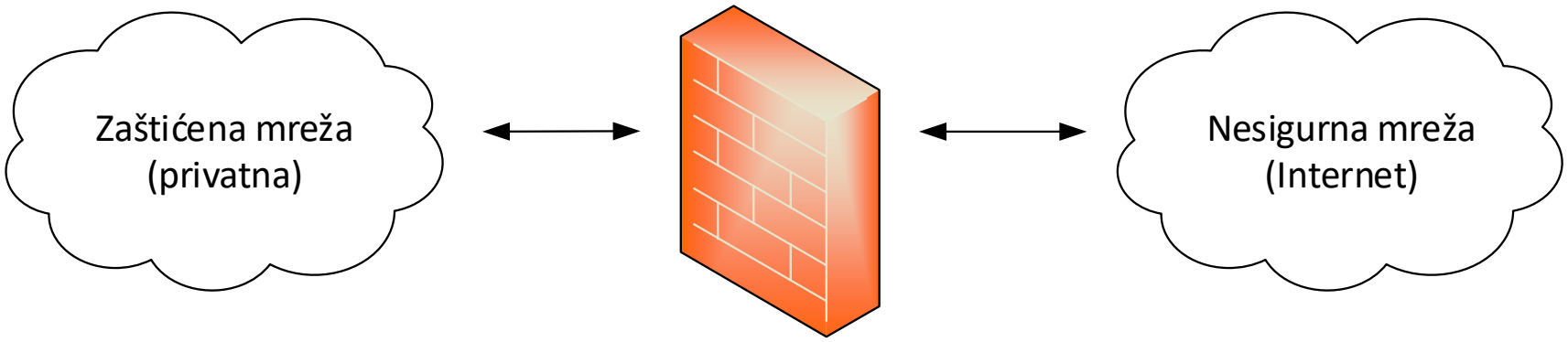
Mrežne barijere

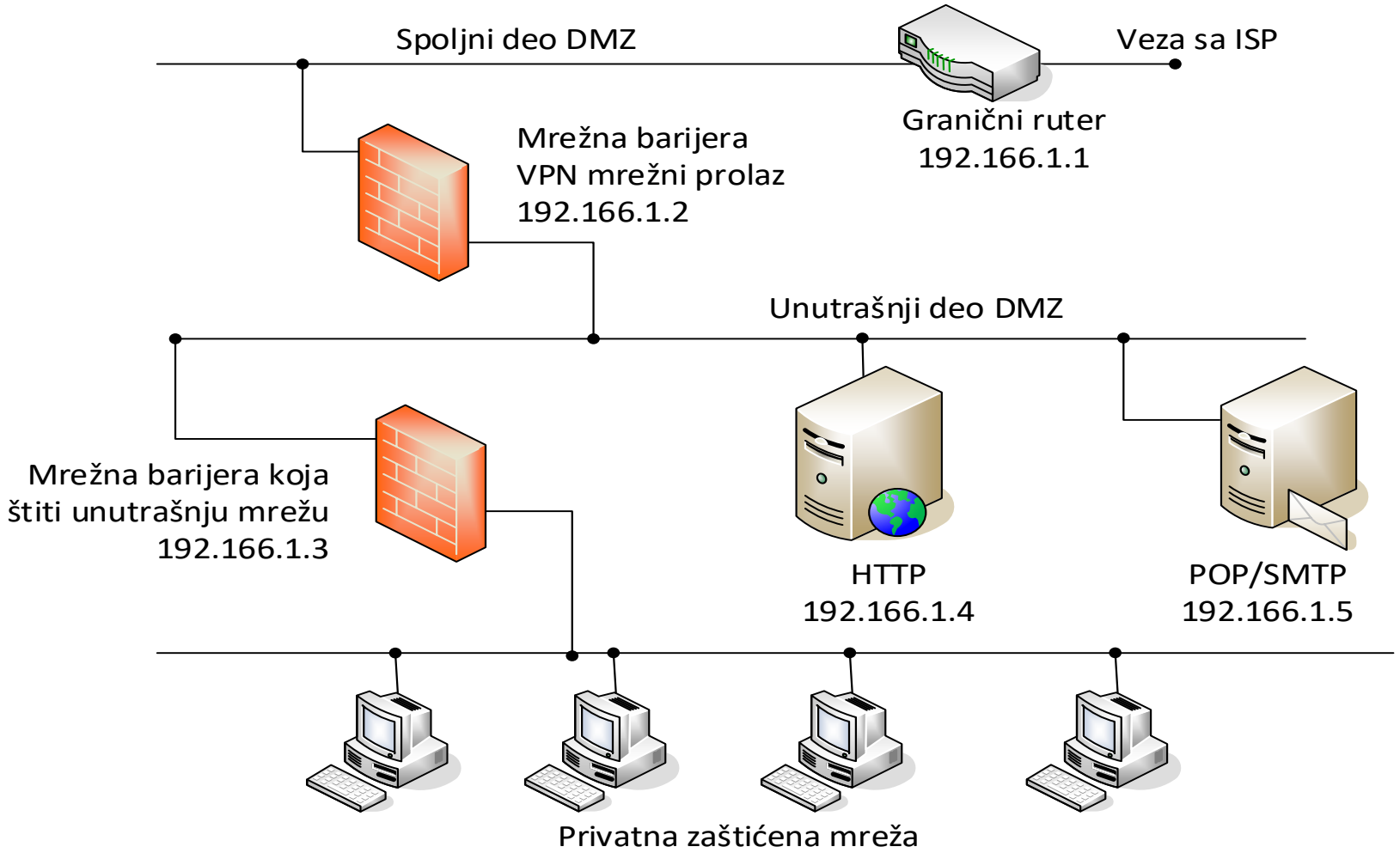
Sadržaj

Šta je mrežna barijera?

- Mrežne barijere (engl. *firewalls*)
  - Koriste se za postavljanje kontrolnih tačaka bezbednosti na granicama privatnih mreža.
  - U kontrolnim tačkama ispituju sve pakete koji prolaze između privatne mreže i Interneta.
  - U zavisnosti od toga da li paketi zadovoljavaju pravila definisana listama za kontrolu pristupa firewall će dozvoliti ili zabraniti protok tog paketa.
- Jednostavno rečeno: *firewall* je filtar na relaciji lokalna mreža – Internet.



Skica povezivanja privatne mreže na javnu preko firewall-a



Funkcije firewall-a

- Osnovne funkcije:
  - **Filtriranje paketa** na osnovu izvorišne i odredišne adrese i broj porta.
  - **Prevođenje mrežnih adresa** (engl. *network adress translation*, NAT).
    - Adrese računara u privatnoj mreži prevode se u jednu ili više javnih IP adresa.
  - **Proksi servisi** (engl. *proxy*).
    - Omogućavaju većem broju računara da dele jednu vezu ka Internetu.
    - Keširaju podatke kako bi se ubrzao pristup tim podacima sa lokalne mreže.
- Dodatne funkcije:
  - **Šifrovana autentifikacija** korisnika sa javne mreže firewall-u.
    - Time se kontroliše pristup privatnim mrežama sa spoljnih lokacija.
  - **Virtualno privatno umrežavanje** (VPN).
    - Uspostavljanje kriptografski zaštićene veze između dve privatne mreže preko Interneta.
- Dodatne funkcije *firewall*-a koji su sposobni da obave *deep packet inspection*:
  - **Filtriranje na osnovu sadržaja** (zlonamerni softver, pornografija i slično).

Filtriranje paketa

- Firewall analizira pakete i upoređuje ih s prethodno definisanim skupom pravila.
- Filtriranje je moguće na osnovu bilo kog dela zaglavlja.
- Većina filtara donosi odluku na osnovu:
  - **tipa protokola**,
  - izvorišne i/ili odredišne **IP adrese** ili **adresnog opsega**,
  - **broja porta**.
- Primer:
  - Svim računarima se može dozvoliti da pristupe TCP portu 80.
  - Pristup TCP portu 22 ograničen je računarima koji pripadaju određenom opsegu IP adresa (lokalna mreža).
- Na osnovu definisanih pravila i zaglavlja konkretnog IP paketa filtar može da:
  - prihvati paket,
  - odbaci paket,
  - pdbaci paket i obavesti pošiljaoca da njegov paket nije prihvaćen.

