Primjena Network Mapper alata u mrežnoj dijagnostici

Tolj, Filip

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:751947

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

PRIMJENA NETWORK MAPPER ALATA U MREŽNOJ DIJAGNOSTICI

Završni rad

Filip Tolj

Osijek, 2020.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. OSNOVE NETWORK MAPPER SKENIRANJA 2
2.1. Sintaksa i korištenje 2
2.1.1 Greške u sintaksi 4
2.2 Odabir ciljanog računala 5
2.2.1. Skeniranje "nekoliko" računala 5
2.2.2. CIDR notacija i rasponi 5
2.2.3. Lista odredišnih računala 6
2.2.4. Izostavljanje hostova iz skeniranja 6
2.3 Otkrivanje domaćina 6
2.3.1. List scan (-sL) 7
2.3.2. Ping scan (-sP) 7
2.3.3. Sken bez pinganja (-PN) 8
2.3.4. TCP SYN Ping (-PS<port list>) 9
2.3.5. TCP ACK Ping (-PA<port list>) 10
2.3.6. UDP Ping (-PU<port list>) 10
2.3.7. ICMP Echo Ping (-PE) 11
2.3.8. ARP Scan (-PR) 12
3. SKENIRANJE PORTOVA 13
3.1. Stanja portova 13
3.2. Opcije za određivanje portova 14
3.2.1. Opcija –p <port ranges> 14
3.2.2. Preskakanje portova (--exclude-ports <port ranges>) 15
3.2.3. Brzi sken 15
3.3. TCP SYN scan (-sS) 15
3.3.1. Otvoreni port 16
3.3.2. Zatvoreni port 16
3.3.3. Filtrirani port 17
3.4. TCP Connect Scan (-sT) 17
3.5. UDP Scan (-sU) 18
3.6. TCP FIN, NULL, Xmas Scans (-sF, -sN, -sX) 19
3.7. TCP ACK Scan (-sA) 20
3.8. TCP Idle Scan (-sI) 21
3.8.1. Idle Scan otvorenog porta 22
3.8.2 Idle Scan zatvorenog porta 23
3.8.3 Idle Scan filtriranog porta 23

4. OSTALE OPCIJE I ZENMAP 24
4.1. Prepoznavanje servisa i njihovih verzija 24
4.2. Otkrivanje operacijskog sustava 25
4.3. NSE 26
4.3.1. Pokretanje podrazumijevanih skripti 26
4.4. Formatiranje izlaza 27
4.5. Zenmap 27
4.5.1. Skeniranje pomoću zenmapa 29
4.5.2. Usporedba rezultata 31
4.5.3. Stvaranje vlastitih profila 32
5. TESTIRANJE U VIRTUALNOM LABORATORIJSKOM OKRUŽENJU 33
5.1. Otkrivanje domaćina u laboratoriju 34
5.2. Quick scan laboratorijske mreže 35
5.3. Izbjegavanje vatrozida 39
5.3.1. Vatrozid koji odbacuje TCP SYN pakete 39
6. ZAKLJUČAK 41
LITERATURA 42
SAŽETAK 43
ABSTRACT 44

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 06.10.2020.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Filip Tolj
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3996, 28.09.2019.
OIB studenta:	37310562791
Mentor:	Izv. prof. dr. sc. Krešimir Grgić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Primjena Network Mapper alata u mrežnoj dijagnostici
Znanstvena grana rada:	**Telekomunikacije i informatika (zn. polje elektrotehnika)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	06.10.2020.
Datum potvrde ocjene Odbora:	14.10.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 20.10.2020.

Ime i prezime studenta:	Filip Tolj
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3996, 28.09.2019.
Turnitin podudaranje [%]:	4%

Ovom izjavom izjavljujem da je rad pod nazivom**: Primjena Network Mapper alata u mrežnoj dijagnostici**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Krešimir Grgić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

1. UVOD

Network mapper je objavljen 1997. godine kao besplatni mrežni skener otvorenog koda. U svojim počecima je to bio vrlo jednostavan i moćan alat koji je omogućavao efikasno skeniranje portova velikih mreža. Kroz godine su implementirane razne nove funkcionalnosti kao što su: udaljeno otkrivanje verzije operacijskog sustava, prepoznavanje usluga i njihovih verzija, nmap skriptni *engine* i mnoge druge. Također je predstavljeno i grafičko korisničko sučelje zenmap koje olakšava skeniranje i prikaz rezultata. Danas, više od 20 godina nakon objave, nmap se smatra jednim od najpoznatijih alata koji se koriste u mrežnoj dijagnostici te je postao gotovo neizbježan u testiranju sigurnosti računalnih sustava.

Rad je podijeljen u četiri dijela. U prvom dijelu rada objašnjeno je kako se *nmap* koristi kroz upoznavanje sa sintaksom samoga alata te su objašnjene metode odabira ciljanog računala i metode otkrivanja domaćina nad kojima će se vršiti daljnje skeniranje. U drugom dijelu je proučeno skeniranje portova te su detaljnije opisane i primjerima iz literature objašnjene neke od metoda za provođenje istog. Treći dio ukratko opisuje neke od ostalih mogućnosti *nmap* alata te pruža uvid u korištenje *zenmap* korisničkog sučelja. U četvrtom dijelu rada prikazana je primjena stečenih znanja u laboratorijskom okruženju kroz scenarij skeniranja lokalne mreže sa jednim ranjivim i jednim zaštićenim *hostom* te dodavanje i zaobilaženje pravila vatrozida ranjivog *hosta*.

1.1. Zadatak završnog rada

U suvremenim mrežama koje međusobno povezuju velik broj korisnika sve je teže kvalitetno obavljati zadatke nadzora, održavanja, ažuriranja i dijagnostike sigurnosnih problema. Za olakšanje ovih operacija primjenjuje se velik broj različitih alata. Jedan od najkorištenijih alata u mrežnoj dijagnostici i sigurnosnom pregledu je network mapper (nmap). Potrebno je detaljnije se upoznati sa mogućnostima i načinom rada ovoga alata, te ga implementirati unutar testnog okruženja. Potrebno je primijeniti network mapper u svrhu mrežne dijagnostike i sigurnosnog pregleda kroz nekoliko različitih testnih scenarija, izvedenih u laboratorijskom okruženju. Prikupljene podatke je potrebno detaljno analizirati i objasniti.

2. OSNOVE NETWORK MAPPER SKENIRANJA

Za komunikaciju *hostova* na računalnoj mreži, uz IP adrese koje određuju krajnje *hostove*, potreban je i mehanizam koji određuje između kojih procesa tih *hostova* se odvija komunikacija. Taj mehanizam je implementiran putem TCP i UDP protokola, protokola transportnog sloja OSI referentnog modela. U zaglavljima navedenih protokola definirani su port izvorišta i port odredišta koji su zapravo dva broja koji označavaju proces izvorišnog i proces odredišnog hosta koji komuniciraju. Standardne usluge poslužitelja koriste dobro poznate portove (*engl. well-known ports*) za koje je određeno kojim aplikacijama pripadaju. Skeniranje portova vrši se s namjerom otkrivanja stanja portova, točnije otkrivanja dostupnih usluga određenog hosta. Port skeneri imaju vrlo važnu ulogu u mrežnoj sigurnosti. Mrežni administratori ih koriste kako bi otkrili ranjivosti sustava koje kasnije otklanjaju, a napadačima služe za otkrivanje ranjivosti koje kasnije zloupotrebljavaju.

Prije samog skeniranja portova i korištenja ostalih funkcionalnosti *nmap* programa, jednog od najpoznatijih port skenera, potrebno je razumjeti neke osnove *nmap* skeniranja. Potrebno je znati kako pravilno pokrenuti naredbu te se upoznati s načinima za odabir ciljanog/ciljanih računala. Nakon toga, vrlo je bitno upoznati se s metodama za otkrivanje domaćina kako bi se moglo odrediti koji domaćini su potencijalne mete za daljnje testiranje.

2.1. Sintaksa i korištenje

Nmap alat može biti instaliran i radi na svim poznatijim operacijskim sustavima. Za potrebe ovog završnog rada koristi se pomoću Kali Linux distribucije koja je zasnovana na *Debianu*. Ova distribucija, objavljena i održavana od strane *Offensive Security* organizacije, sadrži nekoliko stotina alata namijenjenih za provođenje raznih sigurnosnih testiranja.

Naredbe *nmap* programa se u Kali Linux-u upisuju u komandnu liniju. Svaka naredba *Network Mapper* programa započinje ključnom riječju *nmap* pomoću koje Kali pokreće *nmap* program. Nakon ključne riječi slijedi lista opcija i njihovih parametara koji određuju vrstu skeniranja i pobliže definiraju samo skeniranje. Lista argumenata može ostati prazna, odnosno nije potrebno navesti niti jednu opciju ili parametar za uspješno pokretanje naredbe. Ono što je ipak nužno navesti uz ključnu riječ jest ciljano računalo nad kojim se skeniranje provodi.

```
# nmap scanme.nmap.org

Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
Not shown: 995 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
646/tcp   filtered ldp
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 6.04 seconds
```

Sl. 2.1. Primjer ispravne *nmap* naredbe s praznom listom argumenata

Slika 2.1. prikazuje primjer jezgrovite *nmap* naredbe. Pomoću ključne riječi *nmap, Kali Linux* pokreće *Network Mapper*, a *"scanme.nmap.org"* je u ovom primjeru ciljano računalo. Prema [1] i [2] dozvoljeno je provesti nekoliko skeniranja dnevno nad tim računalom jer je postavljeno kako bi krajnji korisnici mogli legalno testirati alat na svojem računalu. Obzirom da je lista argumenata prazna, točnije da nije navedena niti jedna opcija koja određuje tip skeniranja, pokreće se podrazumijevani *(engl. defaultni)* sken.

```
# nmap -sS 45.33.32.156

Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
Not shown: 995 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
646/tcp   filtered ldp
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 27.03 seconds
```

Sl. 2.2. Primjer ispravne *nmap* naredbe s argumentom

Argumente *Network Mapper* alata je potrebno označiti crticom "-". Na slici 2.2. može se vidjeti primjer ispravne *nmap* naredbe s argumentom *–sS*. Argument *–sS* spada u opcije koje određuju tip skeniranja. Radi se o *TCP SYN* metodi za skeniranje portova koja je detaljnije opisana u poglavlju o skeniranju portova. Ciljano računalo u ovom primjeru je isto računalo kao i u primjeru sa slike 2.1., ali razlika je u tome što je na slici 2.1. računalo definirano imenom *hosta* (*hostname*) dok je na slici 2.2. definirano IP adresom.

2.1.1 Greške u sintaksi

Nakon ključne riječi *nmap*, u terminal *Kali Linux-a* moguće je napisati bilo što, pa tako i neispravne naredbe. U ovisnosti o tome kakva je greška počinjena, *nmap* će odgovoriti na odgovarajući način.

```
# nmap sS

Failed to resolve "sS".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.77 seconds
```

Sl. 2.3. Naredba bez definiranog ciljanog računala

Naredba na slici 2.3. je pogrešna. Sasvim je svejedno piše li *nmap sS* ili *nmap greška* jer *nmap* ne može odrediti značenje dijela "naredbe" *sS* kao argument zato što za to nedostaje crtica. Također, s obzirom da nije zadano ciljano računalo, se ne provodi nikakvo skeniranje.

```
# nmap sS 45.33.32.156
Failed to resolve "sS".
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
Not shown: 995 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
646/tcp   filtered ldp
9929/tcp  open     nping-echo
31337/tcp open     Elite

Nmap done: 1 IP address (1 host up) scanned in 30.95 seconds
```

Sl. 2.4. Naredba s greškom u sintaksi

Naredba sa slike 2.4. se razlikuje od naredbe sa slike 2.3. po tome što sadrži adresu ciljanog računala. Iz rezultata se vidi da je skeniranje provedeno iako naredba sadrži nepoznati dio *sS*. *Nmap* je samo ignorirao taj dio i nastavio dalje čitati naredbu.

```
# nmap -jč 45.33.32.156
Nmap: unrecognized option '-jč'
See the output of nmap -h for a summary of options.
```

Sl. 2.5. Naredba s nepostojećim argumentom

Ukoliko se u naredbi pojavi crtica (-), *nmap* očekuje da iza nje slijedi argument. Na slici 2.5. je prikazana naredba sa nepostojećim argumentom i ispravnom adresom ciljanog računala. Kada se

dogodi ovakva situacija, *nmap* javlja da ne prepoznaje argument (opciju) te se skeniranje ne provodi. Ispisuje se i poruka koja savjetuje korisnika da prouči sve opcije uz pomoć naredbe *nmap –h.*

2.2 Odabir ciljanog računala

Prema [1] *nmap* će sve što naredba sadrži, a da to nije opcija ili parametar opcije, tumačiti kao ciljano računalo. U dosadašnjim primjerima, u kojima je objašnjena sintaksa *nmap* programa, odredište je bilo jedno računalo (*host*) te je pokazano da postoji više od jednog načina na koji se taj odredišni *host* može definirati. Najjednostavniji način za skeniranje jednog odredišta je zadavanje istog IP adresom ili nazivom ciljanog računala (*hostname*). U općenitom zapisu to se može zapisati kao:

nmap [target]

2.2.1. Skeniranje "nekoliko" računala

Skeniranje dva, tri ili više računala vrši se vrlo slično kao i skeniranje jednog računala. Samo je potrebno navesti ciljana računala (IP adrese ili imena *hostova*) odvojene razmakom jedno pokraj drugog. U općenitom zapisu to bi izgledalo ovako: *nmap [target1 target2 target3 etc]*

2.2.2. CIDR notacija i rasponi

Nmap također podržava i *CIDR* (*Classless Inter-Domain Routing*) adresiranje prilikom zadavanja ciljanih računala što je vrlo korisno za skeniranje čitavih mreža i podmreža (subnet). Na taj način moguće je skenirati čitavu lokalnu mrežu naredbom *nmap 192.168.1.0/24*. Ta naredba ima isti učinak kao i naredba *nmap 192.168.1.100/24* zato što se u oba slučaja skeniraju sve IP adrese koje imaju prva 24 bita jednaka, točnije adrese od 192.168.1.0 do 192.168.1.255..

Prema [1] *CIDR* adresiranje nije dovoljno fleksibilno, ali *nmap* podržava i adresiranje na razini okteta pa je moguće skenirati raspon IP adresa i na sljedeće načine:

- *nmap 192.168.10.1-200* - skeniraju se sve adrese od 192.168.10.1 do 192.168.10.200.
- *nmap 192.168.1-100.** - skeniraju se sve adrese od 192.168.1.0 do 192.168.100.255, treći broj adrese je određen rasponom od 1-100, a četvrti zvjezdicom koja je zamjena za 0-255

2.2.3. Lista odredišnih računala

Uz pomoć argumenta *–iL* može se skenirati lista *hostova*. U toj listi se može koristiti bilo koji od do sada prikazanih načina za odabir ciljanih računala (IP adrese, naziv *hosta, CIDR*, oktet nekog područja), ali se svaki unos mora odvojiti razmakom, *tabom* ili novim redom. Općenito:
nmap –iL [list.txt]

2.2.4. Izostavljanje hostova iz skeniranja

Ako je potrebno "preskočiti" neka računala, moguće je to učiniti ovako:

- *nmap 192.168.1.0/24 --exclude 192.168.1.10, osjetljivi-host* – Skeniraju se sve adrese od 192.168.1.0 do 192.168.1.255 osim adrese 192.168.1.10 i računala osjetljivi-host.

- *nmap 192.168.1.0/24 --excludefile list.txt* – Skeniraju se sve adrese od 192.168.1.0 do 192.168.1.255 osim adresa navedenih u datoteci *list.txt*.

2.3 Otkrivanje domaćina

Prikupljanje informacija je prva faza testiranja računalne mreže. Mreže koje je potrebno skenirati i testirati mogu imati jako puno IP adresa, od kojih je nerijetko većina neaktivna, te je stoga potrebno suziti taj popis na popis adresa zanimljivih domaćina. Kriterije koji čine domaćina zanimljivim određuje osoba koja vrši testiranje na temelju razloga skeniranja.

Otkrivanje domaćina(*ping scanning*) omogućava da se otkriju "budni" *hostovi* nad kojima se onda kasnije vrši skeniranje portova(*port scanning*) ili neka od ostalih mogućnosti *nmap* skeniranja. Samo ime *ping scanning* može izazvati zabunu. Metode za otkrivanje domaćina ne koriste samo jednostavne *ICMP echo request* pakete na koje podsjeća riječ *ping*.

Valja napomenuti da *nmap* svakako pokreće *ping* skeniranje, osim ako mu nije eksplicitno naglašeno da to ne radi, prije nego što izvrši skeniranje portova ili bilo koji drugi "agresivniji" sken. Na taj način se izbjegava zahtjevnije skeniranje nedostupnih odredišnih računala te se štede vrijeme i resursi.

