregled ne postavlja nijednu zastavicu u zaglavlju (TCP „flag" zaglavlje je postavljeno na nulu). Fin pregled postavlja samo FIN bit. Xmas pregled postavlja FIN, PSH i URG zastavice. Ove tri vrste pregleda su po ponašanju u potpunosti isti. Jedina razlika je u postavljenim zastavicama u TCP zaglavlju. Ukoliko se kao odgovor primi RST paket, port se smatra zatvorenim. Ako nema odgovora port je otvoren|filtriran, a u slučaju primljenog ICMP *port unreachable error*-a port je filtriran.

`-sA` (TCP ACK pregled)
Ova vrsta pregleda razlikuje se od ostalih u tome što ne otkriva otvorene (pa čak ni otvorene|filtrirane portove), već služi isključivo za mapiranje pravila vatrozida i za otkrivanje koji su portovi filtrirani. ACK pregled šalje samo ACK paket. Otvoreni i zatvoreni portovi kao odgovor šalju RST paket te ih *nmap* označava kao *unfiltered*. Portovi koji ne odgovore ili odgovore nekom ICMP „*port unreachable*" greškom se označavaju kao *filtered*.

`-sW` (TCP Window pregled)
TCP Window pregled se razlikuje od ACK pregleda po tome što se mogu odrediti otvoreni i zatvoreni portovi. Window pregled koristi činjenicu da neki sustavi razlikuju otvorene i zatvorene portove prilikom slanja RST paketa. Na takvim sustavima „window" dio TCP paketa ima pozitivnu veličinu kad je port otvoren, dok je kod zatvorenih portova ta veličina jednaka nuli.
Ovakva primjena je relativno rijetka tako da rezultate ovakvog pregleda treba uzimati s rezervom.

`-sM` (*Maimon* pregled)
Maimon pregled je dobio ime po svom autoru Urielu Maimonu. Ovaj pregled je isti kao i *Xmas* pregled, s razlikom da se šalju FIN/ACK paketi. Prema RFC standardu 793, odgovor na ovakav paket bi trebao biti RST paket bez obzira da li je port otvoren ili zatvoren. *Maimon* je uočio da velik broj BSD sustava jednostavno odbacuje pakete na otvorenim portovima.

`--scanflags` (Proizvoljni TCP pregled)
Ova vrsta pregleda omogućuje naprednim korisnicima kreiranje vlastitog pregleda postavljanjem proizvoljnih TCP zastavica. Kao argument ovoj opciji moguće je predati brojčane vrijednosti, ali je jednostavnije koristiti simbolička imena (URG, ACK, PSH, RST, SYN, FIN).

`-sI` <zombie host[:port]> (*Idlescan*)
Kod ovog pregleda koristi se još jednog računalo - posrednik (*zombie*) s kojeg se šalju paketi na računalo koje se pregledava, čime se omogućava potpuna tajnost. Ovaj tip pregleda je izuzetno kompleksan te je detaljno opisan na stranicama programa [4].

`-sO` (pregled IP protokola)
Ovaj pregled se razlikuje od ostalih po tome što ovo nije pregled portova, već pregled otvorenih protokola.

5.3. Izbjegavanje vatrozida i IDS-a

U današnjem svijetu veliki se naglasak stavlja na sigurnost pa tako izuzetak nisu ni računalne mreže. Većina današnjih mreža zaštićena je vatrozidima. *Nmap* je između ostalog moguće koristiti za provjeru ispravnosti rada vatrozida. Međutim, *nmap* posjeduje i mehanizme za zaobilaženje loše konfiguriranih vatrozida.
Osim ograničavanja prometa vatrozidom, na većini današnjih mreža sustav sigurnosti uključuje i praćenje prometa na mreži korištenjem sustava za detekciju neovlaštenog prometa – IDS (eng. *Intrusion Detection System*). Većina IDS sustava prepoznaje skeniranje *nmap* alatom, ali je korištenjem nekih mogućnosti *nmap*-a moguće prikriti to skeniranje, a ponekad ga je moguće učiniti čak i u potpunosti nevidljivim za ovakve programe.
Neke od mogućnosti za izbjegavanje vatrozida i IDS sustava jesu:

`-f; --mtu`
`-f` opcija govori *nmap*-u da prilikom skeniranja razlomi pakete koje šalje u puno manjih paketa. Svrha ovakve aktivnosti je da TCP zaglavlja budu razlomljena u više paketa kako bi programima za filtriranje paketa, IDS-ovima i sličnim programima bilo čim teže shvatiti namjenu dobivenih paketa. Navođenjem `-f` opcije paketi će biti razlomljeni na dijelove

veličine 8 okteta. Ukoliko navedemo još jednom `-f`, veličina poslanih paketa bit će 16 okteta. Za slanje paketa proizvoljne veličine koristimo opciju `--mtu`. Broj korišten u kombinaciji s opcijom `--mtu` mora biti višekratnik broja 8.