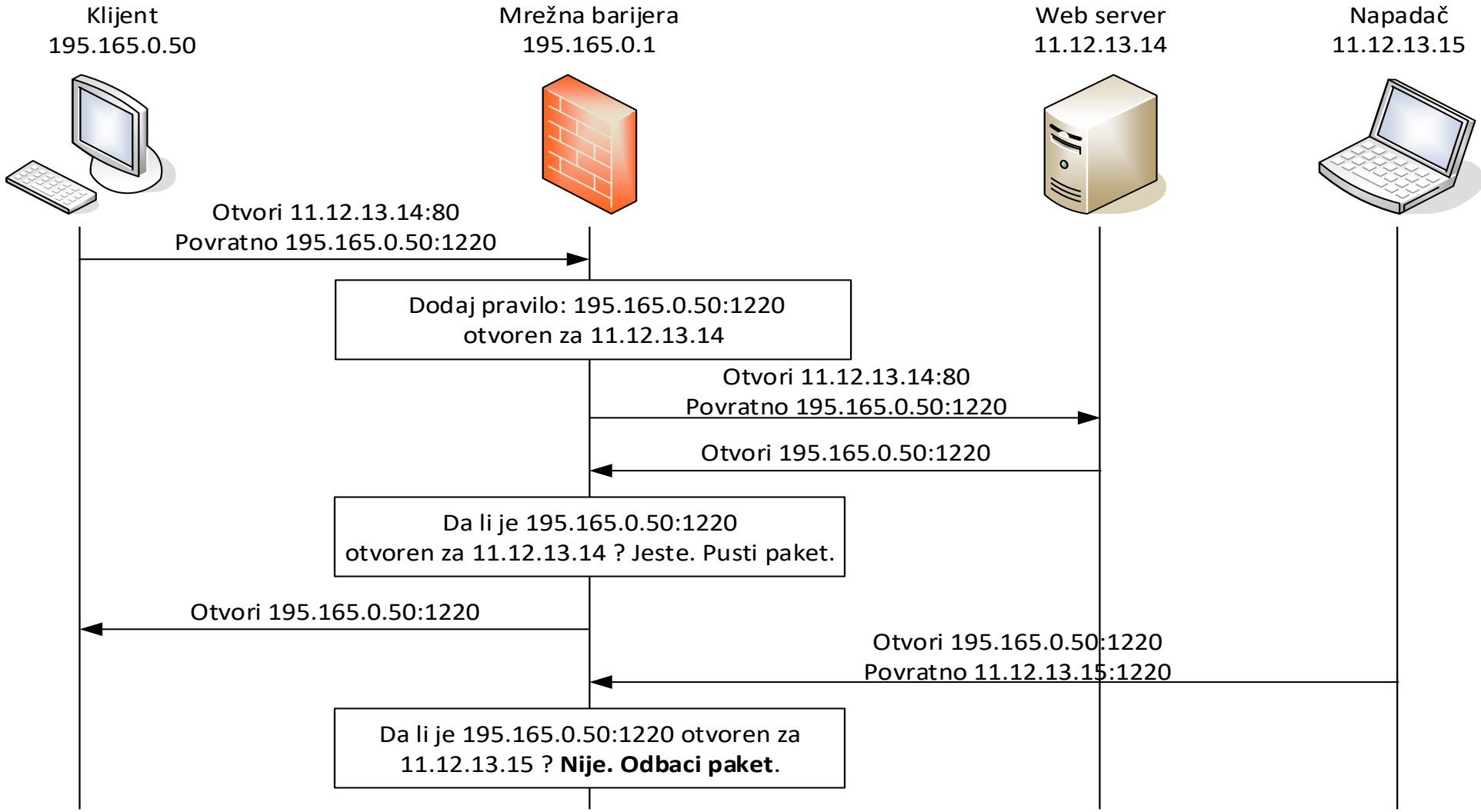
Filtriranje paketa

- Postoje dve vrste filtara paketa.
- **Firewall bez uspostavljanja stanja** (engl. *stateless firewall*).
  - Odbacuje paket ukoliko nema dovoljno informacija šta bi s njim trebalo da uradi.
  - Većina *firewall*-ova ovog tipa ostavlja portove veće od 1024 otvorene.
    - Time se omogućava slanje odgovora računaru koji je poslao zahtev.
    - Ozbiljan sigurnosni propust: trojanski konji mogu da iskoriste ove portove!
- **Firewall sa uspostavljanjem stanja** (engl. *statefull firewall*).
  - Pamti zahteve za uspostavljanjem veze i to koristi prilikom donošenja odluka.
  - *Firewall* u tabeli stanja vodi evidenciju o trenutnim stanjima veza.
    - *Firewall* ovog tipa dozvoljava slanje odgovora ka računarima koji su uspostavili vezu.
  - Potencijalne rupe ostaju otvorene samo onoliko dugo koliko je potrebno!
    - Kada učesnici u sesiji zatvore TCP vezu firewall briše zapise u tabeli stanja.
    - Ukoliko računar u lokalnoj mreži prestane da odgovara računaru na Internetu pre zatvaranja veze *firewall* nakon određenog vremena briše zapis iz tabele stanja.

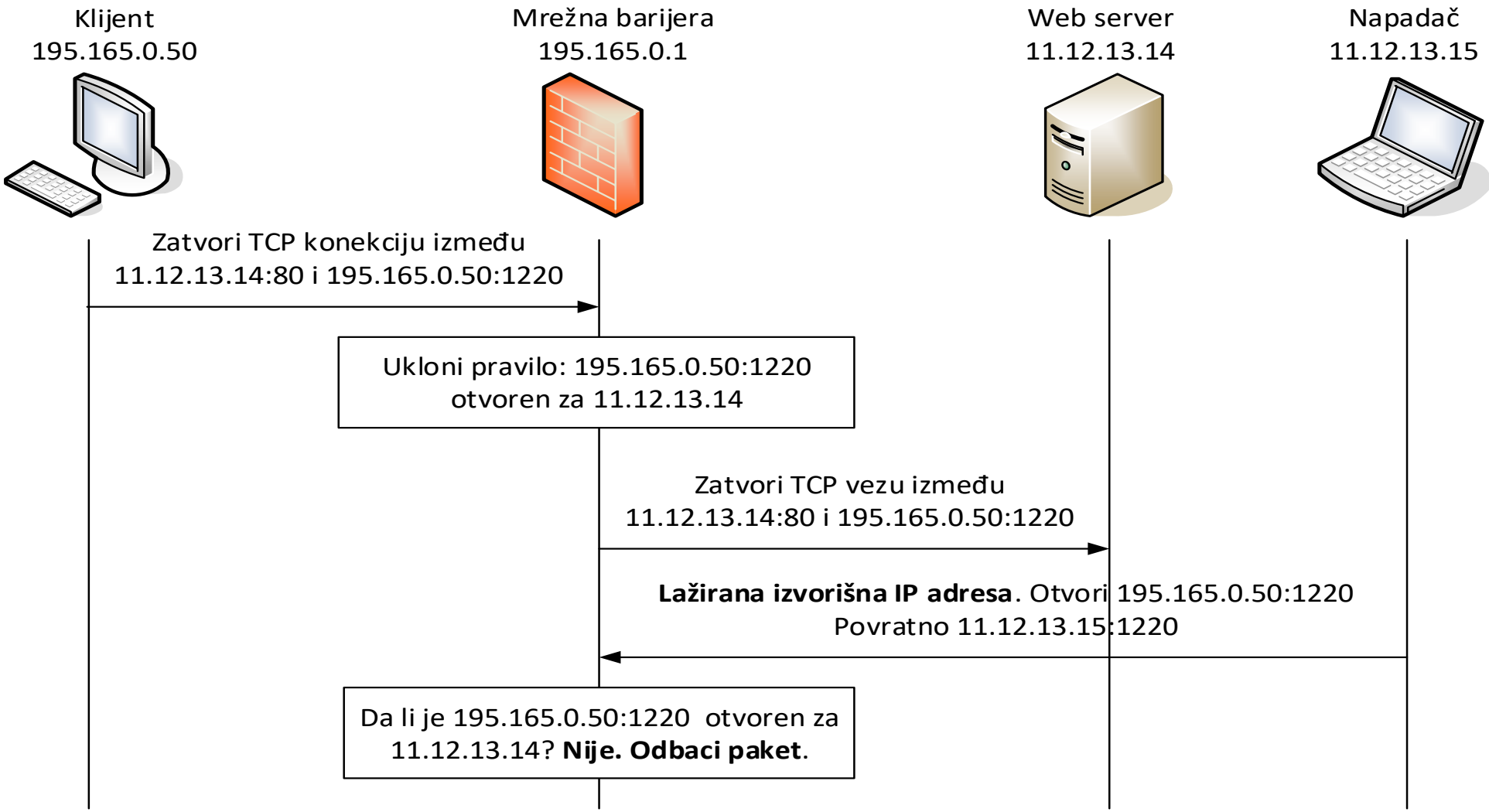
Filtar sa uspostavljanjem stanja (primer)