Nmap nudi razne opcije s kojima omogućuje više tehnika otkrivanja domaćina, a neke od tih tehnika biti će detaljnije objašnjene u ovom poglavlju.

2.3.1. List scan (-sL)

List scan služi za potvrđivanje odredišnih *hostova* koji će biti skenirani. Ovaj sken zapravo ne šalje nikakve pakete odredišnim *hostovima*, nego samo ispisuje svako računalo na mreži i vrši *reverse-DNS* postupak nad njima. Također se dobije uvid o broju IP adresa koje su na toj mreži. Za pokretanje list skena potrebno je u listi argumenata navesti argument –sL.

```
felix~> nmap -sL www.stanford.edu/28

Starting Nmap ( http://nmap.org )
Host www9.Stanford.EDU (171.67.16.80) not scanned
Host www10.Stanford.EDU (171.67.16.81) not scanned
Host scriptorium.Stanford.EDU (171.67.16.82) not scanned
Host coursework-a.Stanford.EDU (171.67.16.83) not scanned
Host coursework-e.Stanford.EDU (171.67.16.84) not scanned
Host www3.Stanford.EDU (171.67.16.85) not scanned
Host leland-dev.Stanford.EDU (171.67.16.86) not scanned
Host coursework-preprod.Stanford.EDU (171.67.16.87) not scanned
Host stanfordwho-dev.Stanford.EDU (171.67.16.88) not scanned
Host workgroup-dev.Stanford.EDU (171.67.16.89) not scanned
Host courseworkbeta.Stanford.EDU (171.67.16.90) not scanned
Host www4.          EDU (171.67.16.91) not scanned
Host coursework-i.Stanford.EDU (171.67.16.92) not scanned
Host leland2.Stanford.EDU (171.67.16.93) not scanned
Host coursework-j.Stanford.EDU (171.67.16.94) not scanned
Host 171.67.16.95 not scanned
Nmap done: 16 IP addresses (0 hosts up) scanned in 0.38 seconds
```

Sl. 2.6. *List scan* proveden nad "*stanford.edu* " [1]

Slika 2.6. prikazuje rezultat list skena provedenog nad mrežom koja okružuje glavni web server Sveučilišta Stanford. *CIDR /28* notacijom se pretraživalo 16 *hostova* i ispisane su njihove IP adrese i imena (*hostname*).

Prema[3] čak i samo ime *hosta* može otkriti zanimljive informacije o IP adresama. Iz nekih imena se može vidjeti za što se *host* koristi i gdje se nalazi (geografski).

2.3.2. Ping scan (-sP)

Opcija –sP pokreće *ping scan*, a kao rezultat ispisuje dostupne *hostove*. Dostupnim *hostovima* se smatraju svi oni koji su odgovorili na skeniranje. Po zadanom, ovom opcijom se šalje *ICMP echo request* i *TCP ACK* paket na port 80 odredišta. Kada je opcija pokrenuta od stane neprivilegiranog korisnika, tada se šalje *TCP SYN* paket na port 80. Kada se skeniraju *hostovi* na lokalnoj mreži koriste se *ARP* zahtjevi, a kao rezultat se dobiju i MAC adrese uređaja na lokalnoj mreži. Opciju *–sP* je moguće kombinirati sa većinom skenova za otkrivanje domaćina.

2.3.3. Sken bez pinganja (-PN)

Pomoću opcije –PN eksplicitno se naglašava da se preskače faza otkrivanja *hostova.* Skeniranje portova, otkrivanje verzije ili operacijskog sustava i ostali agresivniji skenovi se obično provode nad *hostovima* za koje se zna da su "budni". Ova opcija se koristi ako je potrebno skenirati sve *hostove*, uključujući i one koji su prikazani kao neaktivni. Ponekad je *host* jako dobro zaštićen i naizgled neaktivan jer ne odgovara na skenove za otkrivanje domaćina, ali to ne znači da nad njim nije moguće primijeniti skeniranje portova uz odabir pravih opcija. Ovakvim načinom skeniranja, skeniranje može potrajati znatno duže nego obično. Prema[1] može se činiti beskorisnim slati pakete na IP adrese koje vjerojatno nemaju *host* koji sluša, ali ozbiljni penetracijski testeri su voljni platiti cijenu kako bi izbjegli rizik da im aktivno računalo prođe neopaženo.

```
$ nmap 10.10.5.11

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-13 08:43 CDT
Note: Host seems down. If it is really up, but blocking our ping
probes, try -PN
Nmap done: 1 IP address (0 hosts up) scanned in 3.16 seconds
```

Sl. 2.7. *Host* neaktivan ili odbija *pinganje* [3]

Na slikama 2.7. i 2.8. je prikazano točno ono što je opisano u ovom potpoglavlju. Pokušaj skeniranja odredišta je prvotno prošao neuspješno zato što odredišni *host* blokira *pinganje*. Korištenjem opcije *–PN, pinganje* je preskočeno i odredišni *host* je skeniran podrazumijevanim *(engl. defaultnim)* port skenom. Do situacija kao što je ova dolazi kada je vatrozid (*firewall*) postavljen tako da blokira pinganje.

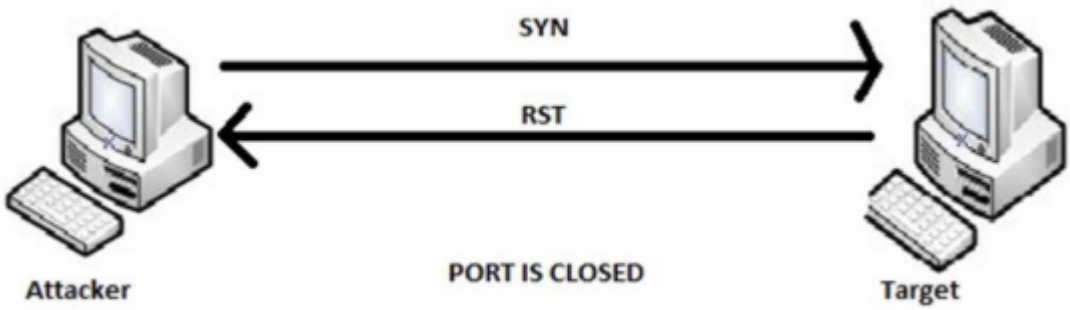
```
$ nmap -PN 10.10.5.11

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-13 08:43 CDT
Interesting ports on 10.10.5.11:
Not shown: 999 filtered ports
PORT     STATE  SERVICE
3389/tcp open   ms-term-serv

Nmap done: 1 IP address (1 host up) scanned in 6.51 seconds
```

Sl. 2.8. *Host* je aktivan i moguće ga je skenirati [3]

2.3.4. TCP SYN Ping (-PS<port list>)

TCP SYN Ping sken je tehnika za skeniranje sustava koji blokiraju standardne *ICMP pingove*. Navođenjem *–PS* opcije šalje se prazan *TCP* paket sa postavljenom *SYN* zastavicom. Zadani *(engl. defaultni)* odredišni port je port 80, ali mogu se navesti i drugi portovi u listi parametara pa se onda *TCP* paket šalje paralelno svim tim portovima. Na slici 2.9. prikazana je komunikacija s računalom koje ima otvoren odabrani *port*. Ako je port otvoren, odredišni *host* odgovara sa *SYN/ACK TCP* paketom. Nakon toga računalo s kojeg je pokrenut sken prekida vezu tako što odgovori sa *RST* paketom. Slikom 2.10. prikazano je kako to izgleda kada je odbrani port zatvoren. Ako je port zatvoren, odredišni *host* šalje nazad *TCP* paket sa *RST* zastavicom.

Sl. 2.9. *TCP SYN Ping* sken domaćina sa otvorenim odabranim portom[4]

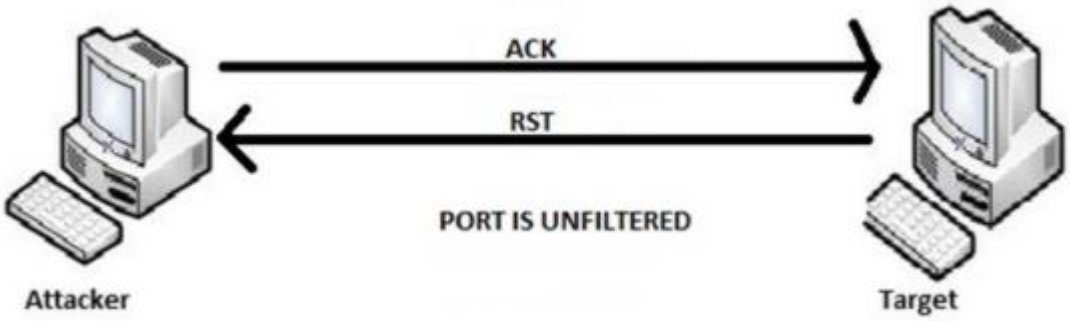


Sl. 2.10. *TCP SYN Ping* sken domaćina sa zatvorenim odabranim portom[4]

Ova metoda skeniranja zapravo koristi korake metode za uspostavljanje konekcije koja se naziva *"Three-way-handshake"*. Obzirom da je *TCP SYN Ping* metoda za otkrivanje domaćina, nije bitno je li odredišni *host* odgovorio sa *RST* ili *SYN/ACK*. Bitno je samo je li odgovor poslan ili ne. Ako je poslan to znači da je *host* aktivan, a ako nikakav odgovor nije primljen to se tumači kao da je odredište neaktivno ili da je zaštita blokirala skeniranje.

2.3.5. TCP ACK Ping (-PA<port list>)

TCP ACK Ping sken se razlikuje od *TCP SYN Ping* skena po tome što koristi *TCP ACK* zastavicu kao što i samo ime kaže. Odredišni *host* primanjem paketa s tom zastavicom zapravo prima informaciju o potvrđenoj konekciji, ali ta konekcija nije niti bila započeta. Obzirom da je veza nepostojeća, ukoliko je odredište aktivno, poslati će *RST* paket kao odgovor kao što je prikazano na slici 2.11. Ova opcija se također može koristiti kada su *ICMP pingovi* blokirani.

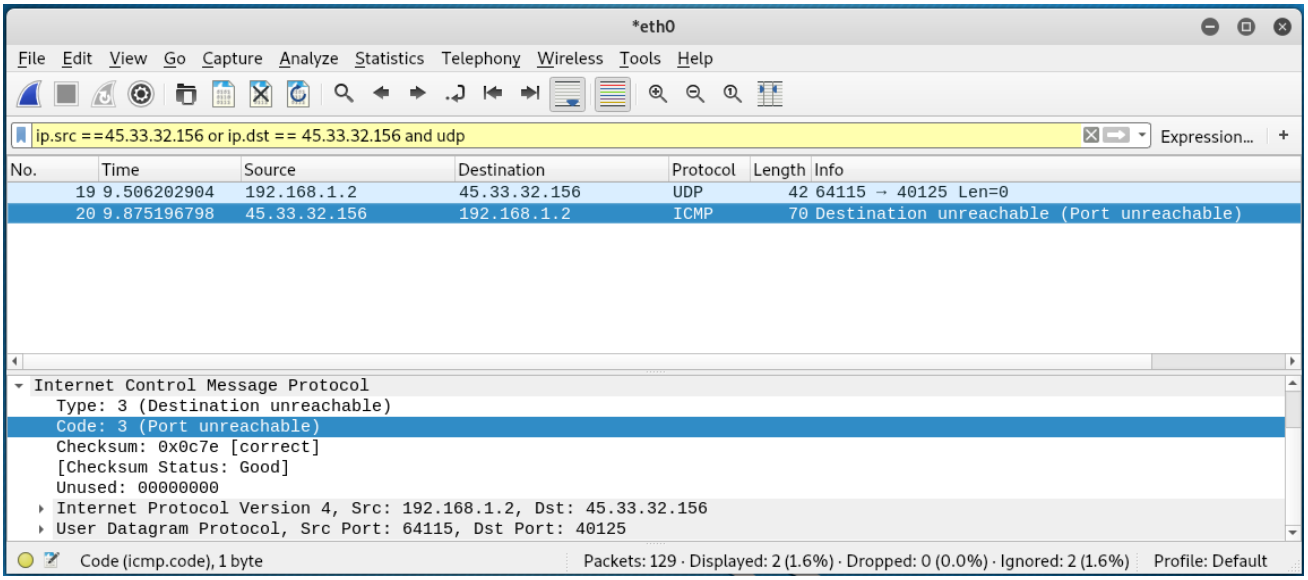


Sl.2.11. *TCP ACK Ping* sken aktivnog domaćina.[4]

Glavni razlog zbog kojeg *nmap* nudi i ovu opciju, vrlo sličnu *TCP SYN Ping* skenu, je taj što se na taj način povećavaju šanse za zaobilaženje vatrozida (*firewalls*). U ovisnosti o tome kako je vatrozid konfiguriran, određeni skenovi će dati željene rezultate, dok određeni neće.

2.3.6. UDP Ping (-PU<port list>)

UDP Ping je još jedna od opcija za otkrivanje domaćina. Ova opcija šalje prazan UDP paket željenim portovima odredišta. Po zadanom *(engl. defaultu)* je postavljen *port* 40125. Kod ovog skena poželjno je koristiti portove koji se ne koriste jer ukoliko je paket stigao otvorenom portu, odredišni host će ga ignorirati obzirom da je prazan. Kada prazni *UDP* paket stigne odredištu, a definirani port odredišnog hosta je zatvoren, odredišni *host* odgovara sa *ICMP* porukom koja javlja da je port nedostupan. Takav odgovor zapravo znači da je *host* aktivan. Ako *host* nije aktivan, najčešći odgovori su ostale *ICMP* greške ili jednostavno odredišni host ne pošalje nikakav odgovor.



Sl. 2.12. *UDP Ping* metoda prikazana kroz razmjenu paketa pomoću *Wiresharka*

Slika 2.12. prikazuje *UDP Ping scan* ip adrese 45.33.32.156. (*scanme.nmap.org*) u *Wiresharku*, alatu za analizu mrežnih paketa. Paket broj 19 je paket koji šalje računalo koje pokreće sken. Radi se o praznom UDP paketu koji se šalje portu 40125 odredišta. Port 64115 je nasumično odabran port koji očekuje odgovor. Paket broj 20 je paket koji je odredišni host poslao na port 64115. Radi se o ICMP paketu tipa 3, koda 3 koji je zapravo poruka da je port nedostupan što ujedno znači i da je odredišni *host* aktivan.

Ovom metodom se mogu otkriti *hostovi* čiji vatrozid "filtrira" sve *TCP* portove, ali i dalje odgovaraju na *UDP* pakete kako je opisano u ovom potpoglavlju. Prema[3] većina vatrozida, osim onih loše konfiguriranih, ipak blokira ovaj način uspostavljanja veze.

2.3.7. ICMP Echo Ping (-PE)

Ovom opcijom se odredištu šalje *ICMP* tip 8 paket (*echo request packet*). Ako je odredište aktivno, očekuje se *ICMP* tip 0 paket (*echo replay*). Prema[1] u današnje vrijeme većina vatrozida blokira ove pakete te su skenovi koji koriste samo *ICMP* rijetko dovoljno pouzdani kod nepoznatih odredišta, ali mogu biti korisni mrežnim i sistemskim administratorima za nadgledanje lokalne mreže.

2.3.8. ARP Scan (-PR)

Ovaj tip otkrivanja domaćina se koristi isključivo na lokalnoj mreži. Puno je brži od ostalih metoda opisanih do sada. [1]

```
# nmap -sn -PR 192.168.1.0/24
Nmap scan report for 192.168.1.1
Host is up (0.010s latency).
MAC Address: **:**:**:**:**:** (Unknown)
Nmap scan report for 192.168.1.4
Host is up (0.00023s latency).
MAC Address: 08:00:27:5C:21:E3 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.6
Host is up (0.059s latency).
MAC Address: D4:6A:6A:E1:76:47 (Hon Hai Precision Ind.)
Nmap scan report for 192.168.1.7
Host is up (0.00058s latency).
MAC Address: 08:00:27:DE:4E:19 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.8
Host is up (0.00036s latency).
MAC Address: **:**:**:**:**:** (Asustek Computer)
Nmap scan report for 192.168.1.2 (Apple)
Host is up.
Nmap done : 256 IP addresses (6 hosts up) scanned in 2.33 seconds
```

Sl. 2.13. Rezultati *ARP* skena

Slika 2.13. prikazuje rezultate *ARP* skena za otkrivanje domaćina. Obzirom da je sken pokrenut na lokalnoj mreži *ARP* sken bi bio proveden i da je u naredbi bio izostavljen argument –PR. U rezultatima primjera se vidi da je aktivno 6 *hostova* te su također ispisane njihove *MAC* adrese.