`-D <mamac1 [,mamac2] [,ME],...>` (Prikrivanje skeniranja mamcima)

Ovom opcijom stvaramo privid ciljanom računalu da ga se istovremeno skenira s nekoliko računala. Na taj način otežavamo otkrivanje pravog izvora skeniranja. Važno je obratiti pažnju da sva računala koja koristimo kao mamce moraju biti u tom trenutku dostupna.

Korištenjem izraza `ME` u listi mamaca biramo mjesto na kojem će se nalaziti adresa našeg računala. Ukoliko ne navedemo `ME`, *nmap* će slučajno odabrati mjesto. U slučaju da smo `ME` stavili na šesto ili kasnije mjesto, neki programi za otkrivanje pregledavanja portova uopće neće zabilježiti adresu računala s kojeg se vrši skeniranje.

`--source-port <broj porta>; -g <broj porta>`

Vrlo česta greška prilikom konfiguracije vatrozida je vjerovanje prometu koji dolazi s određenog porta. Nakon postavljanja vatrozida često dolazi do problema u radu nekih programa. Primjerice, može doći do prestanka rada DNS-a jer odgovori vanjskih poslužitelja više ne mogu ući u lokalnu mrežu. Vrlo često se javlja i problem u radu FTP-a. Kod aktivnog FTP prijenosa, FTP poslužitelj pokušava otvoriti povratnu vezu prema klijentu kako bi prebacio tražene podatke.

Ponekad administratori vatrozida pribjegnu najbržem i najjednostavnijem rješenju takvih problema tj. pretpostave da će na određenim portovima komunicirati samo određeni programi. Tako primjerice pretpostave da sav promet koji dolazi s porta 53 dolazi od DNS-a, sav promet s porta 20 od aktivnog FTP-a i sl.

Nmap alat omogućava iskorištavanje ovakvih propusta u konfiguraciji vatrozida lažiranjem izvorišnog porta.

`--data-lenght <broj>`

Nmap uobičajeno šalje minimalne pakete koji sadrže samo zaglavlje. Korištenjem ove mogućnosti *nmap* će u većinu paketa dodati određeni broj okteta podataka do maksimalne veličine definirane s „`broj`“. To će usporiti skeniranje, ali će ga istovremeno učiniti manje uočljivim.

Potpuniji opis svih mogućnosti *nmap*-a moguće je pronaći na stranicama *nmap*-a na engleskom jeziku [6], ali također i na hrvatskom [7]. Potrebno je naglasiti da dokumentacija na hrvatskom nije potpuno ažurna pa se ipak preporuča proučiti englesku inačicu.

6. Zaključak

Nmap je izuzetno moćan program s brojnim mogućnostima, od kojih je samo manji dio opisan u ovom dokumentu. Razina korištenja programa u prvom redu ovisi o poznavanju TCP i UDP protokola. Stoga će iskusniji korisnici moći iskoristiti sve napredne opcije *nmap*-a. S druge strane, alat je jednostavan za korištenje pa ga mogu upotrijebiti i korisnici slabije upućeni u navedene protokole. Upravo to i jest razlog i jedna od velikih prednosti *nmap*-a, pa će program dobro doći različitim profilima korisnika, od onih početnih koji samo žele znati što se sve nalazi u njihovom mrežnom okruženju, pa do onih naprednih, koji će njime testirati sigurnost svojih mreža i ispravnost rada sigurnosnih mehanizama.

7. Reference

[1] http://www.insecure.org/stf/Nmap-4.00-Release.html
[2] http://www.insecure.org/nmap/download.html
[3] http://www.insecure.org/nmap/install/
[4] http://www.insecure.org/nmap/idlescan.html
[5] http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing
[6] http://www.insecure.org/nmap/man/
[7] http://www.insecure.org/nmap/man/hr/
[8] http://en.wikipedia.org/wiki/Transmission_Control_Protocol
[9] http://en.wikipedia.org/wiki/User_Datagram_Protocol