- (1)
  - Klijent šalje serveru zahtev na port 80 i zahteva odgovor na portu 1220.
  - *Firewall* prosleđuje paket i u tabelu stanja dodaje pravilo:
    - Server 11.12.13.14 može slati pakete računaru 195.165.0.50 na port 1220.
  - Server prima zahtev i šalje odgovor na 195.165.0.50:1220.
  - *Firewall* proverava tabelu stanja i utvrđuje da server 11.12.13.14 može slati pakete računaru 195.165.0.50 na port 1220.
  - Računar 195.165.0.50 prima odgovor na portu 1220.
  - Ukoliko napadač sa IP adresom 11.12.13.15 pokuša da odgovori na zahtev klijenta:
    - *Firewall* u tabeli stanja neće pronaći zapis koji to dozvoljava → neće proslediti odgovor.
- (2)
  - Klijent zatvara TCP vezu.
    - Napadač sa lažiranom IP adresom ne može da nastavi komunikaciju sa računarom 195.165.0.50:1220.

Filtar sa uspostavljanjem stanja (primer, 1. deo)

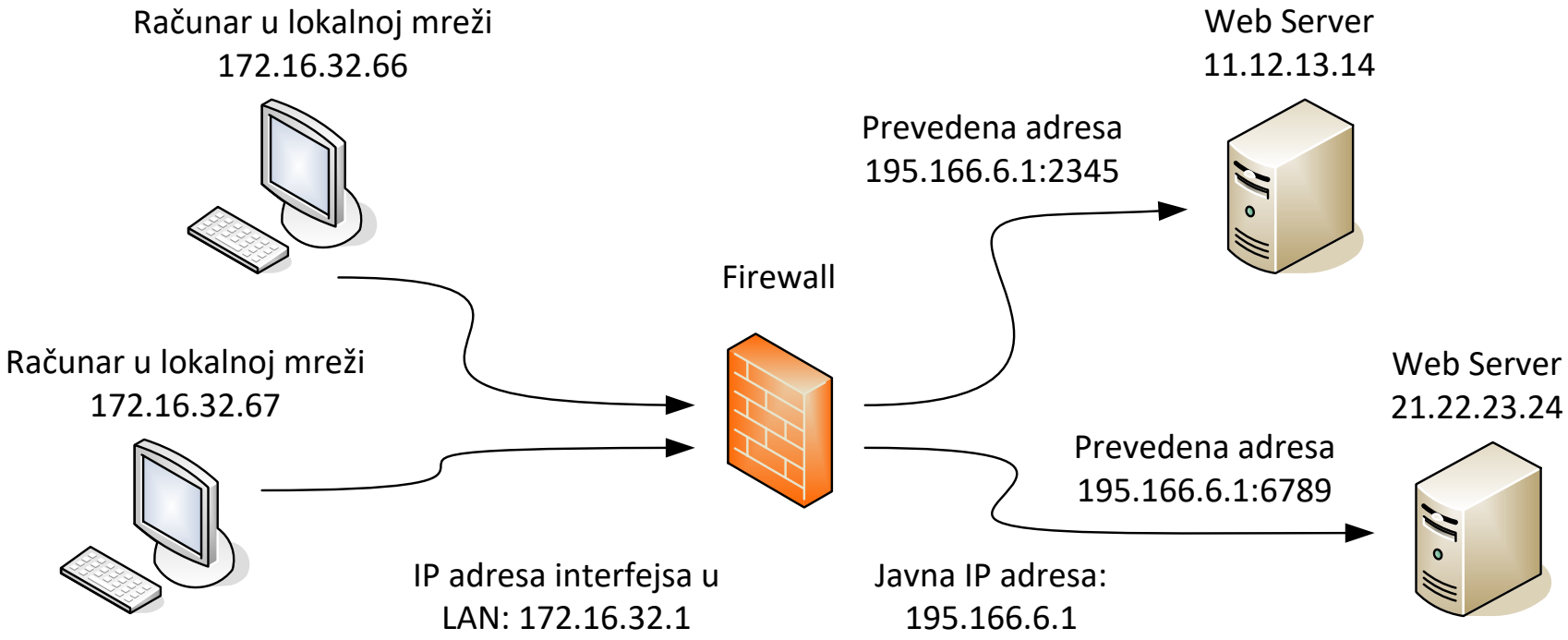


Filtar sa uspostavljanjem stanja (primer, 2. deo)



Prevođenje mrežnih adresa (NAT)

- NAT prevodi IP adrese računara iz privatne mreže u javnu IP adresu *firewall*-a (ili rutera).
- NAT omogućava uštedu javnih IP adresa.
  - Jedna javna IP adresa uz korišćenje različitih brojeva porta prevodi se u veći broj privatnih.

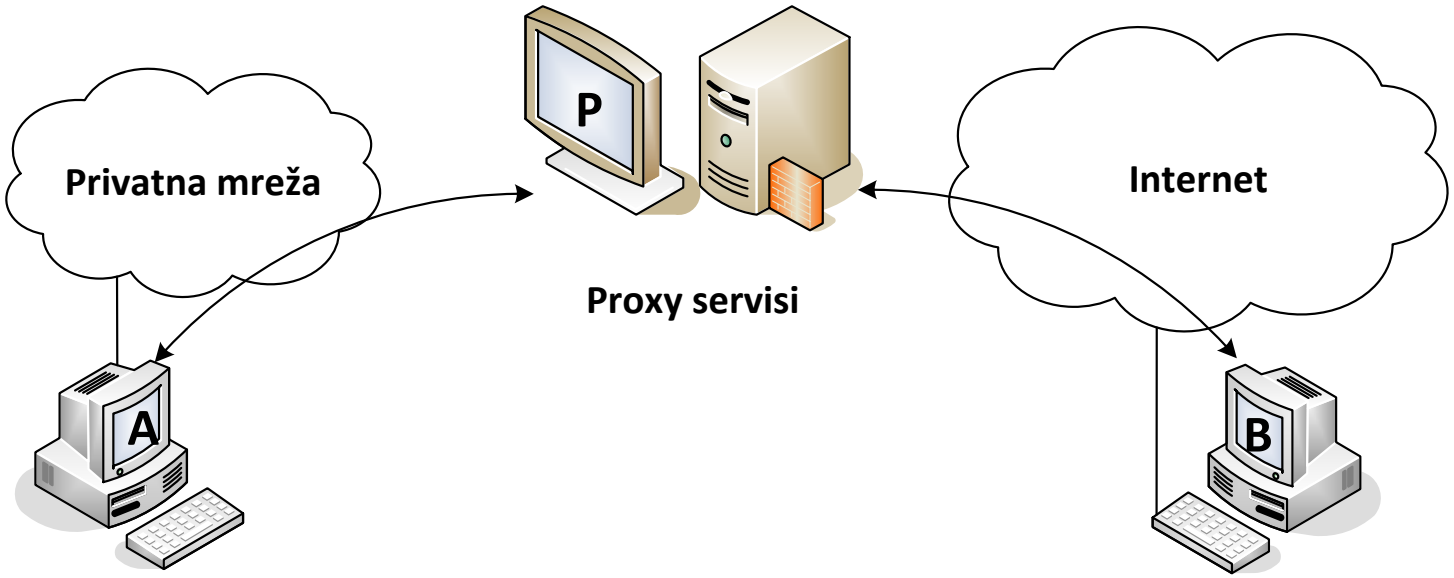


Proksi serveri

- Proksi aplikativnog sloja je klijent-server arhitektura specifična za konkretan protokol koji se koristi.
- Primer: klijent u zaštićenoj mreži i server na javnoj mreži.
  - **Serverski deo proksija** prihvata veze klijenata unutrašnje mreže.
  - **Klijentski deo proksija** povezuje se na javni server.
  - Funcionisanje:
    - Klijent zaštićene mreže inicira zahtev prema serveru javne mreže.
    - Proksi server preuzima taj zahtev.
    - Proksi se povezuje se na server javne mreže u ime klijenta zaštićene mreže.
    - Klijentski deo prima podatke od javnog servera.
    - Serverska strana proksi aplikacije šalje podatke krajnjem klijentu na unutrašnjoj mreži.