3. SKENIRANJE PORTOVA

Nakon provođenja skenova za otkrivanje domaćina, osoba koja provodi testiranje ima popis adresa nad kojima vrši daljnje testiranje. *Nmap* se u svojim počecima koristio isključivo kao port skener. Kroz godine je opseg funkcionalnosti proširen, ali je skeniranje portova i dalje glavna mogućnost ovoga programa.

Skeniranje portova je zapravo testiranje portova udaljenog računala, kako bi se odredilo u kojem su stanju. U ovom poglavlju biti će objašnjeno koja su to stanja te će se detaljnije opisati neke od metoda skeniranja portova.

Skeniranje portova je vrlo važno za mrežnu sigurnost. Prema[1] smanjivanje broja i složenosti pruženih usluga smanjuje mogućnosti za potencijalne napadače koji pokušavaju provaliti te zbog toga mrežni administratori trebaju redovno skenirati mrežu za koju su zaduženi pomoću alata kao što je *Nmap*. Dostupne trebaju ostati jedino usluge koje se koriste. Također je potrebno i postaviti pravila vatrozida kako bi se pristup dozvolio samo pouzdanim korisnicima.

Najjednostavnija naredba *nmap [target]* skenira 1000 najpoznatijih *TCP* portova odredišnog *hosta* i kategorizira ih u jednu od 6 kategorija.

3.1. Stanja portova

Nmap prepoznaje 6 stanja portova: *open, closed, filtered, unfiltered, open|filtered, closed|filtered*. Važno je napomenuti da prema [1] ova stanja nisu unutrašnja stanja portova, nego samo opisuju kako ih *nmap* vidi. Različite metode skeniranja svrstavaju portove u različite kategorije zbog utjecaja same konfiguracije mreže koja se skenira, vatrozida i ostalih mehanizama zaštite. U nastavku teksta opisano je svih 6 kategorija (stanja):

- Ako je port otvoren (*open*) to znači da aplikacija aktivno prihvaća TCP i/ili UDP pakete koji se šalju uz pomoć *nmapa*. Otvoreni portovi prikazuju koje su sve usluge dostupne na mreži. Zadaća mrežnih administratora je da drže otvorene portove pod kontrolom i zaštite ih od napadača.

- Port koji je dostupan, ali na njemu ne sluša niti jedna aplikacija je zatvoreni port.

- Ako su upiti prema portu blokirani putem vatrozida, pravila na *routeru* ili na neki drugi način tada *nmap* ne može utvrditi je li port otvoren ili ne. Takav port *nmap* označi kao filtriran (*filtered*). Filtrirani portovi pružaju vrlo malo informacija onome tko vrši skeniranje.

- Nefiltrirani (*unfiltered*) stanje znači da je port dostupan, ali *nmap* ne može odrediti je li otvoren ili zatvoren.

- Kada *nmap* ne može odrediti je li port otvoren ili filtriran tada ga svrstava u otvoren|filtriran (*open*|*filtered*) stanje. Do te situacije dolazi ako na upit ne dobije nikakav odgovor.

- Stanje zatvoren|filtriran (*closed*|*filtered*) pojavljuje se kada *nmap* ne može odrediti je li port zatvoren ili filtriran.

3.2. Opcije za određivanje portova

Ukoliko se ne koriste opcije za određivanje portova, nmap će skenirati portove od 0 do 1023 (*well known ports*).

3.2.1. Opcija –p <port ranges>

Opcijom –p se skeniraju samo portovi koji se navedu. Portove je uz ovu opciju moguće navesti na nekoliko načina:

- Pojedinačne portove moguće je navesti jedne do drugih odvojene zarezom. Naredba *nmap –p 25,80 target* skenira samo portove 25 i 80 odredišta.
- Raspon portova moguće je skenirati pomoću crtice "-" koja odvaja donju i gornju granicu raspona. Naredba *nmap –p 10-20 target* skenira portove od 10 do 20 odredišta.
- Za skeniranje samo *TCP* portova moguće je navesti identifikator "T:". Isto tako moguće je za *UDP* portove navesti identifikator "U:". Ako se koristi ovaj način odabiranja portova, potrebno je navesti i opciju –sU i jednu od opcija za *TCP* tip skeniranja jer će u suprotnom portovi biti skenirani i po *TCP* i po *UDP* protokolu.

- Portove je također moguće definirati imenima koja su navedena u datoteci *nmap-services*. Također se može koristiti i znak "*" koji označava "sve". Skeniranje svih portova čije ime počinje sa "http" moguće je provesti naredbom *nmap –p http* target*.

3.2.2. Preskakanje portova (--exclude-ports <port ranges>)

Pomoću opcije *--exclude-ports <port ranges>* moguće je preskočiti određene portove. Portovi se navode kao i u opciji "-p". Korištenjem ove opcije navedeni portovi će biti preskočeni i prilikom otkrivanja domaćina.

3.2.3. Brzi sken

Ovom opcijom se naglašava da se skenira manje od podrazumijevanih *(engl. defaultnih)* 1000. Opcija –F služi za skeniranje 100 najčešće korištenih portova. Podaci o učestalosti korištenja portova mogu se pronaći u datoteci "*nmap-services*".

3.3. TCP SYN scan (-sS)

Ovaj sken za otkrivanje portova je podrazumijevani *(engl. defaultni)* sken ukoliko skeniranje pokreće privilegirani korisnik ("root" na Unix/Linux operacijskom sustavu ili "Administrator na Windowsu). Prema [1] provodi se vrlo brzo i poprilično je nenametljiv.

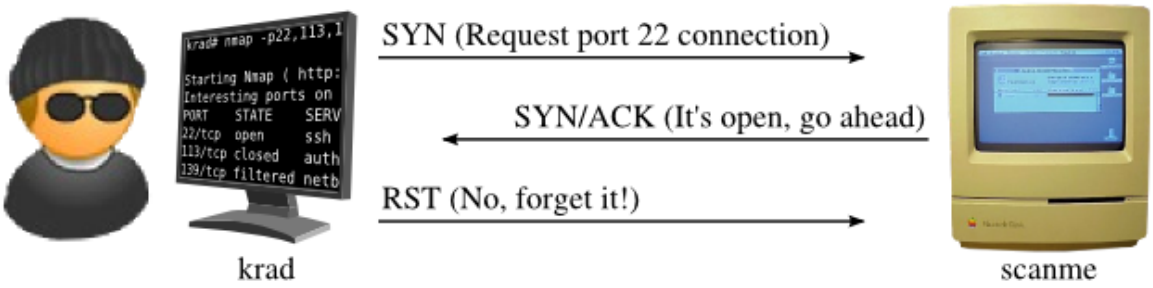
```
krad# nmap -p22,113,139 scanme.nmap.org

Starting Nmap ( http://nmap.org )
Interesting ports on scanme.nmap.org (64.13.134.52):
PORT    STATE    SERVICE
22/tcp  open     ssh
113/tcp closed   auth
139/tcp filtered netbios-ssn

Nmap done: 1 IP address (1 host up) scanned in 1.35 seconds
```

Sl. 3.1. Rezultati *TCP SYN* skena nad "scanme.nmap.org" [1]

Slika 3.1. prikazuje rezultat *TCP SYN* skena provedenog nad portovima 22,113 i 139 *hosta* "*scanme.nmap.org"*. Port 22 je otvoren, 113 zatvoren, a 139 filtriran. U nastavku je objašnjeno na koji način su određena stanja portova u ovom primjeru.

3.3.1. Otvoreni port

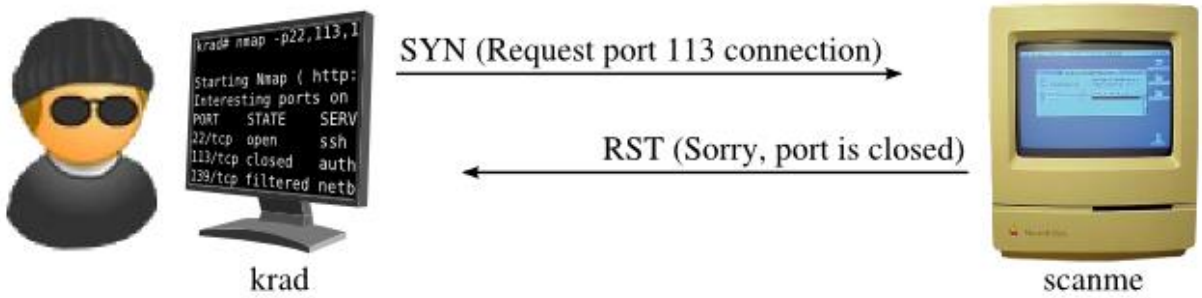
Nmap započinje *TCP SYN* sken slanjem *TCP* paketa sa postavljenom *SYN* zastavicom svim portovima koji su navedeni (ako nisu navedeni, po zadanom se šalje dobro poznatim portovima).



Sl. 3.2. *TCP SYN* sken otvorenog porta 22 [2]

Slika 3.2. prikazuje kako izgleda komunikacija kada je port 22 otvoren. Postupak započinje tako što "krad" šalje *TCP* paket sa postavljenom *SYN* zastavicom na port 22 računalu "scanme". Obzirom da je port 22 otvoren, računalo "scanme" odgovara *TCP* paketom sa postavljenim *SYN* i *ACK* zastavicama. Kada "krad" primi ovaj paket, saznaje da je port 22 otvoren i u tom trenutku šalje *RST* paket računalu "scanme". *RST* paket je potrebno poslati jer bi u suprotnom računalo "scanme" pretpostavilo da je paket *SYN/ACK* izgubljen i nastavilo bi ga iznova slati "kradu". Postupak je prikazan na portu 22, ali je isti za sve otvorene portove.

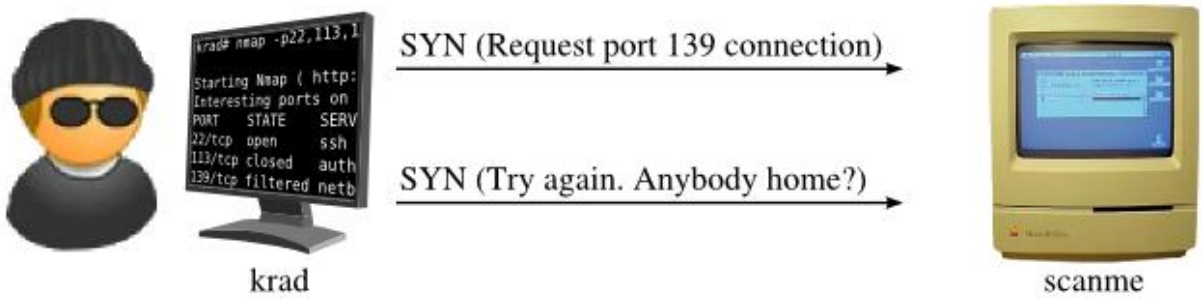
3.3.2. Zatvoreni port



Sl. 3.3. TCP SYN sken zatvorenog porta 113 [2]

Slika 3.3. prikazuje komunikaciju u slučaju da je port zatvoren. Kada računalo "scanme" primi *TCP SYN* paket na port 113 i zaključi da je taj port zatvoren, nazad šalje *RST* paket. *RST* paket "kradu" daje do znanja da je port 113 zatvoren i daljnja komunikacija nije potrebna.

3.3.3. Filtrirani port



Sl. 3.4. *TCP SYN* sken filtriranog porta 139 [2]

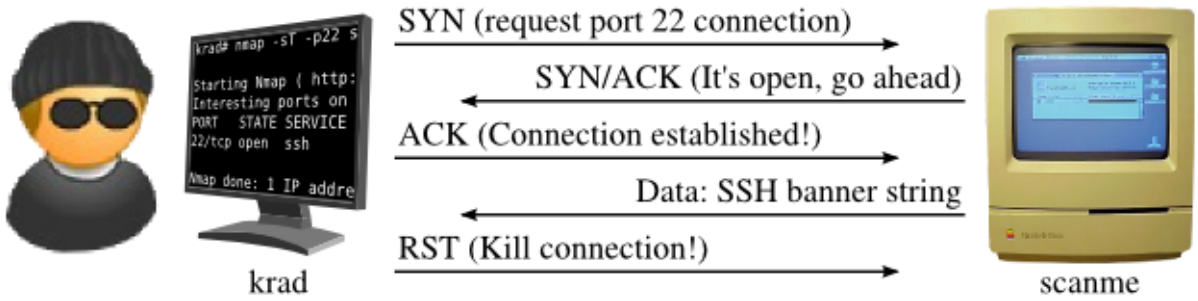
Slika 3.4. pokazuje slučaj u kojem *nmap* tumači port kao filtriran prilikom *TCP SYN* skeniranja. Računalo "krad" šalje *TCP SYN* paket računalu "scanme" na port 139 i ne prima nikakav odgovor. Obzirom da je moguće da je paket izgubljen, "krad" ponovno šalje isti paket. Ovaj proces se može ponavljati nekoliko puta u ovisnosti o tome jesu li ostali portovi poslali svoje odgovore. *Nmap* vodi statistiku izgubljenih paketa i pokušati će više puta poslati paket ako je mreža nestabilnija. U slučaju da "krad" primi *ICMP* poruku greške od "scanme" vezano za port 139 također će zaključiti da je port filtriran.

3.4. TCP Connect Scan (-sT)

TCP connect sken je podrazumijevani *(engl. defaultni)* sken korisnika bez privilegija za rad sa "*raw paketima*" (*raw packet privileges*) ili prilikom skeniranja IPv6 mreža. Ova metoda radi na način da se uspostavi konekcija sa odredištem putem sistemskog poziva *connect*.

Prema [1] *TCP SYN* sken je najčešće bolja opcija od *TCP Connect* skena zato što *nmap* ima veću kontrolu nad radom sa "sirovim" paketima (*raw packets*), nego nad sistemskim pozivima visoke razine. *TCP Connect* sken je također sporiji i lakše ga je otkriti.

Razmjena paketa prilikom skeniranja zatvorenih i filtriranih portova je jednaka kao i kod *TCP SYN* skena. Komunikacija prilikom skeniranja otvorenih portova zahtjeva razmjenu više paketa nego kod *TCP* skena.



Sl. 3.5. *TCP Connect* sken otvorenog porta 22 [2]

Na slici 3.5. je prikazana razmjena paketa prilikom provođenja *TCP Connect* skena nad otvorenim portom 22 računala "scanme". Komunikacija započinje tako što "krad" pošalje *TCP* SYN paket na port 22 od "scanme". Računalo "scanme" odgovori sa *SYN/ACK* paketom. Zatim "krad", umjesto da samo prekine konekciju, potvrđuje *SYN/ACK* paket slanjem *ACK* paketa računalu "scanme". Ovim načinom uspostavljena je konekcija između ta dva računala. Konekcija se prekida slanjem *RST* paketa čim operacijski sustav potvrdi nmap-u da je uspostavljena. Na slici je prikazan primjer u kojem "scanme" pošalje *SSH banner* kroz tu konekciju prije nego li je konekcija prekinuta.

3.5. UDP Scan (-sU)

Opcijom –sU većini ciljanih portova se šalju prazni *UDP* paketi. Nekima od dobro poznatih portova se ipak šalju paketi sa specifičnim sadržajem. Prema [1] *UDP* portovi su često zanemareni u odnosu na *TCP* portove pošto je *UDP* skeniranje sporije i teže od *TCP* skeniranja.

Tab. 3.1. Tablica stanja portova obzirom na odgovore odredišnog *hosta* na *UDP Scan*[2]

Odgovor odredišnog hosta	Stanje porta
UDP odgovor (neuobičajeno)	otvoren
Nema nikakvog odgovora (nakon nekoliko pokušaja)	otvoren|filtriran
ICMP greška nedostupnog porta (tip 3, kod 3)	zatvoren
Ostale ICMP greške nedostupnosti (tip 3, kod 1, 2, 9, 10, 13)	filtriran

Tablicom 3.1. prikazano je kako *nmap* tumači pojedine odgovore odredišta. *UDP* odgovor nije uobičajen pošto otvoreni *UDP* portovi uglavnom ne odgovaraju na prazne pakete. Ovaj slučaj je vjerojatniji ako se radi o portu kojem se šalje paket sa specifičnim sadržajem za protokol na tom portu. Ako je odgovor *ICMP* port nedostupan (tip 3, kod 3) greška, port je zatvoren. Ostale

ICMP greške označavaju da je port filtriran. Prema[1] najzanimljiviji scenarij je ipak kada nema nikakvog odgovora. U tom slučaju port je određen kao otvoren|filtriran.
Problem stvaraju vatrozidi i mehanizmi za filtriranje mrežnog prometa koji potencijalno odbacuju primljene pakete i ne šalju nikakav odgovor te se zbog toga ne može odrediti je li *port* otvoren ili filtriran.

3.6. TCP FIN, NULL, Xmas Scans (-sF, -sN, -sX)

U *TCP RFC* standardu postoji propust zbog kojeg je moguće odrediti je li port otvoren ili zatvoren. *TCP FIN, Null i Xmas* koriste taj propust.

Standardom je definirano ponašanje zatvorenog porta na način da, ukoliko je primio segment bez *RST* zastavice, odgovori sa *RST* paketom. Također je u istom standardu definirano ponašanje otvorenog porta koji je primio paket bez postavljenih *SYN, RST i ACK* zastavica da samo ispusti paket.

Ova pravila omogućavaju da skenovi uz pomoć paketa koji nemaju postavljene *SYN, RST i ACK* zastavice odrede stanja portova odredišnog *hosta*.

Null sken (-sN) je sken uz pomoć paketa koji sadrži sve zastavice postavljene na 0, *FIN* sken (-sF) ima postavljenu samo *FIN* zastavicu na 1, a *Xmas* sken (-sX) koristi paket sa *FIN, PSH i URG* zastavicom.

Tab. 3.2. Tablica stanja portova obzirom na odgovore na *TCP FIN, Null ili Xmas Scan*[2]

Odgovor odredišnog hosta	Stanje porta
Nema nikakvog odgovora (nakon nekoliko pokušaja)	otvoren|filtriran
TCP RST paket	zatvoren
Ostale ICMP greške nedostupnosti (tip 3, kod 1, 2, 3, 9, 10, 13)	filtriran

Sva 3 skena rade na istom principu i tumače odgovore na isti način prikazan tablicom 3.2.. Ako nikakav odgovor nije primljen, port se kategorizira kao otvoren|filtriran. Ako je odgovor *TCP RST* paket, port se smatra zatvorenim. Filtiran port je onaj za koji se kao odgovor dobije *ICMP* greška tipa 3.

Prema [1] prednost ovih skenova je ta što su "tiši" čak i od *TCP SYN* skena. Mogu čak zaobići neke vatrozide i filtere mrežnih paketa. Mana ovih skenova bi bila ta što se ne drže svi sustavi *RFC 793* standarda u potpunosti. Mnogi sustavi odgovaraju sa *RST* paketima bez obzira je li port otvoren ili ne. Problem kod ovih skenova je i što ne mogu sa sigurnošću odrediti otvorene portove.

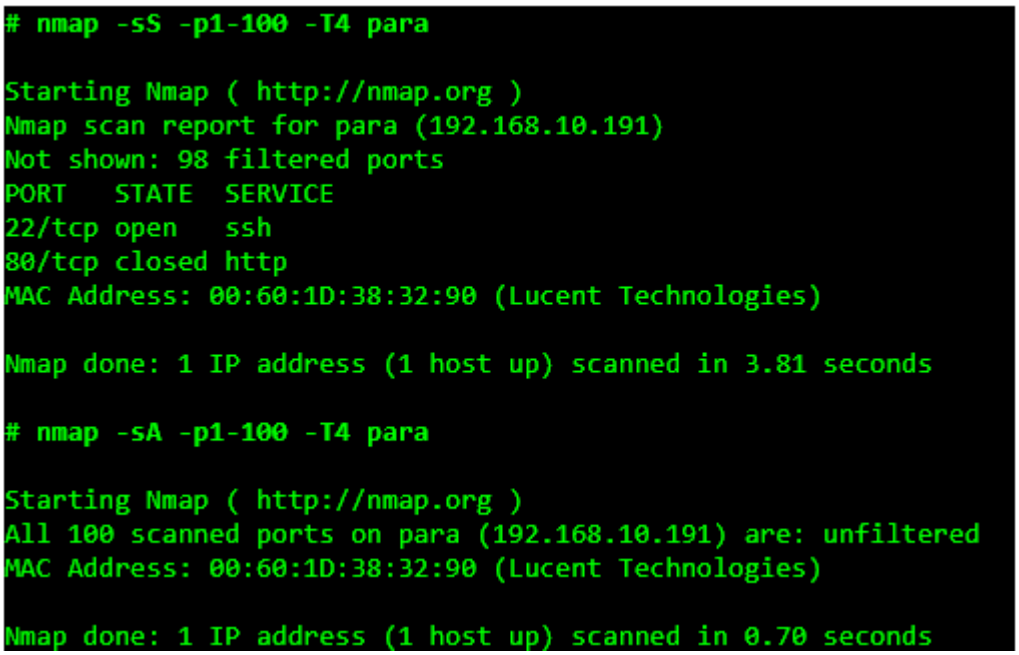
Pomoću opcije *--scanflags* moguće je definirati vlastite skenove. Ova opcija omogućuje korisniku postavljanje željenih zastavica paketa.

3.7. TCP ACK Scan (-sA)

TCP ACK sken je jedinstven po tome što nikada ne određuje portove kao otvorene niti otvorene|filtrirane. Koristan je za određivanje o kakvom vatrozidu je riječ i koji portovi su filtrirani. Za skeniranje se koriste paketi sa postavljenom *ACK* zastavicom.

Kod skeniranja sustava koji ne filtriraju *ACK* pakete, otvoreni i zatvoreni portovi vraćaju *RST* paket, a portovi koji ne odgovaraju ili vraćaju *ICMP* poruke greške su označeni kao filtrirani. Obzirom da i otvoreni i zatvoreni portovi odgovaraju na isti način, ne može se odrediti u kojem je točno stanju taj port ovom metodom skeniranja.

Prema [2] *nmap ACK* sken često pokazuje manje filtriranih portova od *SYN* skena provedenog na istom odredišnom *hostu* zato što je *ACK* skenove teže filtrirati. Često se mrežama dozvoljavaju odlazne konekcije, ali se žele blokirati nadolazeće. Jednostavan način za postizanje toga je da se blokiraju samo nadolazeći *SYN* paketi. Blokiranje *ACK* paketa od nepouzdanih izvora i istovremeno propuštanje *ACK* paketa pouzdanih izvora zahtjeva da *host* ima konfiguriran *stateful* vatrozid. Takav vatrozid nadgleda potpuno stanje aktivnih mrežnih konekcija na način da konstantno analizira cijeli kontekst mrežnog prometa i vrši proces odobravanja i blokiranja.



Sl. 3.6. Usporedba *SYN* i *ACK* skena na računalu koje ima *stateless* vatrozid[2]

Na slici 3.6. se može vidjeti usporedba *SYN* i *ACK* skena provedenih nad istim računalom. *SYN* sken je prikazao 98 portova kao filtrirane te 1 otvoren i 1 zatvoren port. *ACK* sken je odredio sve portove kao nefiltrirane što znači da *TCP ACK* paketi prolaze kroz zaštitu *hosta* "para" i odgovaraju sa *RST* paketima. Iz ovih rezultata da se zaključiti da vatrozid odredišnog *hosta* nije *stateful* nego *stateless.*

3.8. TCP Idle Scan (-sI)

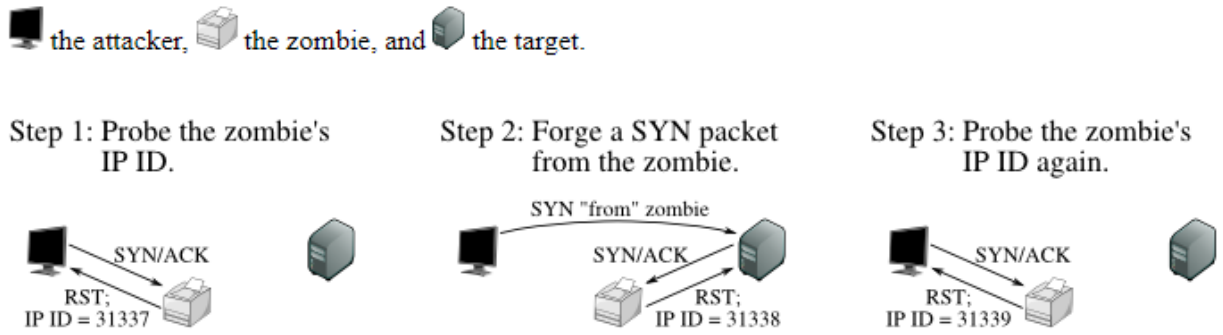
Idle sken je port sken koji omogućava skeniranje odredišnog *hosta* bez slanja ijednog paketa sa svoje IP adrese. Za izvođenje ovog skena potrebno je "zombi" računalo, a ovaj sken je moguće izvesti zato što:

- Ako se odredištu pošalje *TCP SYN* paket na port, otvoreni port odgovora sa *SYN/ACK*, a zatvoreni sa *RST*.
- Računalo koje primi *SYN/ACK* paket koji nije očekivalo odgovara sa *RST* paketom. Neželjeni *RST* paket se ignorira.
- Svaki IP paket sadrži identifikacijski broj fragmenta (*IP ID*)

Koraci *idle* skena su sljedeći:

1. Ispitivanje *IP ID*-a zombi računala
2. Stvaranje *SYN* paketa zombi računala i slanje tog paketa na željeni port mete.
3. Ponovno ispitivanje *IP ID*-a zombi računala i određivanje stanje porta na temelju istog.

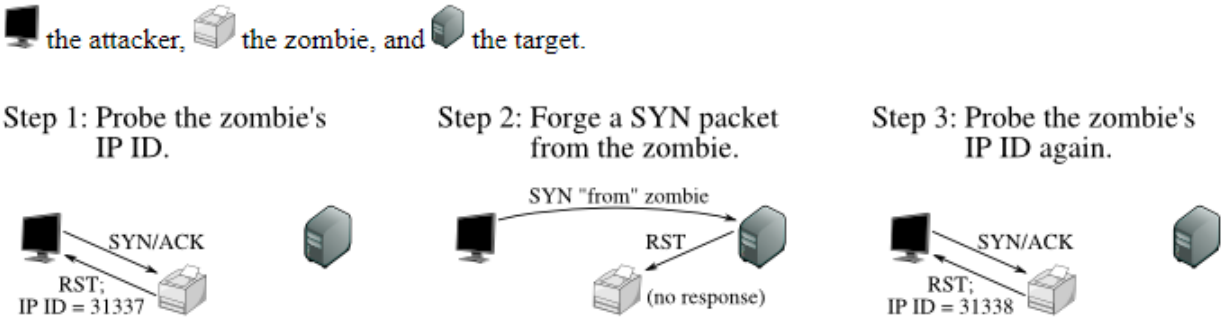
3.8.1. Idle Scan otvorenog porta



Sl. 3.7. Postupak *idle* skena otvorenog porta[2]

Na slici 3.7. prikazan je postupak *idle* skena otvorenog porta. Skeniranje započinje tako što se ispita *IP ID* zombi računala slanjem *SYN/ACK* paketa istom. Zombi računalo ne očekuje taj *SYN/ACK* paket pa odgovara *RST* paketom koji u sebi sadrži i *IP ID* (u ovom primjeru 31337). Na temelju informacija koje se posjeduju o zombiju može se kreirati *SYN* paket koji se pošalje meti u ime zombija. Meta šalje *SYN/ACK* paket zombiju pošto se radi o otvorenom portu. Obzirom da zombi ne očekuje ovaj paket od mete jer nije inicirao nikakvu konekciju, odgovara meti sa *RST* paketom. Taj paket ima *IP ID* uvećan za jedan (31338 u ovom primjeru) u odnosu na onaj koji je zombi poslao napadaču (31337). Napadač ponovno šalje zombiju *SYN/ACK* paket na koji zombi odgovara sa *RST*, ali ovaj put sa *IP ID* uvećanim za dva (31339) u odnosu na onaj koji je napadač primio ranije (31337). Napadač nakon usporedbe ta dva *IP ID-a* zaključuje da je port otvoren.

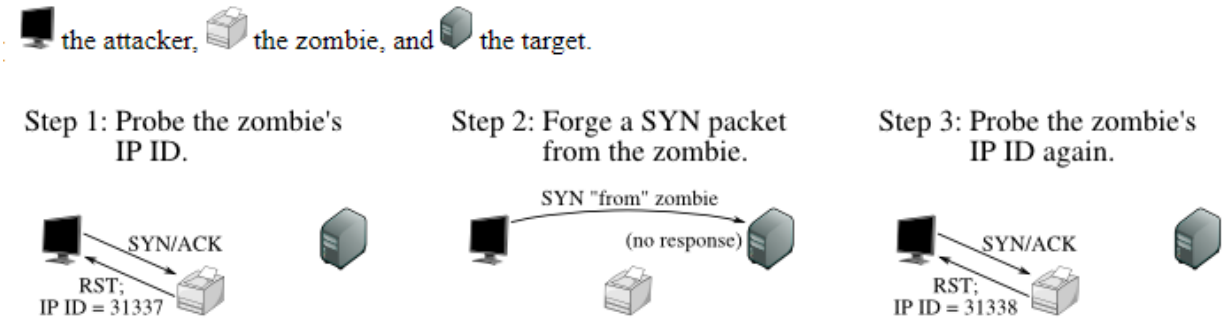
3.8.2 Idle Scan zatvorenog porta



Sl. 3.8. Postupak *idle* skena zatvorenog porta[2]

Slika 3.8. prikazuje kako je izgledao postupak *idle* skena porta kada *nmap* prijavi da je taj port zatvoren. Prvi korak je identičan kao i kod skeniranja otvorenog porta. Saznaje se *IP ID* zombija uz pomoć *SYN/ACK* paketa na koji on odgovara sa *RST* paketom u kojem otkriva svoj *IP ID* (u ovom primjeru 31337). Zatim se, u ime zombija, meti šalje *SYN* paket. Obzirom da je port mete zatvoren, meta šalje zombiju *RST* paket. Nakon ponovnog ispitivanja *IP ID*-a zombija uočava se da se povećao za jedan (31338) što znači da je port mete zatvoren.

3.8.3 Idle Scan filtriranog porta



Sl. 3.9. Postupak *idle* skena filtriranog porta[2]

Slikom 3.9. je prikazan postupak u slučaju filtriranog porta mete. Razlika između skeniranja zatvorenog i filtriranog porta ovom metodom je ta što u slučaju filtriranog porta meta ne šalje zombiju nikakav paket kao odgovor na "lažni" *SYN/ACK* paket koji napadač šalje u ime zombija. Obzirom da zombi ne ostvaruje apsolutno nikakvu komunikaciju s metom, IP ID zombija ostaje nepromijenjen. Nakon ponovljenog slanja SYN/ACK paketa zombiju, napadač otkriva da se IP ID povećao za jedan u odnosu na prvo slanje (sa 31337 na 31338 u ovom primjeru) te je ovaj slučaj za napadača isti kao i slučaj skeniranja zatvorenog porta.

4. OSTALE OPCIJE I ZENMAP

Nmap, iako je prvenstveno osmišljen za skeniranje portova, ima i neke druge mogućnosti. Prema [1] jedna od najpoznatijih odlika mu je otkrivanje operacijskog sustava na ciljanom računalu. Također su bitne i opcije za prepoznavanje servisa i njihovih verzija te *NSE (Nmap Scripting Engine)*.

Rezultate testiranja potrebno je znati dobro dokumentirati. Obzirom da postoji više opcija za formatiranje izlaza potrebno je i njih upoznati.

Zenmap, kao službeno grafičko sučelje za *nmap*, je aplikacija koja olakšava skeniranje početnicima koji se teško snalaze u komandnoj liniji. Također ju koriste i napredniji korisnici zbog jednostavnosti i preglednosti rezultata.

U ovom poglavlju ukratko će biti pojašnjene neke od ostalih opcija *nmap* programa te će se prikazati rad sa *zenmapom*.

4.1. Prepoznavanje servisa i njihovih verzija

Nakon što je poznato koji portovi su otvoreni, na red dolazi otkrivanje usluga i njihovih verzija kako bi se otkrilo što se u stvari krije na tim portovima. Pomoću *nmap-services*, baze podataka u kojoj se nalazi više od 2200 poznatih servisa, *nmap* može povezati određene portove sa određenim servisima. Postoji i baza podataka *nmap-service-probes* pomoću koje *nmap* ispituje servise i prepoznaje ih na temelju njihovog odgovora. Pokušavaju se otkriti informacije poput protokola na kojem servis radi, naziv aplikacije, verzija, tip uređaja i porodica operacijskog sustava .

```
# nmap -sV  10.10.1.48

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-11 12:49 Central
Daylight Time

Interesting ports on 10.10.1.48:
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 2.0.6
22/tcp   open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
25/tcp   open  smtp    Postfix smtpd
80/tcp   open  http    Apache httpd 2.2.8 ((Ubuntu))
MAC Address: 00:0C:29:D5:38:F4 (VMware)
Service Info: Host:  10.10.1.48; OSs: Unix, Linux

Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 8.33 seconds
```

Sl. 4.1. Prikaz rezultata pokretanja opcije za otkrivanje verzija -sV [3]

Na slici 4.1. prikazani su rezultati skena koji uz skeniranje portova vrši i otkrivanje verzija nad tim portovima. *Nmap*, uz stupac sa nazivima servisa, ispisuje stupac sa verzijama servisa. Na ovaj način saznaju se dodatne informacije o skeniranom računalu i mogu se tražiti propusti u sustavu s obzirom na verzije servisa koje se koriste. Također je otkriven i operacijski sustav kao Unix, Linux.