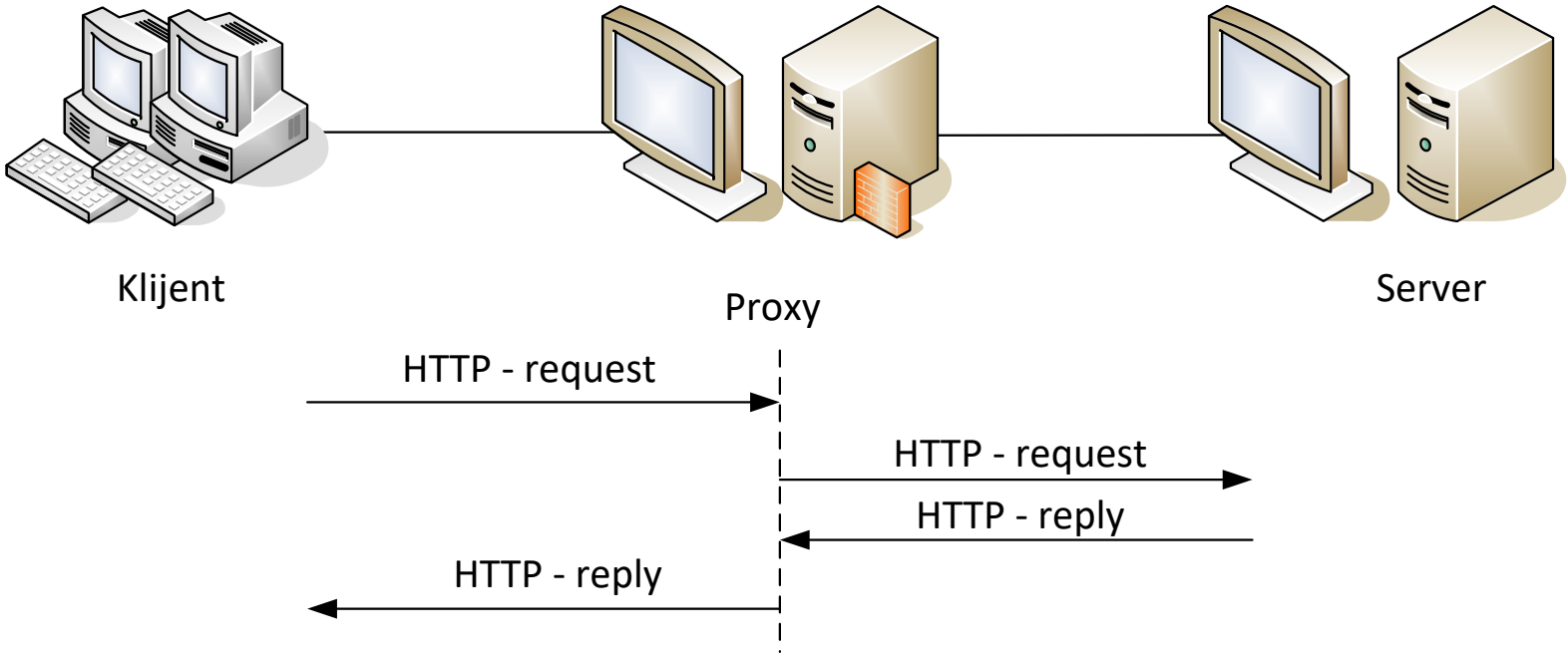
Proksi serveri

- Proksi serveri su specifični po tome što su namenjeni konkretnim protokolima.
- To znači da za različite protokole morate imati različite proksi servere.
- Tipični proksi agenti: DNS, FTP, HTTP, HTTPS, LDAP, NNTP, SMTP, …



Proksi serveri

- Primer: ako u Firefox čitaču Weba unesete adresu (ili ime) proksi servera, Firefox će slati sve zahteve tom serveru, umesto da uspostavlja direktne veze.



Neki problemi koje firewall-ovi ne mogu rešiti

- **Zaštita servisa** u privatnoj mreži kojima je dozvoljen pristup spolja kroz *firewall*.
  - Primer: iskorišćavanje sigurnosnih propusta u IIS-u kome je dozvoljen pristup na portu 80.
  - Nema zaštite od: *buffer overflow* napada, *SQL injection* napada, …
  - Zašto? Firewall ne analizira sadržaj paketa!
- **Interne pretnje**.
  - Firewall ne može zaštiti mrežu od nezadovoljnog zaposlenog koji ima pristup iznutra.
  - Ukoliko neko na USB fleš memoriji unese *ransomware* za koji antivirusni softver nema potpis, *firewall* neće sprečiti širenje u *ransomware*-a u lokalnoj mreži.
- **Skriveni prolazi**.
  - Tačke u kojima se korisnici privatne mreže mogu povezati na Internet i zaobići *firewall*.
  - Primer: bilo koji korisnik lokalne mreže koji ima USB modem može da uspostavi vezu sa svojim ISP i tako kompletno zaobiđe mrežnu barijeru.

Različiti pristupi filtriranju

- Usluge filtriranja paketa na nivou ISP-a.
- Jedan *firewall* sa javnim serverima u privatnoj mreži.
  - Najjednostavnije rešenje – dve zone (**privatna** i **javna** mreža).
  - Rizično zato što se mora otvoriti put do servera koji se nalaze u privatnoj mreži.
- Jedan *firewall* sa javnim serverima van privatne mreže.
  - Postoji rizik od napada na nezaštićene servere.
- Demilitarizovane zone.
  - Tri zone: **Internet**, **DMZ** (javni deo privatne mreže) i **lokalna mreža**.
  - Jedan *firewall* koji na različit način štiti servere u DMZ i računare u lokalnoj mreži.
  - Dva *firewall*-a od kojih jedan štiti servere, drugi lokalnu mrežu.
- Korporativna mrežna barijera.
  - Proizvodi koji centralnu politiku upravljanja mrežnim barijerama distribuiraju na više uređaja.
- Isključenje sa mreže.