4.2. Otkrivanje operacijskog sustava

Nmap-os-db baza podataka sadrži više od 2600 poznatih otisaka operacijskih sustava. Ukoliko dođe do podudaranja, putem nekog od otisaka, operacijskog sustava odredišnog *hosta* s bazom *nmap-os-db, nmap* ispisuje o kojem je operacijskom sustavu riječ. Svaki otisak sadrži podatke o nazivu izdavača, operacijskom sustavu, generaciji operacijskog sustava i tipu uređaja.

```
# nmap -O 10.10.1.48

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-11 13:09 Central
Daylight Time
...

MAC Address: 00:0C:29:D5:38:F4 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.9 - 2.6.28
Network Distance: 1 hop
```

Sl. 4.2. Rezultat skena za otkrivanje verzije operacijskog sustava [3]

Na slici 4.2. je prikazano skeniranje verzije operacijskog sustava odredišta. Otkriveno je da se radi o Linux-u i to nekoj od verzija između 2.6.9 i 2.6.28.

Prema [3] potrebno je da je barem jedan port odredišta otvoren i jedan zatvoren kako bi se pravilno provelo otkrivanje operacijskog sustava. Kod skeniranja više *hostova*, moguće je koristiti opciju *--osscan-limit* koja u kombinaciji sa opcijom *-O* skenira samo *hostove* koji ispunjavaju taj kriterij.

4.3. NSE

Nmap Scripting Engine (NSE) je prema [1] jedana od najsnažnijih i najfleksibilnijih mogućnosti programa *nmap*. *NSE* omogućava korisnicima da razvijaju vlastite jednostavne skripte koje će automatizirati razne mrežne zadatke i podjele ih sa ostalim korisnicima. Također postoje i već ugrađene skripte koje nude razne mogućnosti poput otkrivanja ranjivosti, otkrivanja stražnjeg ulaza (*backdoor detection*) i iskorištavanja sigurnosnih propusta.

4.3.1. Pokretanje podrazumijevanih skripti

```
# nmap --script default 10.10.1.70

Starting Nmap 5.00 ( http://nmap.org ) at 2009-11-13 15:09 CST
Interesting ports on 10.10.1.70:
Not shown: 997 filtered ports
PORT      STATE  SERVICE
139/tcp   open   netbios-ssn
445/tcp   open   microsoft-ds
5900/tcp  open   vnc
MAC Address: 00:0C:F1:A6:1F:16 (Intel)

Host script results:
|_ nbstat: NetBIOS name: AXIS-01, NetBIOS user: <unknown>, NetBIOS
MAC: 00:0c:f1:a6:1f:16
|  smb-os-discovery: Windows XP
|  LAN Manager: Windows 2000 LAN Manager
|  Name: WORKGROUP\AXIS-01
|_ System time: 2009-11-13 15:09:12 UTC-6

Nmap done: 1 IP address (1 host up) scanned in 52.40 seconds
```

Sl. 4.3. Rezultat pokretanja podrazumijevanih *(engl. defaultnih)* skripti [3]

Na slici 4.3. prikazan je rezultat pokretanja podrazumijevanih skripti nad metom. Opcija *--script default* može se zapisati i na kraći način kao opcija *-sC*. Više o kategorijama NSE skripti moguće je pronaći na [2] gdje je naveden popis kategorija i svih skripti u svakoj pojedinoj kategoriji. Također je svaka pojedina skripta opisana te je prikazan primjer korištenja.

4.4. Formatiranje izlaza

Loše organiziran izlaz, odnosno rezultati skeniranja, mogu uvelike otežati analizu i dokumentaciju istog. *Nmap* nudi nekoliko izlaznih formata i opcija za kontrolu izlaza. Neke od opcija su kontrola opsežnosti izlaza i kontrola poruka za otklanjanje pogrešaka. Sam izlaz može biti poslan u datoteku gdje se dalje obrađuje.

Pomoću opcije *-v* moguće je povećati nivo količine informacija o samom skeniranju. Opcijom *-vv* prikazuje se još više informacija nego navođenje opcije *-v*.

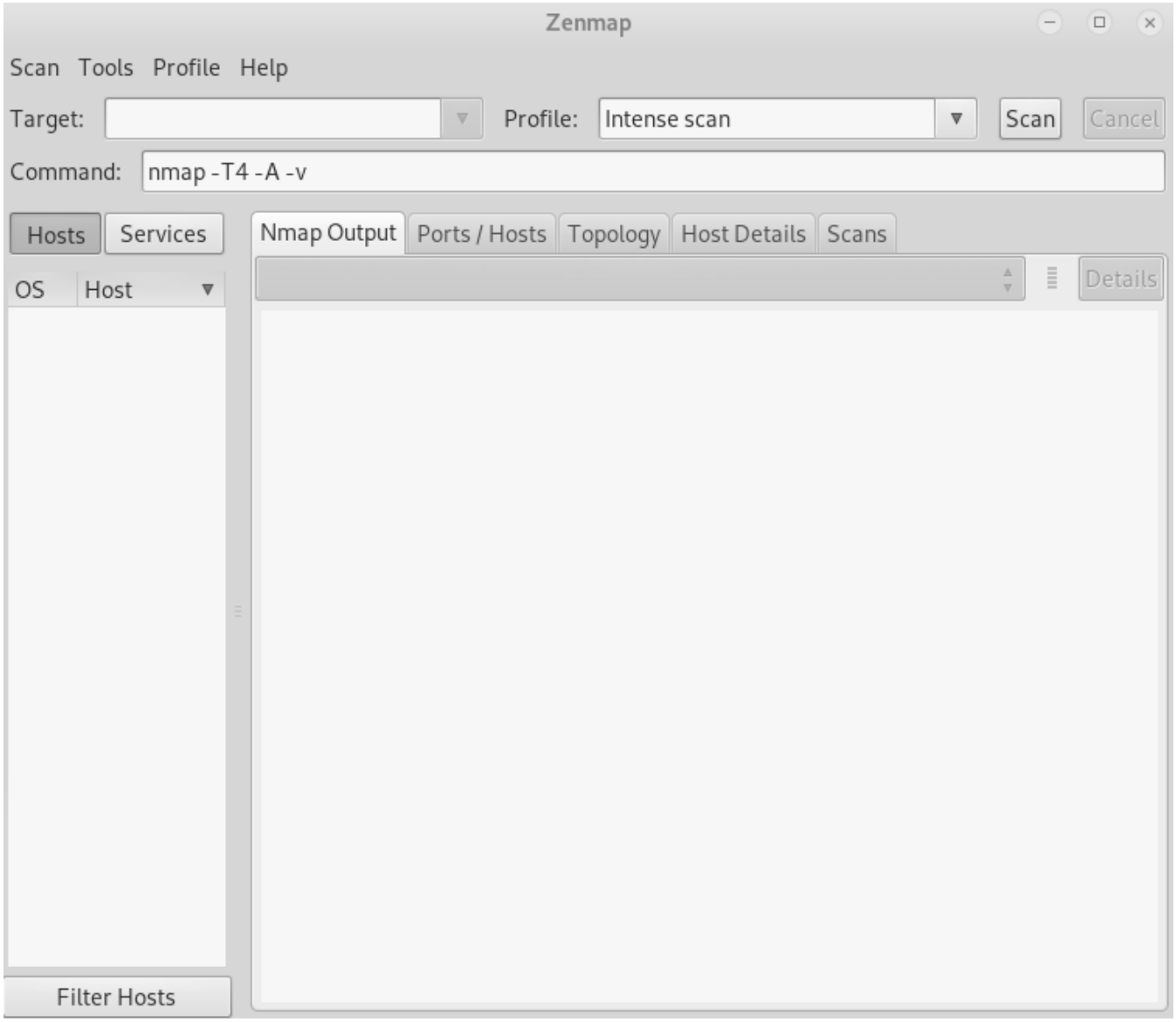
Opcija *-d* služi za povećavanje informacija o otklanjanju pogrešaka. Korisna je ukoliko korisnik sumnja da postoji greška u samom *nmap* programu ili nije siguran što *nmap* radi i zašto.

Opcija *-oN <filespec>* omogućava da izvještaj bude spremljen u vanjsku datoteku imena navedenog pod *filespec*. Naredba *nmap -oN rezultati mojPc* tako sprema rezultate podrazumijevanog *(engl. defaultnog)* port skena računala "mojPc" u datoteku "rezultati".

Ostale naredbe za formatiranje izlaza i primjere korištenja moguće je proučiti na [2].

4.5. Zenmap

Zenmap je službeno grafičko sučelje za *nmap*. Radi se o besplatnoj aplikaciji otvorenog koda koja je osmišljena kako bi pojednostavila *nmap* početnicima dok istovremeno pruža napredne značajke iskusnijim korisnicima. Za pokretanje *zenmapa* u Kali Linux-u potrebno je u komandnoj liniji pokrenuti naredbu *zenmap* nakon čega se pojavljuje sučelje prikazano na slici 4.4..

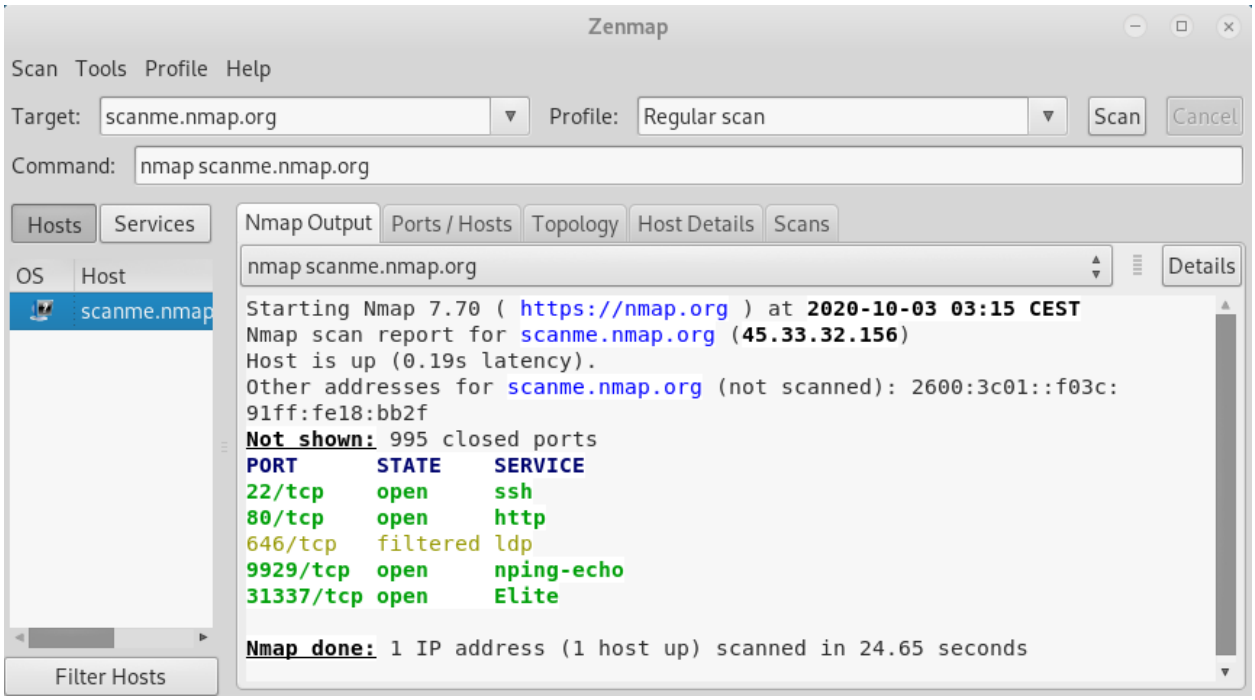


Sl. 4.4. Grafičko sučelje *zenmap*

Prema [1] *zenmap* nije osmišljen kako bi zamijenio *nmap*, nego kako bi ga učinio korisnijim. Neke od prednosti korištenja *zenmapa* su:

- Jednostavno pokretanje skeniranja (nije potrebno poznavati nmap naredbe)
- Jednostavno pokretanje istog skena (iste metode skeniranja) više puta uz pomoć profila
- Interaktivni i grafički prilagođen prikaz rezultata
- Praćenje rezultata dok se korisnik ne odluči odbaciti ih
- Mogućnost usporedbe različitih skenova
- Pruža uvid u *nmap* naredbu koja se "gradi" prilikom odabira opcija u grafičkom sučelju

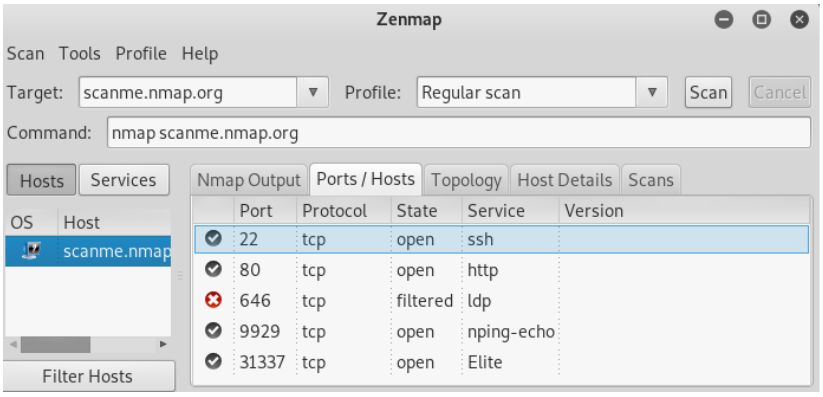
4.5.1. Skeniranje pomoću zenmapa



Sl. 4.5. Skeniranje korištenjem *zenmapa*

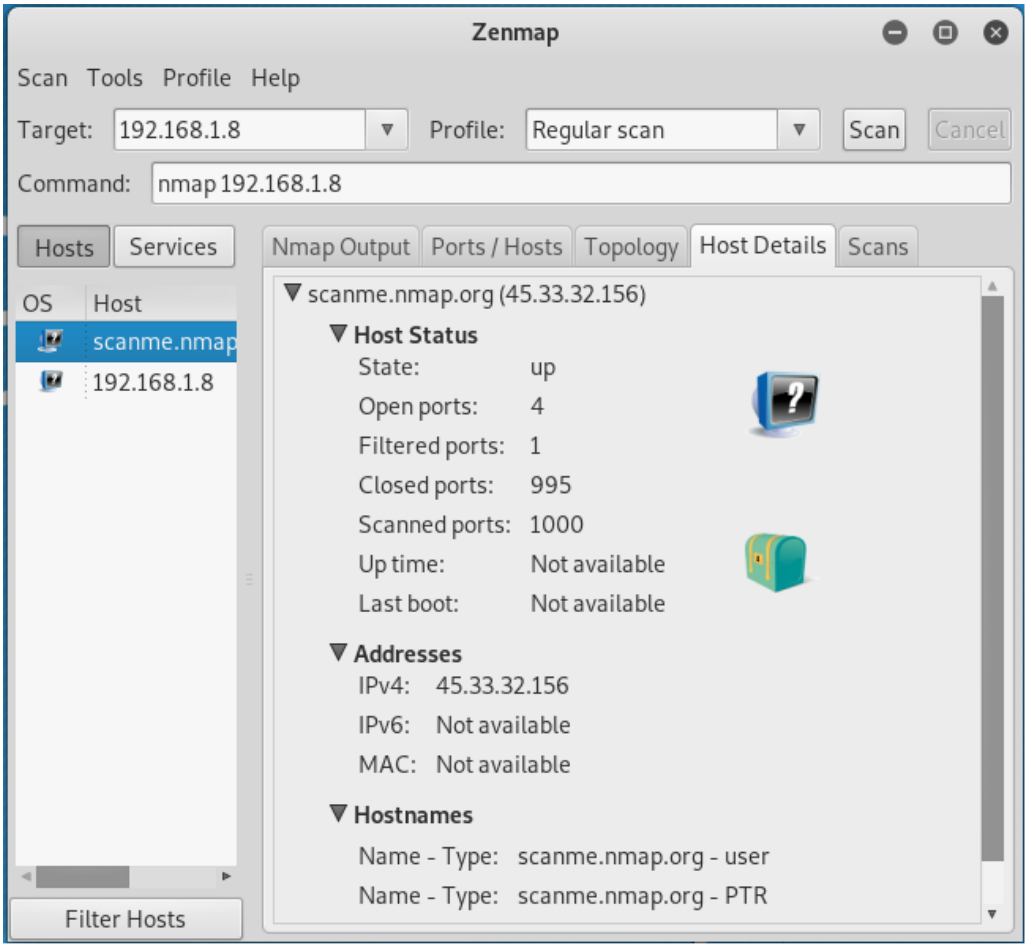
Na slici 4.5. je prikazano kako izgleda pokretanje podrazumijevanog skena nad odredišnim *hostom "scanme.nmap.org"* . U polje *"target"* unosi se odredišni *host* nad kojim se želi izvršiti skeniranje, a polje *"profile"* služi za biranje vrste skeniranja. Mijenjanjem argumenata u ta dva polja mijenja se i naredba u polju *"command"* koja zapravo prikazuje kako izgleda *nmap* naredba sa odabranim argumentima. Naredbu u tom polju moguće je dodatno izmijeniti dodavanjem ili brisanjem argumenata. Samo skeniranje pokreće se klikom na gumb *"Scan"*.

Također je prikazan i izlaz *nmap* alata koji je sadržajem jednak izlazu koji bi se pojavio pokretanjem iste naredbe u komandnoj liniji. Razlika je u tome što *zenmap* koristi boje kako bi rezultate učinio čitljivijim.



Sl. 4.6. *Ports / Hosts* kartica *zenmap* korisničkog sučelja

Na slici 4.6. su prikazani "osnovni" rezultati istog skeniranja. Kartica *"Ports / Hosts"* pruža sažete, korisnicima lako čitljive, informacije o stanjima portova.



Sl. 4.7. *Host Details* kartica *zenmap* korisničkog sučelja

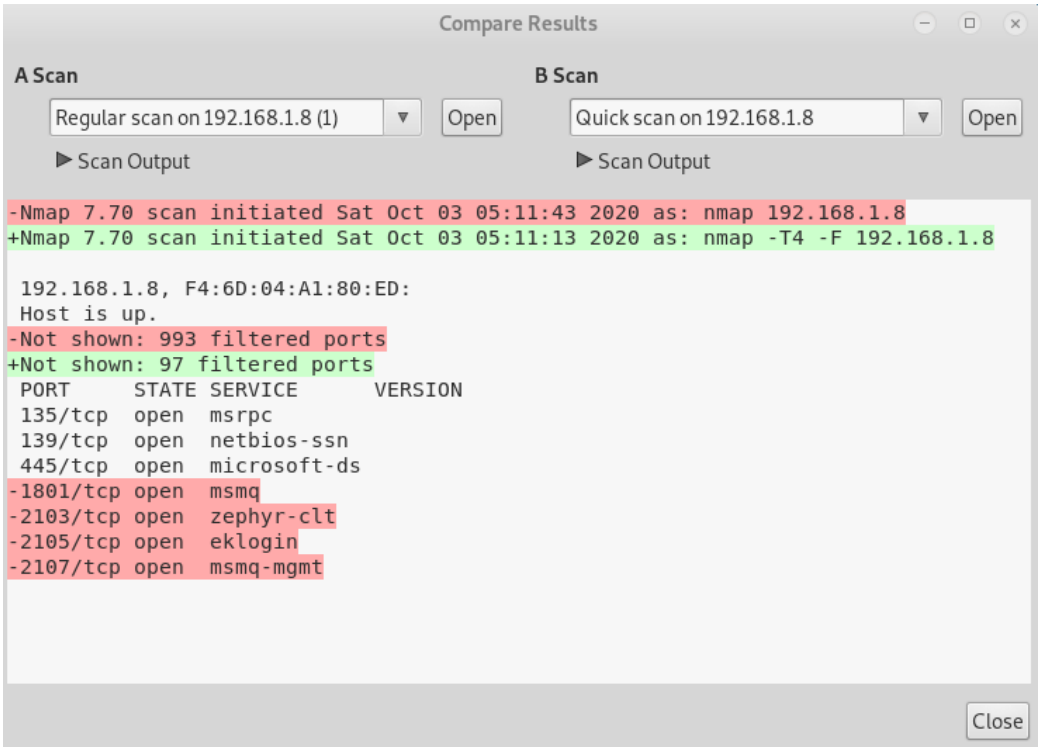
Slikom 4.7. prikazani su rezultati u *"Host Details"* kartici. U ovoj kartici mogu se pronaći informacije o imenu i IP adresi hosta, stanju *hosta*, broju otvorenih, filtriranih i zatvorenih portova, vremenu koje je host budan i vrijeme u koje je host zadnji puta pokrenut. Informacija o operacijskom sustavu odredišnog hosta je prikazana putem ikone. Ikona na ovoj slici označava da otkrivanje operacijskog sustava nije provedeno. Legendu svih ikona moguće je pronaći na [1] i [2].

Kartica *"Topology"* pruža uvid u interaktivni prikaz konekcija između *hostova* na mreži. Legenda za tumačenje topologije mreže postoji u samom grafičkom sučelju, a pristupa joj se klikom na gumb *"Legend"*. U kartici *"Scans"* moguće je pregledati povijest skeniranja trenutne sesije. Obzirom da se pamte svi skenovi pokrenuti za vrijeme sesije, moguće je pregledavati rezultate više odabranih *hostova* na način na koji korisnik to želi. Klikom na gumb *"Hosts"*

moguće je odabrati željeni *host* te proučiti rezultate skeniranja tog odredišnog *hosta* u prethodno opisanim karticama.

4.5.2. Usporedba rezultata

Klikom na *Tools* > *Compare Results* u *zenmap* izborniku otvara se prozor za usporedbu rezultata skeniranja. U padajućim izbornicima mogu se odabrati nedavno pokrenuti skenovi, ali je moguće i učitati *nmap XML* datoteke pritiskom na gumb *"Open"*.

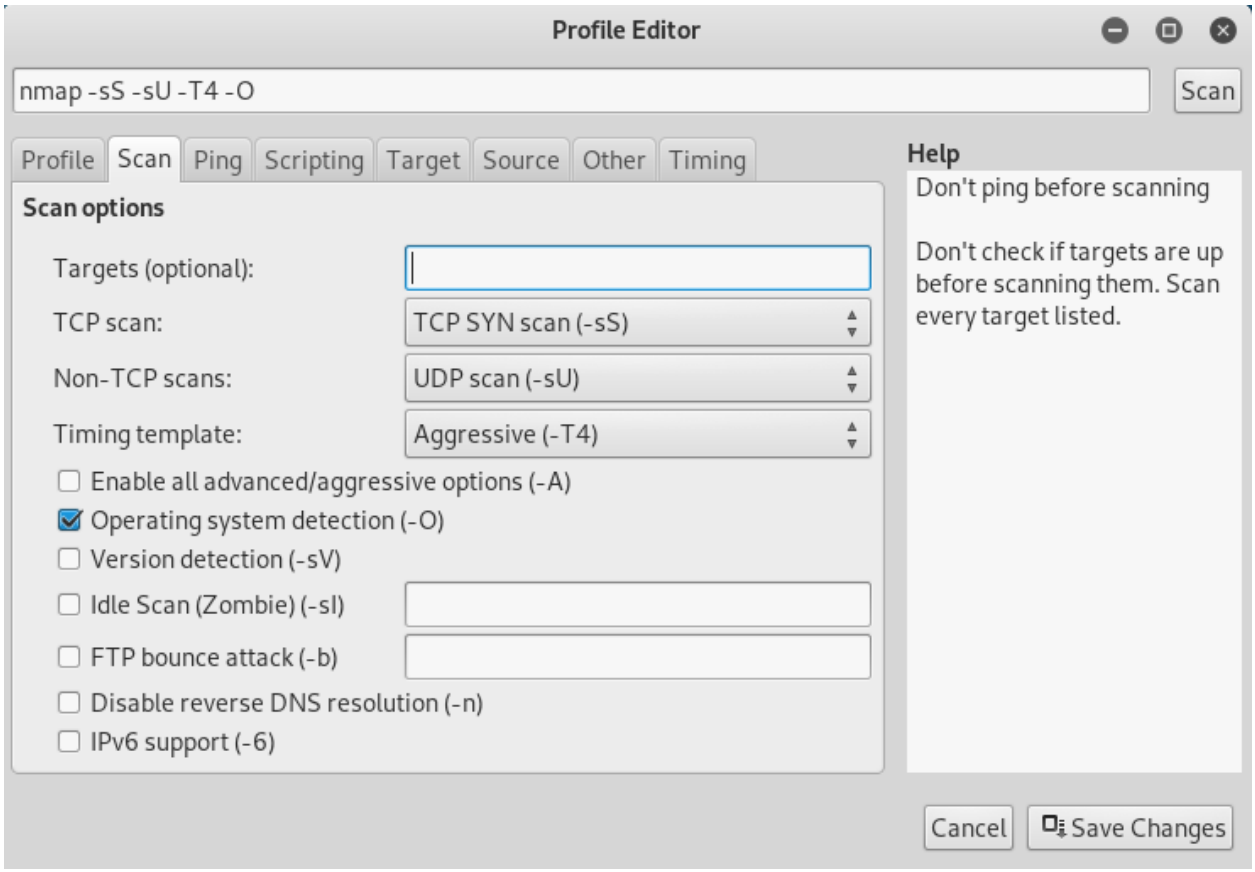


Sl. 4.8. Usporedba rezultata dva različita skena istog odredišnog *hosta*

Na slici 4.8. je prikazan prozor za usporedbu rezultata skeniranja. Za prvi sken (*"A Scan"*) odabran je podrazumijevani sken *hosta* 192.168.1.8, a za drugi (*"B Scan"*) odabran je brzi sken istog odredišnog *hosta*. Dio rezultata koji nije osjenčan niti jednom bojom vrijedi za oba skena. Crvenom bojom osjenčan je dio rezultata koji se veže za prvi (*"A Scan"*), a zelenom za drugi (*"B Scan"*). Prvi sken je skenirao 1000 portova odredišnog *hosta* i pronašao 7 otvorenih dok je drugi skenirao 100 portova i pronašao 3 otvorena. Usporedba rezultata može biti vrlo korisna i ako se pokreće isti sken nad odredišnim *hostovima*, ali u različitim vremenskim periodima. Ovim načinom usporedbe rezultata vrlo je lako uočiti promjenu stanja porta u odnosu na prošlo skeniranje.

4.5.3. Stvaranje vlastitih profila

Klikom na *Profile > New Profile or Command* u *zenmap* izborniku otvara se prozor za stvaranje vlastitih profila. Biranjem opcija u pojedinim karticama stvara se željena naredba kojoj je moguće dati ime i spremiti kao profil. Karticama su pokrivene sve do sada opisane opcije *nmapa* i još mnoge druge.

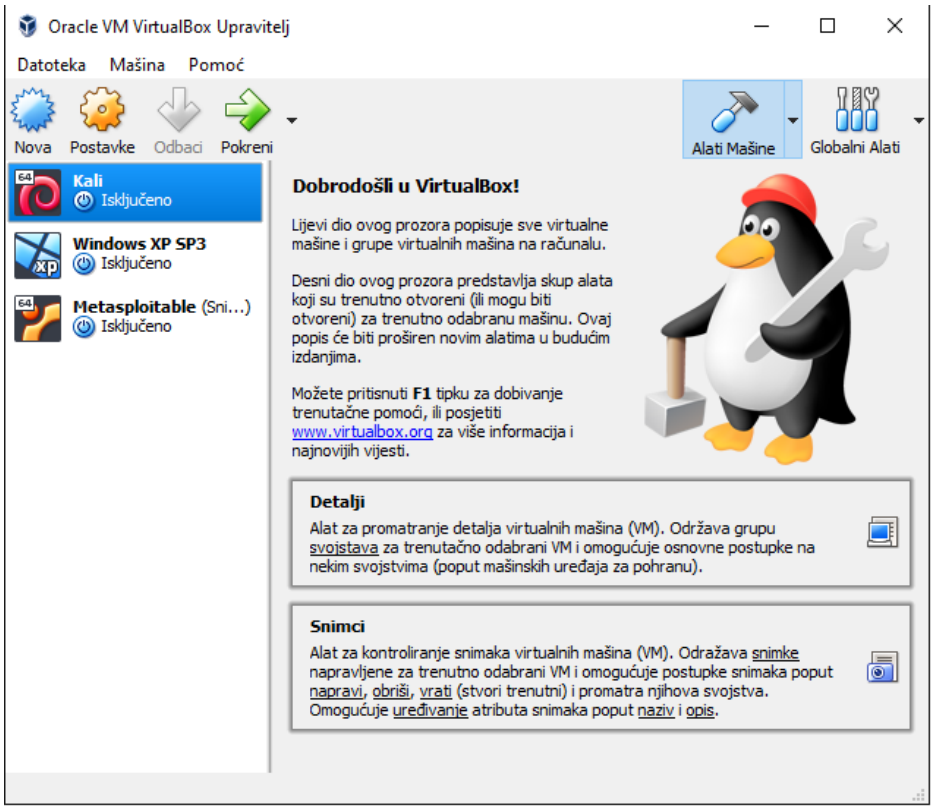


Sl. 4.9. *Zenmap* prozor za stvaranje vlastitih profila i naredbi

Na slici 4.9. prikazano je kako izgleda postupak stvaranja vlastitog *zenmap* profila u kartici *Scan*. U toj kartici se biraju opcije vezane za skeniranje odredišnog *hosta* te su u ovom slučaju odabrane *TCP SYN scan (-sS)* i *UDP scan (-sU)* metode za skeniranje portova. Također su odabrani "agresivni" vremenski predložak *(-T4)* i opcija za skeniranje operacijskog sustava *(-O)*. Naredbu je moguće dalje nadograđivati u ostalim karticama pa potom u kartici profile imenovati profil i spremiti ga klikom na gumb za spremanje promjena *(Save Changes)*.

5. TESTIRANJE U VIRTUALNOM LABORATORIJSKOM OKRUŽENJU

Uz pomoć *Oracle VM VirtualBox* monitora virtualnih mašina postavljeno je laboratorijsko okruženje na lokalnoj mreži. Glavnu ulogu u laboratoriju imaju Kali Linux virtualna mašina (192.168.1.4), Windows XP SP3 virtualna mašina (192.168.1.7) i Metasploitable2 virtualna mašina (192.168.1.2) na Ubuntu Linux operacijskom sustavu.



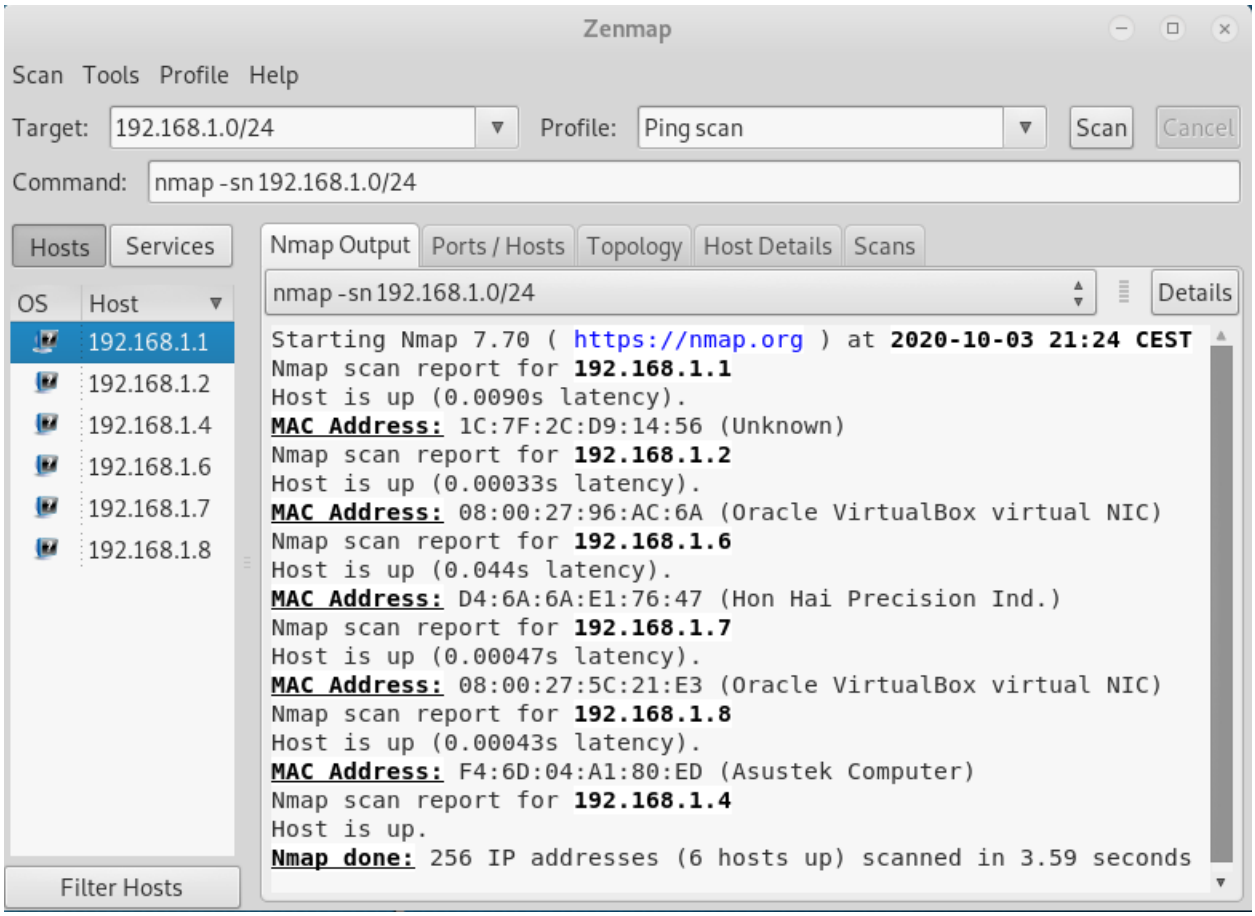
Sl. 5.1. *Oracle VM VirtualBox* upravitelj u laboratoriju

Slikom 5.1. prikazano je kako izgleda *Oracle VM VirtualBox* upravitelj te lista ranije spomenutih mašina u istom.