iptables

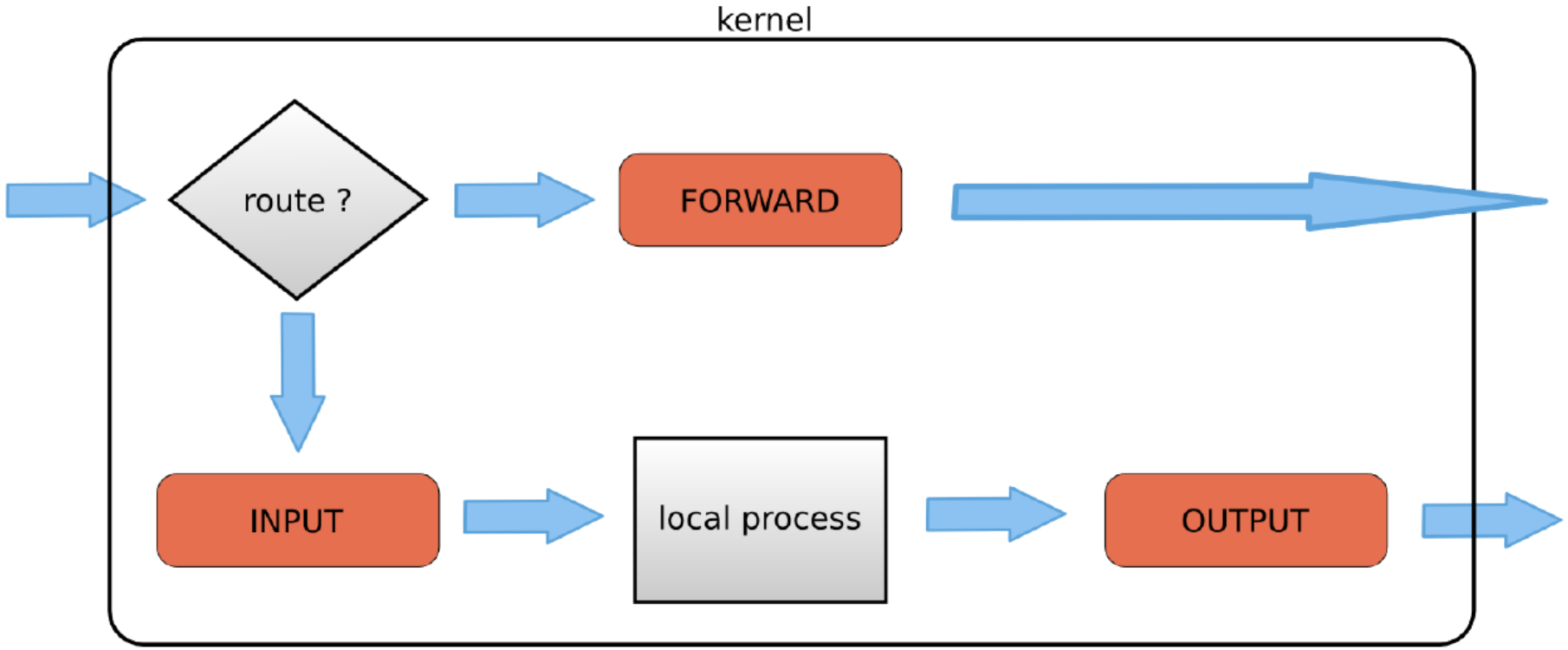
- U jezgru Linux OS nalaze se tri tabele:
  - Tabela `filter` se koristi za filtriranje paketa.
  - Tabela `nat` se koristi za prevođenje mrežnih adresa.
  - Tabela `mangle` služi za postavljanje specifičnih svojstava paketa.
- Skup pravila u svakoj tabeli naziva se **lanac pravila**.
- Pokretanje i zaustavljanje iptables servisa na Red Hat/Fedora/CentOS distribucijama:

```
[root@nihilist ~]# service iptables stop
[root@nihilist ~]# service iptables start
iptables: Applying firewall rules                  [ ok ]
[root@nihilist ~]#
```

iptables – filtriranje paketa

- Tabela `filter` sadrži tri **lanca pravila** (engl. *chains*): `INPUT`, `OUTPUT` i `FORWARD`.
- Svaki lanac sadrži pravila (engl. *rules*).
  - Pravila se primenjuju jedno za drugim na svaki paket koji prolazi kroz lanac.
  - Ako paket zadovolji neko pravilo izvršava se **akcija** (na primer, prihvatanje paketa).
  - Ako paket ne zadovolji pravilo **prosleđuje se dalje** kroz lanac.
  - Ako paket ne zadovolji ni jedno pravilo u lancu primenjuje se **polisa** na kraju lanca.
- Kada paket stigne u jezgro najpre se analizira izvorišna i odredišna adresu paketa.
- Na osnovu toga se odlučuje kom lancu se predaje paket.
- Funkcije lanaca:
  - Lanac `INPUT` je zadužen za dolazeće pakete namenjene lokalnim procesima.
  - Lanac `OUTPUT` je zadužen za pakete odlazeće koji potiču od lokalnih procesa.
  - Lanac `FORWARD` se primenjuje na dolazeće pakete koji nisu namenjeni lokalnim procesima.
    - U ovom slučaju se paket rutira kroz mrežu.

iptables – filtriranje paketa



* Slika preuzeta iz [2]

iptables – filtriranje paketa

- **Pregledanje tabele**.
- Pravila u filtarskim lancima možete pogledati na sledeći način:

```
[root@nihilist ~]# iptables -t filter -nL
Chain INPUT (policy ACCEPT)
target     prot opt    source     destination
Chain FORWARD (policy ACCEPT)
target     prot opt    source     destination
Chain OUTPUT (policy ACCEPT)
target     prot opt    source     destination
[root@nihilist ~]#
```

- Podrazumevano, sva tri lanca filtarske tabele dozvoljavaju prolaz svih paketa.
- To znači da je polisa na kraju lanca `ACCEPT`.

iptables – filtriranje paketa

- **Promena polise**.
- Polisu na kraju lanca možete promeniti da odbaci sve pakete koji ne zadovolje ni jedno pravilo.
- Ovo je najsigurniji način filtriranja.
- Odbačen paket (DROP) ne nastavlja put kroz lanac niti se šalje bilo kakvo obaveštenje o tome.
- Ovo ne treba izvoditi preko **ssh** sesije – tako ćete sami sebi ukidate pristup računaru!

```
[root@nihilist ~]# iptables -P INPUT DROP
[root@nihilist ~]# iptables -P OUTPUT DROP
[root@nihilist ~]# iptables -P FORWARD DROP
[root@nihilist ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt    source     destination
Chain FORWARD (policy DROP)
target     prot opt    source     destination
Chain OUTPUT (policy DROP)
target     prot opt    source     destination
```

iptables – filtriranje paketa

- **Dozvola korišćenja *loopback* interfejsa**.
- Da bi smo mogli da pristupimo servisima na *localhost*-u potrebno je da se dozvoli korišćenje *loopback* interfejsa.
  - U `INPUT` lanac se dodaju pravilo da se dozvole paketi koji su namenjeni *loopback* interfejsu (označen kao `lo`).
  - U `OUTPUT` lanac se dodaju se pravilo da se dozvole paketi koji potiču sa *loopback* interfejsa.

```
[root@nihilist ~]# iptables -A INPUT -i lo -j ACCEPT
[root@nihilist ~]# iptables -A OUTPUT -o lo -j ACCEPT
```

iptables – filtriranje paketa

- **Dozvola pristupa SSH servisu**.
- Dozvolite pristup SSH servisu preko interfejsa `eth0`.
- Napomena: morate da dozvolite i odgovor na zahtev (trenutno radi kao *stateless* filtar).

```
[root@nihilist ~]# iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
[root@nihilist ~]# iptables -A OUTPUT -o eth0 -p tcp --sport 22 -j ACCEPT
```

- **Dozvola pristupa sa podmreže**.
- Dozvolite pristup sa bilo kog računara koji pripada podmreži 10.1.1.0/24 preko interfejsa `eth1` bez obzira na broj porta.

```
[root@nihilist ~]# iptables -A INPUT -i eth1 -s 10.1.1.0/24 -p tcp -j ACCEPT
[root@nihilist ~]# iptables -A OUTPUT -o eth1 -d 10.1.1.0/24 -p tcp -j ACCEPT
```

iptables – filtriranje paketa

- Kako sada izgleda filtarska tabela?

```
[root@nihilist ~]# iptables -nvL
Chain INPUT (policy DROP 7 packets, 609 bytes)
pkts  bytes  target  prot  opt  in    out   source       destination
0     0      ACCEPT  all   --   lo    *     0.0.0.0/0    0.0.0.0/0
0     0      ACCEPT  tcp   --   eth0  *     0.0.0.0/0    0.0.0.0/0 tcp dpt:22
0     0      ACCEPT  tcp   --   eth1  *     10.1.1.0/24  0.0.0.0/0
Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts  bytes  target  prot  opt  in    out   source       destination
Chain OUTPUT (policy DROP 3 packets, 228 bytes)
pkts  bytes  target  prot  opt  in    out   source       destination
0     0      ACCEPT  all   --   *     lo    0.0.0.0/0    0.0.0.0/0
0     0      ACCEPT  tcp   --   *     eth0  0.0.0.0/0    0.0.0.0/0 tcp spt:22
0     0      ACCEPT  tcp   --   *     eth1  0.0.0.0/0    10.1.1.0/24
```

iptables – filtriranje paketa

- **Dozvola ICMP protokola (ping)**.
- Ako nakon tekuće konfiguracija pokrenete `iptables` servis i pokušate da izvršite komandu `ping`:

```
[root@nihilist ~]# ping 192.168.187.130
PING 192.168.187.130 (192.168.187.130) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

- Dozvolite slanje ICMP echo-request paketa sa vašeg računara i primanje odgovora.