Prvo je obavljen postupak otkrivanja domaćina kako bi se, uz pomoć rezultata, otkrili "zanimljivi" *hostovi*. Nakon toga je proveden brzi *TCP SYN* sken nad dva odabrana *hosta* od kojih je jedan (Windows XP SP3) zaštićen vatrozidom, a drugi (Metasploitable2) bez zaštite. Rezultati skeniranja su prikazani pomoću mogućnosti grafičkog sučelja *zenmap* te su detaljno opisani. Na kraju je prikazano postavljanje pravila vatrozida *hosta* Metasploitable2 kojim se odbacuju *TCP SYN* paketi te na taj način sabotiraju rezultati *TCP SYN* skena. Također je pokazano kako zaobići to pravilo i saznati točna stanja portova.

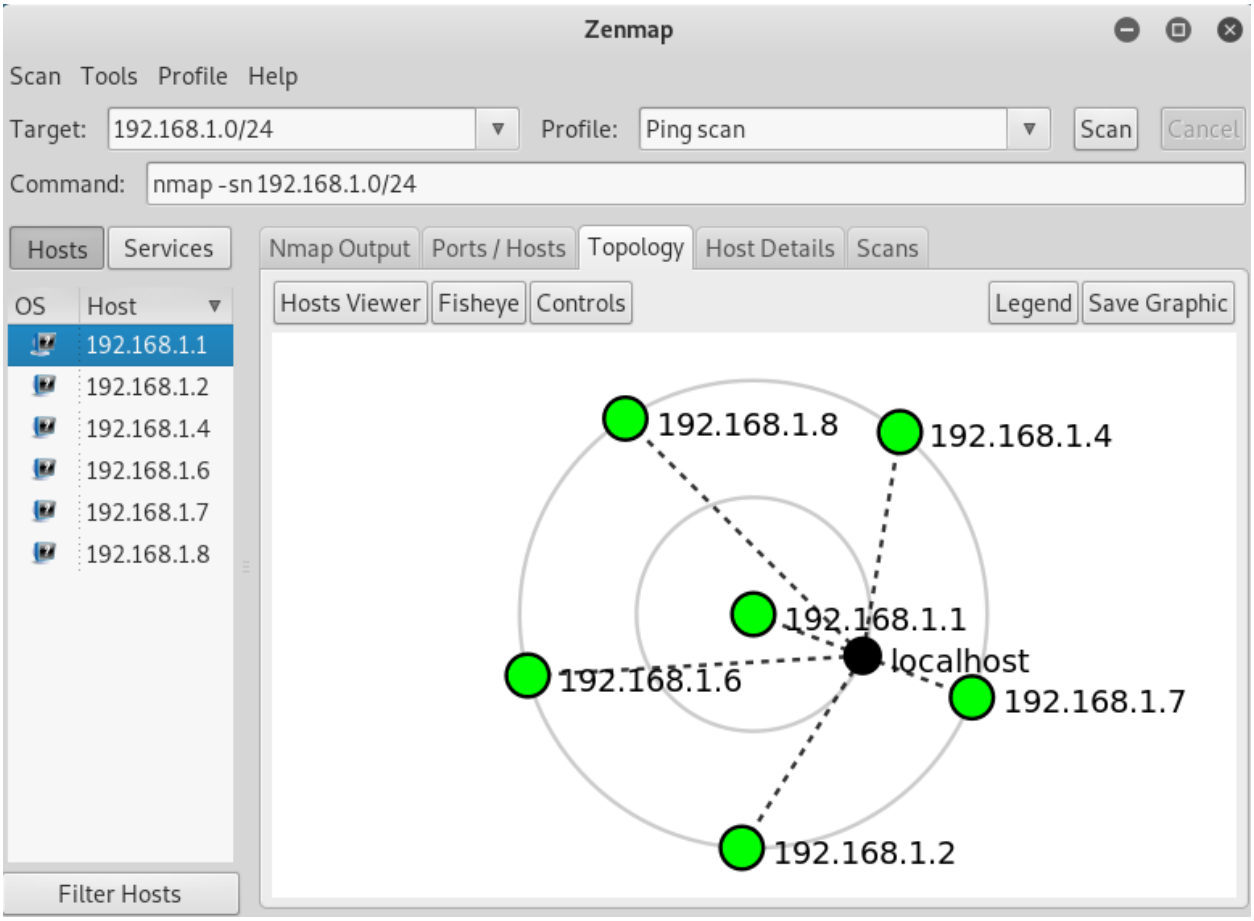
5.1. Otkrivanje domaćina u laboratoriju

Prije samog skeniranja portova potrebno je provjeriti koji su hostovi na mreži aktivni. To je učinjeno uz pomoć *zenmapa* tako što je pokrenut *Ping scan* profil nad cijelom lokalnom mrežom.



Sl. 5.2. Otkrivanje domaćina pomoću *zenmapa*

Na slici 5.2. je prikazan rezultat otkrivanja domaćina laboratorijske mreže uz pomoć *zenmapa*. Ciljani *hostovi* su svi *hostovi* koji se nalaze na lokalnoj mreži, a za profil skeniranja koristi se *Ping* sken. Ove dvije opcije kao rezultat stvaraju *nmap* naredbu koja je prikazana u polju *Command*. Obzirom da se *Ping* scan vrši na lokalnoj razini, provodi se *ARP scan* metoda za otkrivanje domaćina. Skeniranje je prijavilo šest budnih *hostova* te ispisalo njihove *MAC* adrese. Iz *MAC* adrese se mogu saznati neke dodatne informacije kao što su proizvođač mrežne kartice uređaja. Te informacije su javno dostupne, a na slici su prikazane u zagradama pokraj MAC adresa. Zanimljivo je to što je za dva *hosta* sa slike moguće zaključiti da su virtualna računala obzirom da njihove *MAC* adrese upućuju na to da koriste virtualne mrežne adaptere unutar *VirtualBoxa*.

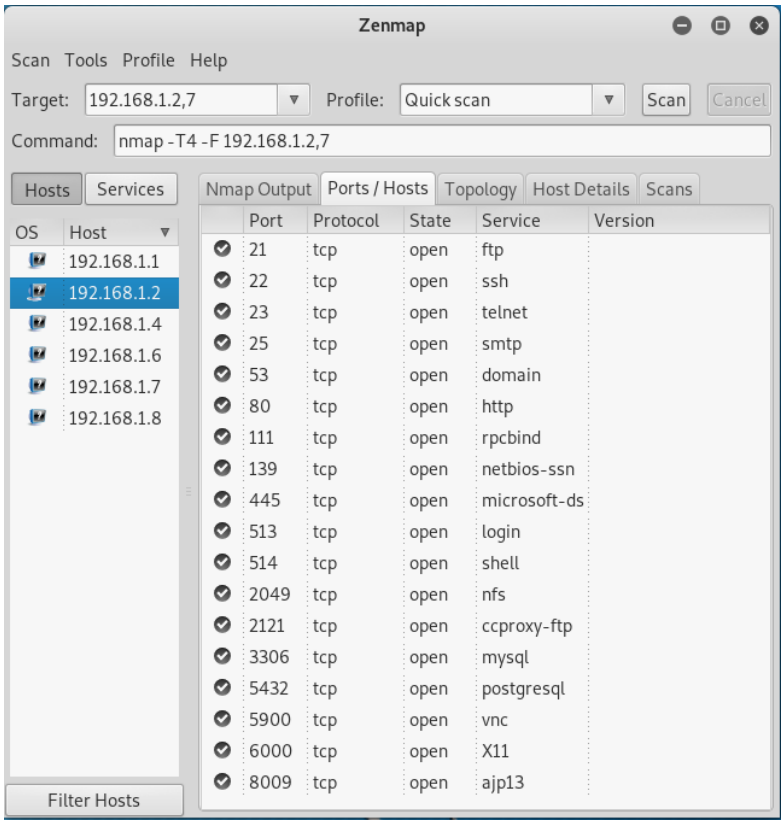


Sl. 5.3. Topologija laboratorijske mreže nakon otkrivanja domaćina

Slikom 5.3. prikazano je kako izgleda topologija laboratorijske mreže u kartici *Topology zenmapa* nakon provedenog postupka otkrivanja domaćina. Svi *hostovi* su označeni zelenim krugovima što označava da su to *hostovi* sa manje od tri otvorena porta. Treba imati na umu da je ovaj graf topologije mreže stvoren nakon pokretanja naredbe koja u sebi naglašava da se skeniranje *portova* ne provodi (opcija *-sn*) te zbog toga *hostove* niti ne može označiti drugom bojom nego zelenom. Iz grafa se također vidi da se radi o lokalnoj mreži i da nisu otkrivene nikakve informacije o postavljenim rutama (isprekidane crte).

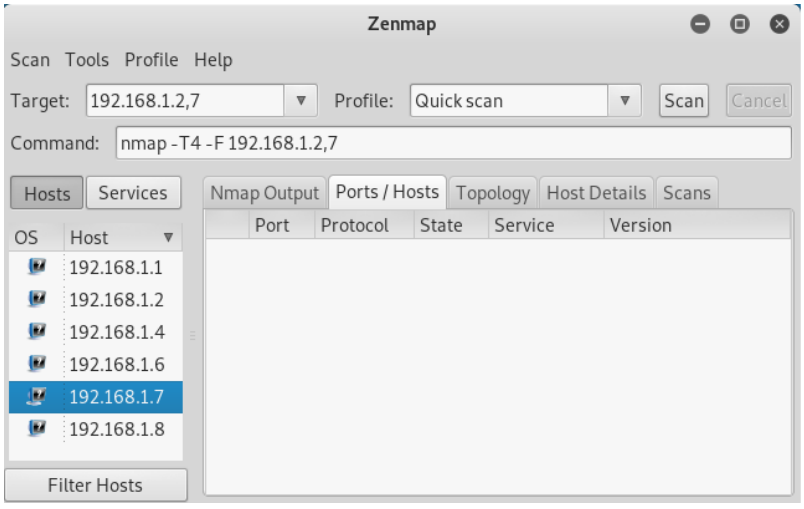
5.2. Quick scan laboratorijske mreže

Postupkom otkrivanja domaćina otkriveno je da su dva *hosta* virtualna računala te će se daljnja testiranja vršiti nad njima. Radi se o *hostovima* s IP adresama 192.168.1.2 i 192.168.1.7. Pokretanjem Quick scan profila moguće je u vrlo kratkom vremenu skenirati 100 dobro poznatih portova odredišnih hostova.



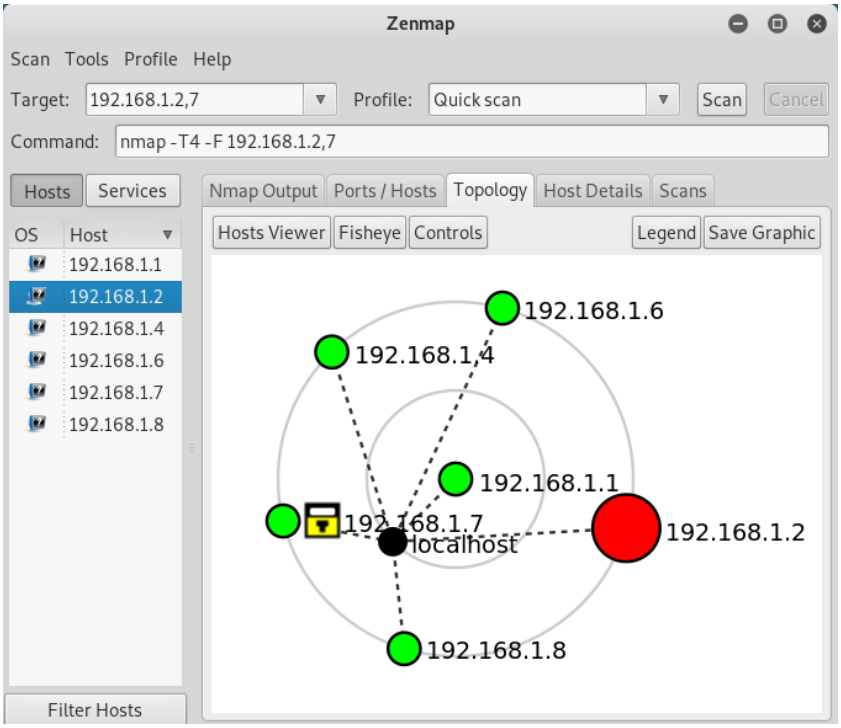
Sl. 5.4. Stanje portova *hosta* 192.168.1.2

Iako su u području za navođenje odredišnih *hostova* navedena oba *hosta* , slika 5.4. prikazuje samo rezultate skeniranja *hosta* 192.168.1.2. Taj *host* je trenutno odabran u području u kojem su navedeni svi aktivni *hostovi* u laboratoriju te su prikazani samo njegovi otvoreni portovi. Na slici se uz popis otvorenih portova, vidi i naziv usluga koje se pružaju tim portom. Obzirom da u naredbi nije navedena metoda skeniranja, pokrenut je podrazumijevani *TCP SYN* sken. Odredišni *host* nema implementirane nikakve mehanizme zaštite od *TCP SYN* skena te pruža uvid u svoje otvorene portove.



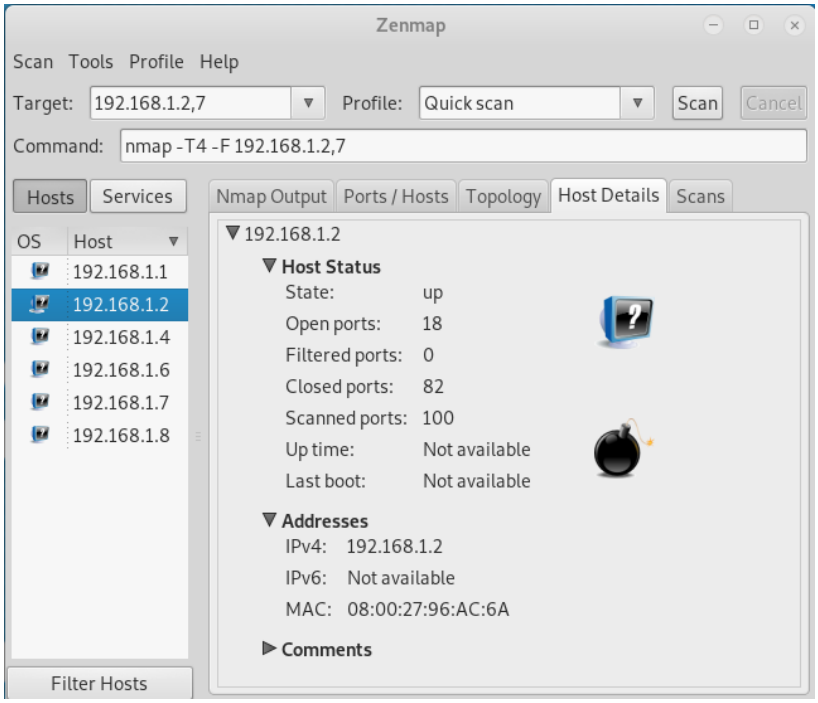
Sl. 5.5. Stanje portova *hosta* 192.168.1.7

Na slici 5.5. se vidi da je lista otvorenih portova *hosta* 192.168.1.7 prazna, što znači da TCP SYN metodom nije otkriven niti jedan otvoreni port toga *hosta*. Moguće je da neki otvoreni port nije otkriven zbog opcije za brzo skeniranje (-F) koja skenira manji broj portova od podrazumijevanog, ali to ovdje nije slučaj. Ovaj host ima uključen vatrozid te je, za razliku od hosta 192.168.1.2, zaštićen od ove metode skeniranja.