```
[root@nihilist ~]# iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
[root@nihilist ~]# iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
```

- Dozvolite prosleđivanje i rutiranje ICMP paketa:

```
[root@nihilist ~]# iptables -A FORWARD -p icmp --icmp-type any -j ACCEPT
```

iptables – filtriranje paketa

- **Primer filtra sa uspostavom stanja.**
- Web serveru se direktno pristupa preko javne IP adrese 66.10.10.1.
- Potrebno je omogućiti HTTP, HTTPS i SSH pristup server i „pingovanje“ servera.
- Deo konfiguracionog skripta 1/3 – inicijalizacija.

```
#!/bin/bash
# Izbrišite sva pravila i postavite restriktivnu podrazumevanu polisu:
iptables --flush
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
# Podesite loopback interfejs:
iptables -A INPUT  -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

iptables – filtriranje paketa

- **Primer filtra sa uspostavom stanja.**
- Deo konfiguracionog skripta 2/3 – elementarna zaštita.

```
# Sprečite elementarno lažiranje izvorišnih IP adresa:
iptables -A INPUT -s 255.0.0.0/8 -j DROP
iptables -A INPUT -s 0.0.0.0/8 -j DROP
iptables -A INPUT -s 127.0.0.0/8 -j DROP
iptables -A INPUT -s 192.168.0.0/16 -j DROP
iptables -A INPUT -s 172.16.0.0/12 -j DROP
iptables -A INPUT -s 10.0.0.0/8 -j DROP
# Sprečite napade tipa „Stealth Scan“ (konekcije uvek počinju SYN paketima):
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

iptables – filtriranje paketa

- **Primer filtra sa uspostavom stanja.**
- Deo konfiguracionog skripta 3/3.

```
# Dozvolite serveru da nastavi već uspostavljenu konekciju
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# Dozvolite serveru da primi pakete koji započinju novu SSH, HTTP ili HTTPS konekciju
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT
# Dozvolite serveru da primi dolazeće ICMP echo request pakete i odgovori na njih
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT --icmp-type echo-reply
# Ostalo ubeležite u dnevnik aktivnosti i odbacite.
iptables -A INPUT -j LOG --log-prefix "Odbaceno INPUT lancem“
iptables -A OUTPUT -j LOG --log-prefix "Odbaceno OUTPUT lancem"
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
```

iptables – filtriranje paketa

- **Zaštita od nekih vrsta napada**.
- **Lažiranje izvorišne IP adrese** (engl. *Source IP spoofing*).
  - Odbrana: odbacite sve pakete koji stižu na javni interfejs a imaju adresu lokalne mreže.

```
[root@nihilist ~]# iptables -A FORWARD -s intranet_IP -i public_interface -j DROP
```

- **Podmetanje žrtve** (engl. *smurf attack*).
  - Napadač šalje broadcast ICMP echo-request pakete računarima u intranetu a kao izvorišnu adresu navodi adresu žrtve. Žrtva trpi bombardovanje echo-reply paketima.
  - Odbrana: odbaciti sve neusmerene echo-request pakete

```
[root@nihilist ~]# iptables -A FORWARD -p icmp --icmp-type echo-request
                            -d intranet_broadcast -j DROP
```

iptables – filtriranje paketa

- **Zaštita od nekih vrsta napada**.
- **Bombardovanje SYN paketima** (engl. *SYN-flood*).
  - Slanje velikog broja SYN paketa (engl. *TCP connection request*).
  - Odbrana: ograničite broj SYN paketa u jedinici vremena.

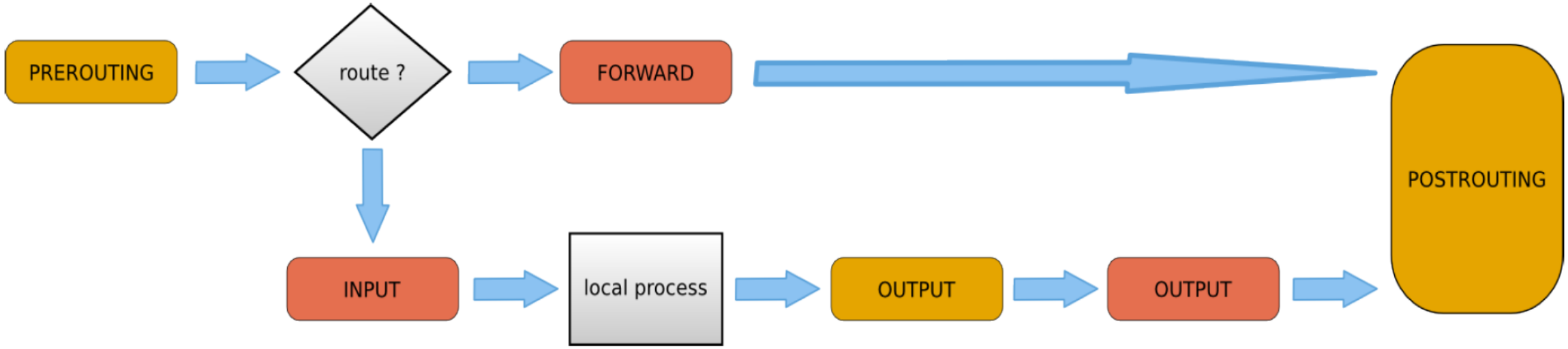
```
[root@nihilist ~]# iptables -A FORWARD -p tcp -i public_interface -syn
                            -m limit -limit 1/s -j ACCEPT
```

- **Skeniranje portova**.
  - Od ovog napada se ne možete potpuno zaštititi → „smorite“ napadača (dozvolite ograničen broj SYN, ACK, FIN i RST paketa u jedinici vremena).

```
[root@nihilist ~]# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST
                            -i public_interface -m limit --limit 1/s -j ACCEPT
```

iptables – prevođenje mrežnih adresa

- **NAT** tabela sadrži dva lanca pravila:
- Lanac `PREROUTING` služi za izmenu zaglavlja paketa pre ulaska u `INPUT` lanac.
- Lanac `POSTROUTING` služi za izmenu zaglavlja paketa nakon izlaska iz `OUTPUT` lanca.



* Slika preuzeta iz [2]

iptables – prevođenje mrežnih adresa

- **SNAT** (engl. *source NAT*) služi za izmenu izvorišne adrese paketa pre nego što napusti sistem.
  - Odgovor na zahtev vraća se NAT uređaju.
  - To znači da NAT uređaj mora da čuva tabelu prevođenja adresa u memoriji kako bi odgovor prosledio odgovarajućem računaru u lokalnoj mreži.
- SNAT se bavi paketima koji napuštaju sistem i zato koristi `POSTROUTING` lanac.
- Primer SNAT pravila:
  - Paketima koji dolaze sa mreže 10.1.1.0/24 i napuštaju uređaj preko interfejsa `eth1` biće dodeljena izvorišna IP adresa 11.12.13.14.

```
iptables -t nat -A POSTROUTING -o eth1 -s 10.1.1.0/24 -j SNAT --to-source 11.12.13.14
```

- NAPOMENA: da bi ovo funkcionisalo neophodno je u filtarskim lancima dozvoliti prolaz paketa sa jedne mreže na drugu!

iptables – prevođenje mrežnih adresa

- **Primer SNAT prevođenja**.
- Računari u internoj mreži (interfejs `eth0`) preko SNAT-a pristupaju Web serverima (portovi 80 i 443) koji se nalaze u spoljnoj mreži (interfejs `eth1`).
- Interfejs `eth1` ima statičku (fiksnu) IP adresu.

```
#!/bin/bash
echo 0 > /proc/sys/net/ipv4/ip_forward
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A FORWARD -i eth0 -o eth1 -s 10.1.1.0/24 -p tcp --dport 80, 443 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -d 10.1.1.0/24 -p tcp --sport 80, 443 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth1 -s 10.1.1.0/24 -j SNAT --to-source 11.12.13.14
echo 1 > /proc/sys/net/ipv4/ip_forward
```

iptables – prevođenje mrežnih adresa

- **IP masquerading**.
- Koristi se u slučaju da ISP dodeljuje javnu dinamičku (promenljivu) IP adresu interfejsu.
- Prethodni primer izmenjen za slučaj da interfejs `eth1` dobija dinamičku adresu:

```
#!/bin/bash
echo 0 > /proc/sys/net/ipv4/ip_forward
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A FORWARD -i eth0 -o eth1 -s 10.1.1.0/24 -p tcp --dport 80, 443 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -d 10.1.1.0/24 -p tcp --sport 80, 443 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth1 -s 10.1.1.0/24 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

iptables – prevođenje mrežnih adresa

- **DNAT** (engl. *destination NAT*) se koristi za redirekciju paketa koji potiču sa Interneta ka serveru u unutrašnjoj mreži ili DMZ (adresni opseg koji nije direktno dostupan sa Interneta).
- Primer: korisnici sa Interneta imaju SSH pristup serveru u unutrašnjoj mreži (192.168.1.99).

```
#!/bin/bash
# eth0 je unutrašnja mreža, eth1 je Internet
echo 0 > /proc/sys/net/ipv4/ip_forward
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A FORWARD -i eth0 -o eth1 -s 10.1.1.0/24 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -p tcp --dport 22 -j ACCEPT
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 22 -j DNAT --to-destination 10.1.1.99
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Uncomplicated FireWall (ufw)

- Komanda `iptables` je osnovni alat za konfigurisanje mrežne barijere u jezgru Linux OS.
- Složenost komande inicirala je razvoj dodatnih alata koji pojednostavljuju taj zadatak.
- Jedan od pomenutih alata je `ufw`.
  - Alat `ufw` ne obezbeđuje kontrolu svih mogućih opcija mrežne barijere.
  - Namenjen je dodavanju i uklanjanju jednostavnijih pravila.
- Podrazumevano, `ufw` je isključen i pre upotrebe ga treba uključiti komandom: `sudo ufw enable`
- **Primeri** upotrebe alata `ufw`.
  - Dozvola SSH pristupa:
    ```
    sudo ufw allow 22
    ```
  - Dodavanje numerisanog pravila koje dozvoljava pristup portu 80:
    ```
    sudo ufw insert 1 allow 80
    ```
  - Zabrana SSH pristupa:
    ```
    sudo ufw deny 22
    ```
  - Brisanje prethodno unetog pravila:
    ```
    sudo ufw delete deny 22
    ```

Uncomplicated FireWall (ufw)

- **Primeri** upotrebe alata `ufw`.
  - Dozvola SSH pristupa sa IP adrese 192.168.0.2 ka bilo kojoj adresi na računaru (pretpostavlja se da računar ima više mrežnih interfejsa):
    ```
    sudo ufw allow proto tcp from 192.168.0.2 to any port 22
    ```
  - Dozvola SSH pristupa sa mreže 192.168.0.0/24 ka bilo kojoj adresi na računaru (pretpostavlja se da računar ima više mrežnih interfejsa):
    ```
    sudo ufw allow proto tcp from 192.168.0.0/24 to any port 22
    ```
  - Prikazivanje pravila nakon dozvole pristupa HTTP portu koje ufw generiše (pravila se samo prikazuju na ekranu ali se ne primenjuju – takozvani „*dry run*“):
    ```
    sudo ufw --dry-run allow http
    ```
  - Prikazivanje `ufw` statusa:
    ```
    sudo ufw status
    ```
  - Uključivanje vođenja dnevnika aktivnosti:
    ```
    sudo ufw logging on
    ```

Uncomplicated FireWall (ufw)

- **Konfigurisanje ufw na osnovu profila aplikacija i mrežnih servisa**.
- Neki servisi sadrže takozvani ufw profil koji sadrže popis portova kojima treba dozvoliti pristup kako bi aplikacija ispravno funkcionisala.
- Na osnovu tih profila moguće je još lakše konfigurisati firewall.
- Primeri korišćenja profila:
  - Pregledanje aplikacija koje su instalirale `ufw` profil:
    `sudo ufw app list`
  - Dozvola pristupa portovima koji su neophodni za ispravan rad servisa Samba:
    `sudo ufw allow Samba`
  - Dozvola pristupa portovima koji su neophodni za ispravan rad servisa Samba sa mreže 192.168.0.0/24 ka bilo kojoj adresi na računaru (računar ima više mrežnih interfejsa):
    `ufw allow from 192.168.0.0/24 to any app Samba`
  - Pregledanje informacijama o portovima, protokolima i ostalim pravilima filtriranja definisanim profilom servisa Samba:
    `sudo ufw app info Samba`

Skeniranje portova

- Skeniranje portova je:
  - Metoda za proveru konfiguracije mrežne barijere (za administratore).
  - Izviđački napad (za etičke hakere i pen-testere).
- **Klasično skeniranje**.
  - Napadač šalje SYN pakete opsegu portova žrtve.
  - Svi portovi za koje žrtva odgovori SYN+ACK paketom su otvoreni.
  - Ukoliko želite da žrtva vaš pokušaj skeniranja ne zapiše u dnevnik događaja pošaljite joj RST paket kojim se veza raskida (većina servera ne beleži ovakve događaje u dnevnik).
- **TCP FIN (*stealth*) skeniranje**.
  - Napadač šalje FIN pakete opsegu portova žrtve.
  - Svi zatvoreni portovi zbog greške u implementaciji TCP protokola odgovaraju RST paketom.
  - Otvoreni portovi ne odgovaraju.
- ***Fingerprinting***.
  - Različiti operativni sistemi različito reaguju na loše formatirane TCP pakete.
  - Ova metoda se koristi za određivanje tipa operativnog sistema na računaru koji je meta.

Skeniranje portova alatom nmap

- **Horizontalno izviđanje** je proces u kome napadač ispituje koji se mrežni uređaji i računari nalaze u računarskoj mreži i koja je njena topologija.
  - Ovakvo izviđanje se može izvesti na različite načine ali je njegov najjednostavniji oblik korišćenje ICMP protokola, odnosno tzv. „pingovanje“ svih IP adresa iz opsega koje koristi računarska mreža na koju se vrši napad.
  - Primer:

```
[root@nihilist ~]# nmap -sP 192.168.1.*
Starting Nmap 5.51 ( http://nmap.org ) at 2012-08-28 11:11 CEST
Nmap scan report for 192.168.1.7
Host is up (0.00012s latency).
MAC Address: 00:0D:60:32:EA:3E (IBM)
...
```

Skeniranje portova alatom nmap

- **Vertikalno izviđanje** ima za cilj da napadaču pruži informaciju o tome koji su portovi na pojedinačnim mrežnim uređajima (najčešće računarima) otvoreni, odnosno koji se mrežni servisi na njima izvršavaju.
  - Primer:

```
[root@nihilist ~]# nmap 192.168.1.122 -p 1-65535
Nmap scan report for 192.168.1.122
Host is up (0.00024s latency).
Not shown: 65533 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
Nmap done: 1 IP address (1 host up) scanned in 336.15 seconds
```

Skeniranje portova alatom nmap

- **Dubinsko izviđanje** ima za cilj da napadaču pruži što detaljnije informacije o softveru koji se izvršava na određenom mrežnom uređaju (tip, verzija i podešavanje softvera, a odnose se i na sistemski i na korisnički softver.)
  - Ove informacije imaju najviši značaj jer na osnovu njih napadač otkriva potencijalne bezbednosne rupe koje može iskoristiti za postizanje ciljeva napada.
  - Primer (utvrđivanje verzije operativnog sistema):

```
[root@nihilist ~]# nmap -O 192.168.1.122
Nmap scan report for 192.168.1.122
Host is up (0.00022s latency).
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.19 - 2.6.36
Network Distance: 1 hop
OS detection performed.
Nmap done: 1 IP address (1 host up) scanned in 12.07 seconds
```

Neki primeri upotrebe alata nmap

- Skeniranje jednog hosta na osnovu imena ili IP adrese.