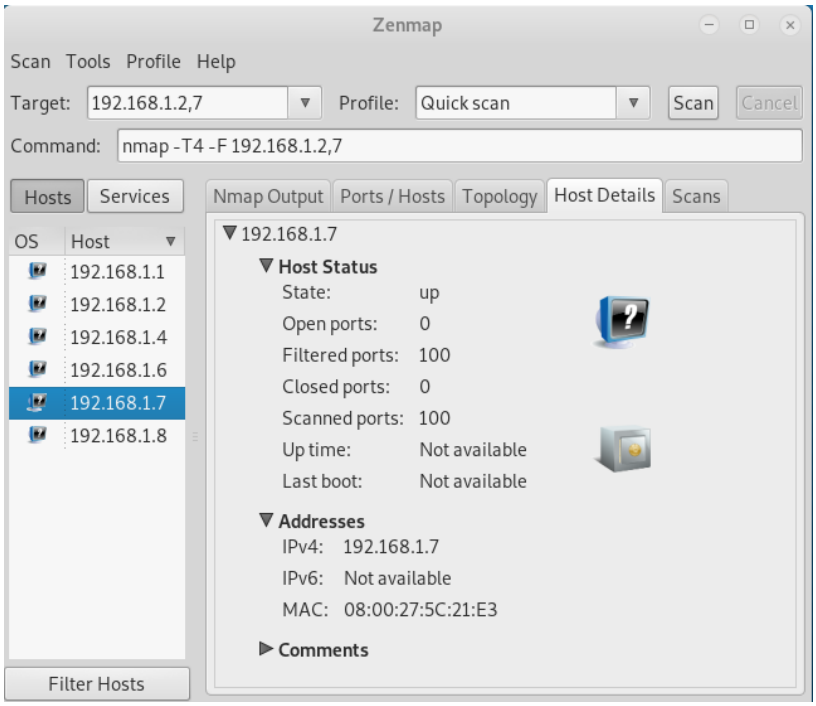
Sl. 5.6. Topologija laboratorijske mreže nakon *quick* skena odabranih *hostova*

Slika 5.6. prikazuje topologiju laboratorijske mreže nakon provedenog skeniranja opisanog u ovom potpoglavlju. Graf topologije se mijenja s obzirom na novootkrivene informacije te je sada *host* 192.168.1.2 označen crvenom bojom, što znači da se radi o *hostu* s više od šest otvorenih portova. Također se uz zeleno označeni *host* 192.168.1.7 pojavio lokot koji označava da je *host* protumačen kao *host* kojemu su neki portovi filtrirani, što upućuje na to da ima implementiran neki od mehanizama zaštite od provedene metode skeniranja.



Sl. 5.7. Rezultati *Quick scana hosta* 192.168.1.2 u *Host Details* kartici

Slike 5.7. i 5.8. prikazuju rezultate skeniranja u kartici *Host Details*. *Hostu* 192.168.1.2 je od 100 skeniranih portova, njih 18 je otvoreno, a 82 su zatvorena. Ikona bombe označava da se radi o *hostu* sa više od 9 otvorenih portova. *Hostu* 192.168.1.7 su svih 100 skeniranih portova definirani kao filtrirani. Ikona sefa predstavlja *host* sa 0-2 otvorena porta.



Sl. 5.8. Rezultati *Quick scana hosta* 192.168.1.7 u *Host Details* kartici

5.3. Izbjegavanje vatrozida

S obzirom da *host* 192.168.1.2 ne koristi nikakve mehanizme zaštite od skeniranja portova, podrazumijevani sken je dovoljan za otkrivanje stanja svih portova. Ovakva situacija je jako loša za sigurnost toga *hosta* i same mreže. U ovom slučaju ipak se radi o *Metasploitable 2* virtualnoj mašini koja je s namjerom osmišljena kao ranjiva kako bi se nad njom provodila penetracijska testiranja i sigurnosna istraživanja. *Metasploitable 2* radi na *Ubuntu Linux* operacijskom sustavu te ima mogućnost postavljanja pravila postupanja s *IPv4* paketima putem alata *iptables*. Zbog navedenih razloga ovaj *host* može poslužiti kao idealna meta za testiranje u ovom potpoglavlju. Nmap, kao jedan od najpoznatijih port skenera, nudi razne metode i mogućnosti za izbjegavanje vatrozida. Prvi i osnovni način kojim se može provesti skeniranje, usprkos postavljenim pravilima vatrozida, je odabir prave metode za skeniranje portova .

5.3.1. Vatrozid koji odbacuje TCP SYN pakete

```
root@metasploitable:/home/msfadmin# iptables -I INPUT -p tcp --tcp-flags ALL SYN
 -j DROP
root@metasploitable:/home/msfadmin# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere            tcp flags:FIN,SYN,R
ST,PSH,ACK,URG/SYN

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@metasploitable:/home/msfadmin# _
```

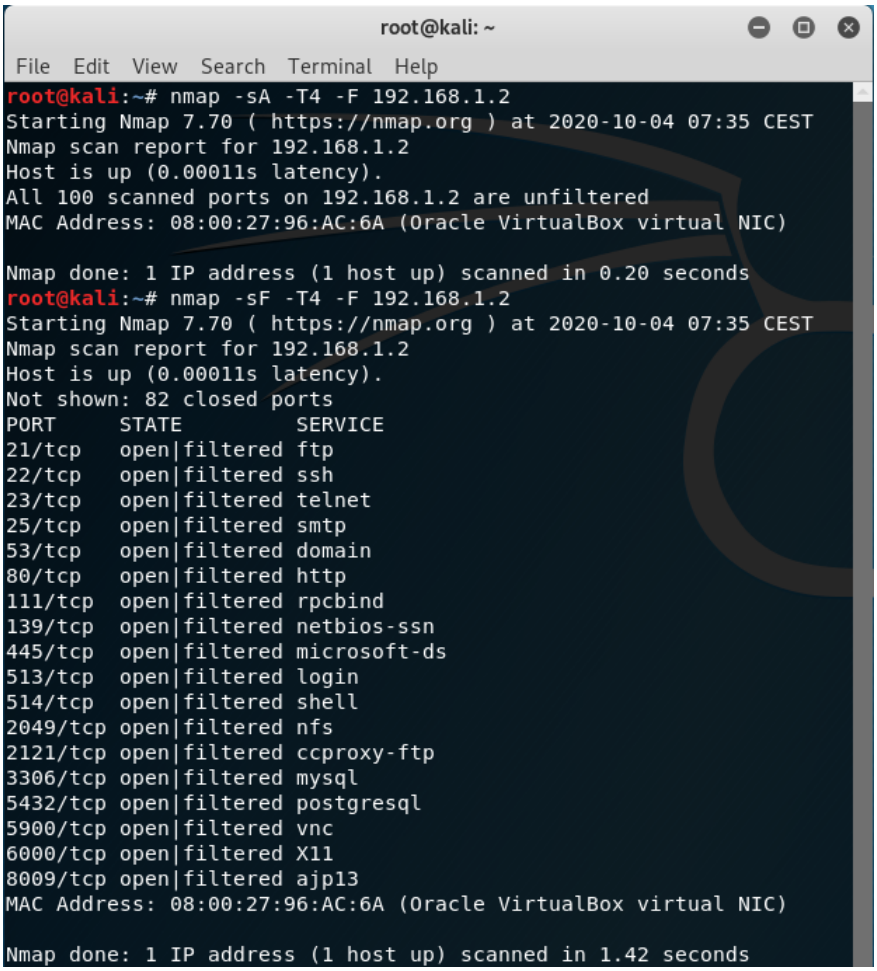
Sl. 5.9. Postavljanje pravila za odbacivanje *SYN* paketa pomoću alata *iptables*

Prva naredba sa slike 5.9. služi za dodavanje novog pravila vatrozida kojim se odbacuju nadolazeći TCP paketi sa postavljenom SYN zastavicom. Također se na slici vidi, pomoću opcije iptables -L, da je to jedino pravilo definirano ovim alatom.

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# nmap -T4 -F 192.168.1.2
Starting Nmap 7.70 ( https://nmap.org ) at 2020-10-04 06:58 CEST
Nmap scan report for 192.168.1.2
Host is up (0.00034s latency).
All 100 scanned ports on 192.168.1.2 are filtered
MAC Address: 08:00:27:96:AC:6A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.23 seconds
```

Sl. 5.10. *Quick scan* hosta 192.168.1.2

Slikom 5.10. prikazani su rezultati *Quick scana hosta* 192.168.1.2 koji ima novopostavljeno pravilo za filtriranje IPv4 paketa. Svaki od 100 skeniranih portova prikazan je kao filtriran port. Ovi rezultati se znatno razlikuju od rezultata dobivenih u potpoglavlju 5.2. u kojem su stanja portova ovoga *hosta* određena kao 18 otvorenih i 82 zatvorena porta. Razlog zbog kojeg se rezultati ta dva jednaka skena toliko razlikuju je taj što je uvedeno pravilo za odbacivanje paketa s postavljenom *TCP SYN* zastavicom, a skeniranje se obavlja *TCP SYN scan* metodom za skeniranje portova koja koristi upravo pakete s tom zastavicom.



Sl. 5.11. Zaobilaženje vatrozida i otkrivanje stvarnih stanja portova *hosta* 192.168.1.2

Slika 5.11. prikazuje rezultate brzog skena izvedenog pomoću *TCP ACK Scan* i *TCP FIN Scan* metode za skeniranje portova odredišnog *hosta* 192.168.1.2. Pomoću prve naredbe (*TCP ACK* skena) utvrđeno je nefiltrirano stanje za svaki od 100 skeniranih portova, a drugom naredbom (*TCP FIN* sken) utvrđeno je da je 82 porta zatvoreno i ispisano je ostalih 18 portova kojima je dodijeljeno stanje otvoren|filtriran. S obzirom da je prvim skenom utvrđeno da niti jedan port nije filtriran, može se zaključiti da je svaki od tih 18 otvoren|filtriran portova zapravo otvoreni port.

6. ZAKLJUČAK

Network Mapper je besplatni alat otvorenog koda za istraživanje i praćenje mreže. U Kali Linux-u se koristi kroz komandnu liniju te je vrlo važno dobro poznavati sintaksu programa kako bi se moglo pokrenuti željeno skeniranje. U nekim slučajevima se čak pokreću i naredbe koje sadrže greške pa zbog toga korisnik treba biti dodatno oprezan.

Skeniranje mreža za koje korisnik programa nema dozvolu nije legalno pa je neophodno upoznati se sa načinima odabira ciljanog računala prije nego što se vrši bilo kakvo ozbiljnije skeniranje. Postupkom otkrivanja domaćina moguće je prepoznati aktivne *hostove* na mreži koji su ujedno i zanimljiviji od onih neaktivnih. Treba razumjeti da dobro čuvani *hostovi* ponekad mogu biti na izgled neaktivni, a zapravo su aktivni te je zbog toga važno upoznati razne metode za otkrivanje domaćina.

Glavna funkcionalnost koju ovaj alat pruža,ona zbog koje je prvotno i nastao, je skeniranje portova. Skeniranje portova je vrlo važna funkcionalnost za mrežnu sigurnost. Smanjivanje broja i složenosti pruženih usluga smanjuje mogućnosti za potencijalne napadače koji pokušavaju provaliti te zbog toga mrežni administratori trebaju redovno skenirati mrežu za koju su zaduženi pomoću alata kao što je *Nmap*. Kada se govori o skeniranju portova zapravo se govori o određivanju stanja portova putem neke od metoda za skeniranje portova. Različite metode skeniranja mogu dati različite rezultate za iste portove u ovisnosti o tome kako je metoda implementirana i kako je zaštita implementirana. Zbog toga nije dovoljno poznavanje samo jedne metode skeniranja portova. Uz skeniranje portova, *nmap* nudi i neke druge funkcionalnosti kao što je prepoznavanje servisa i njihovih verzija. Ova funkcionalnost je poprilično važna zbog toga što je otkrivanje informacije o verziji protokola na nekom portu vrlo korisna informacija. Otkrivanje verzije protokola također otkriva i potencijalne ranjivosti. Još jedna korisna funkcionalnost ovog alata bi bila otkrivanje operacijskog sustava. *Nmap* posjeduje bazu podataka u kojoj su povezani poznati otisci sa operacijskim sustavima i na taj način može odrediti operacijski sustav odredišta. Ova informacija je također jako važna za sigurnost sustava. Grafičko sučelje *zenmap* olakšava korištenje *nmap* alata tako što omogućava slaganje naredbe uz pomoć biranje željenih opcija. Rezultati *nmap* skeniranja koji su prikazani kroz grafičko sučelje *zenmap* su čitljiviji i jednostavnije ih je analizirati.

Kako bi korisnici koji koriste ovaj alat mogli dobro dokumentirati rezultate svojih testiranja i bez korištenja *zenmap* sučelja, u *nmap* su implementirane i funkcionalnosti za formatiranje izlaza. Na kraju krajeva sigurnosni alat je dobar onoliko koliko dobar izvještaj može ponuditi.

LITERATURA

[1] Gordon Fyodor Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure, USA, 2009. ISBN 0979958717, 9780979958717.

[2] Gordon Fyodor Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning (online edition), 2020. URL http://nmap.org/book [Online; accessed 2020-09-23].

[3] Nicholas Marsh: Nmap® Cookbook The Fat-free Guide to Network Scanning, 2010

[4] Satyam Singh:Port Scanners URL https://resources.infosecinstitute.com/port-scanners/ [Online; accessed 2020-09-23].

[5] David Shaw: Nmap Essentials, 2015 ISBN 978-1-78355-406-5

SAŽETAK

Glavni cilj ovog rada bio je upoznati se s Network Mapper alatom. Kako bi se taj cilj ostvario postavljen je virtualni laboratorij koji je realiziran uz pomoću Oracle VM VirtualBox-a. Laboratorij se sastojao od Kali Linux distribucije na kojoj je instaliran nmap alat te Windows XP SP3 i Metasploitable2 računala koja su bila mete. Prvo je proučena sintaksa samog programa kako bi se njime moglo ispravno služiti. Nadalje, objašnjeno je kako se određuju ciljana računala kako bi se skeniranje moglo provoditi samo na željenim računalima. Postupak otkrivanja domaćina objašnjen je kroz nekoliko poznatih metoda za koje su korišteni primjeri iz literature i laboratorija. Zatim je objašnjeno skeniranje portova i detaljno su opisane neke od metoda kojima se portovi skeniraju kako bi se što bolje upoznalo sa načinom na koji određene metode rade. U ovom dijelu rada, koji je služio samo za razumijevanje metoda skeniranja portova, korišteni su samo primjeri iz literature. Zatim su ukratko pokazane ostale opcije nmap alata kao što su opcije za prepoznavanje usluga i njihovih usluga, otkrivanje operacijskog sustava, korištenje NSE(Nmap Scripting Engine i formatiranje izlaza. Predstavljen je i zenmap, službeno grafičko sučelje nmap alata. Na kraju je provedena vježba u laboratorijskom okruženju kojom je prikazan scenarij testiranja lokalne mreže te dodavanje i zaobilaženje pravila vatrozida. Izradom ovog završnog rada ostvaren je dobar temelj za daljnje proučavanje mrežne sigurnosti.

Ključne riječi: Mrežna sigurnost, Network Mapper, otkrivanje domaćina, skeniranje portova, zenmap

ABSTRACT

Application of Network Mapper tool in network diagnostics

Main goal of this bachelor's thesis was getting familiar with Network Mapper tool. In order to achieve this goal, a virtual laboratory was set using Oracle VM VirtualBox. The laboratory included Kali Linux with nmap installed, Windows XP SP3 machine and Metasploitable2 machine. Windows and Metasploitable2 machines were the targets. To make it possible to use the tool, it was necessary to study the syntax first. Then, it was needed to understand how to specify the target hosts. Host discovery was researched thru some known methods which were also tested in laboratory. Port scanning was the main focus of this thesis since Network Mapper is known as port scanner. Methods of port scanning were explored and explained in detail in this paper using examples from literature. Other functionalities of nmap were briefly researched. These functionalities include service and version detection, operating system detection, usage of Nmap Scripting Engin (NSE) and output formatting. Zenmap, the official nmap graphical user interface, was also introduced. Finally, an laboratory exercise was conducted and the scenario of testing the local network and adding and bypassing firewall rules was shown. The preparation of this thesis provided a good basis for further study of network security.

Keywords: Network security, Network Mapper, host discovery, port scanning, zenmap