```
nmap 192.168.1.1
nmap server1.viser.edu.rs
## detaljnije informacije
nmap -v server1.viser.edu.rs
```

- Skeniranje većeg broja adresa ili podmreže.

```
nmap 192.168.1.1 192.168.1.2 192.168.1.3
nmap 192.168.1.1,2,3
nmap 192.168.1.1-20
nmap 192.168.1.*
nmap 192.168.1.0/24
## Adrese, hostovi ili opsezi su upisani u datoteci /tmp/test.txt (po jedan u redu)
nmap –iL /tmp/test.txt
## Dodavanje izuzetaka
nmap 192.168.1.0/24 --exclude 192.168.1.5,192.168.1.254
nmap -iL /tmp/scanlist.txt --excludefile /tmp/exclude.txt
```

Neki primeri upotrebe alata nmap

- Otkrivanje da li je host ili mreža zaštićena firewall-om.

```
nmap -sA 192.168.1.254
nmap -sA server1.viser.edu.rs
```

- Skeniranje hosta zaštićenog firewall-om.

```
nmap -PN 192.168.1.1
nmap -PN server1.viser.edu.rs
```

- Skeniranje IPv6 hosta / adrese

```
nmap -6 server1.viser.edu.rs
nmap -6 2607:f0d0:1002:51::4
```

- Pronalaženje živih hostova (*host discovery*, *ping scan*).

```
nmap -sP 192.168.1.0/24
```

- Prikazivanje razloga zbog kog se određni port nalazi u datom stanju.

```
nmap --reason 192.168.1.1
nmap --reason server1.viser.edu.rs
```

Neki primeri upotrebe alata nmap

- Skeniranje određenih portova ili opsega portova.

```
## Skeniraj TCP port 80
nmap -p T:80 192.168.1.1
## Skeniraj UDP port 53
nmap -p U:53 192.168.1.1
## Skeniraj dva porta
nmap -p 80,443 192.168.1.1
## Skeniraj opseg portova
nmap -p 80-200 192.168.1.1
## Kobinovanje opcija
nmap -p U:53,111,137,T:21-25,80,139,8080 192.168.1.1
nmap -p U:53,111,137,T:21-25,80,139,8080 server1.viser.edu.rs
nmap -v -sU -sT -p U:53,111,137,T:21-25,80,139,8080 192.168.1.254
```

Neki primeri upotrebe alata nmap

- Detekcija operativnog sistema.

```
nmap -O 192.168.1.1
nmap -O  --osscan-guess 192.168.1.1
nmap -v -O --osscan-guess 192.168.1.1
```

- Određivanje verzija udaljenih servisa.

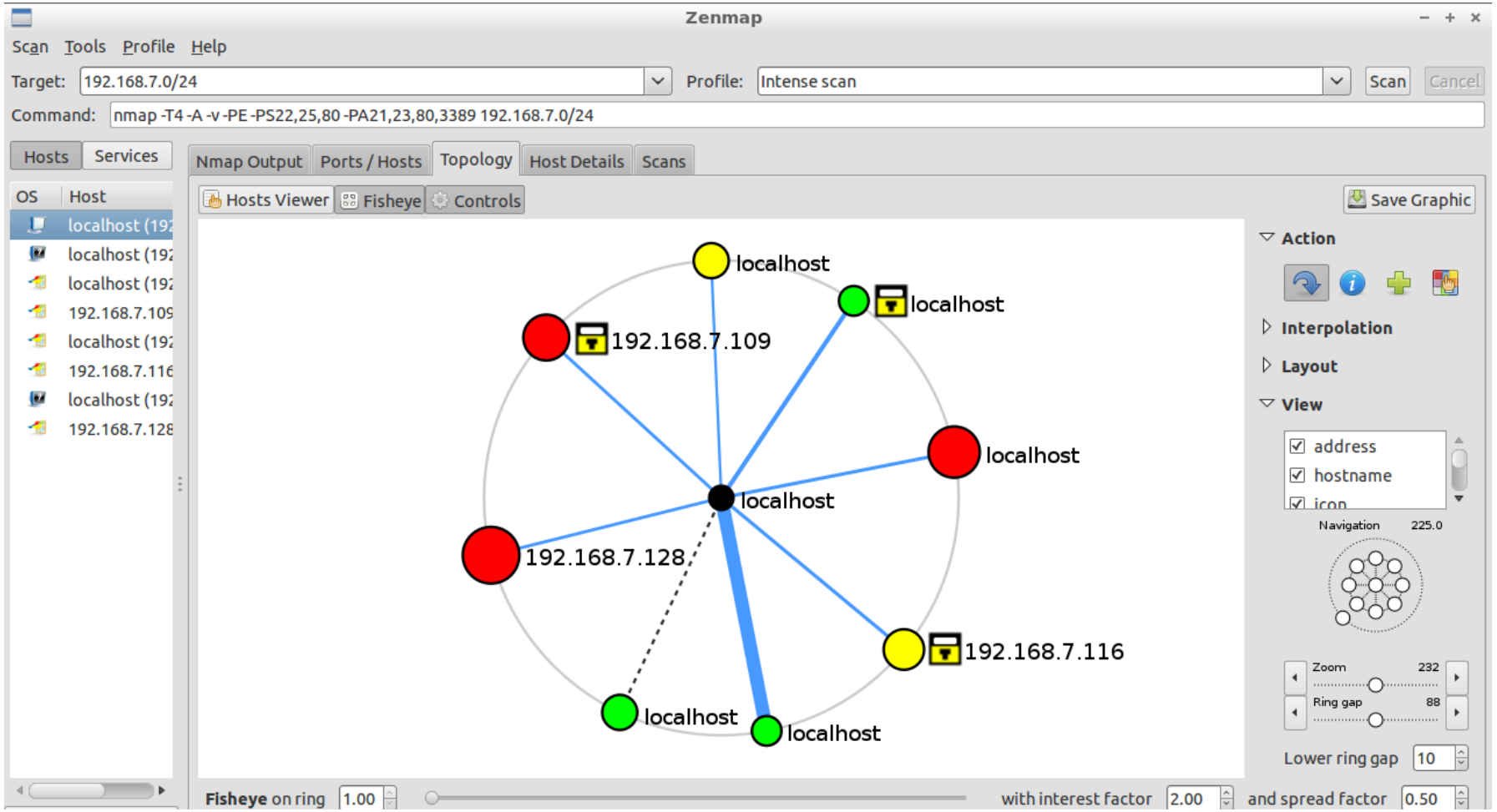
```
nmap -sV 192.168.1.1
# primer izlaza
Starting Nmap 5.00 ( http://nmap.org ) at 2012-11-27 01:34 IST
Interesting ports on 192.168.1.1:
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     Dropbear sshd 0.52 (protocol 2.0)
80/tcp open  http?
1 service unrecognized despite returning data.
```

Neki primeri upotrebe alata nmap

- Tipovi skeniranja.

```
## Stealthy scan
nmap -sS 192.168.1.1
## TCP connect scan
nmap -sT 192.168.1.1
## TCP ACK scan
nmap -sA 192.168.1.1
## TCP Window scan
nmap -sW 192.168.1.1
## TCP Maimon scan
nmap -sM 192.168.1.1
## UDP scan
nmap -sU 192.168.1.1
```

Zenmap


Zenmap
Scan Tools Profile Help
Target: 192.168.7.0/24
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v -PE -PS22,25,80 -PA21,23,80,3389 192.168.7.0/24
Hosts
Services
OS
Host
localhost (192
localhost (192
localhost (192
192.168.7.109
localhost (192
192.168.7.116
localhost (192
192.168.7.128
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Hosts Viewer
Fisheye
Controls
Save Graphic
localhost
localhost
192.168.7.109
localhost
localhost
192.168.7.128
192.168.7.116
localhost
localhost
Action
Interpolation
Layout
View
address
hostname
icon
Navigation
225.0
Zoom
232
Ring gap
88
Lower ring gap
10
Fisheye on ring
1.00
with interest factor
2.00
and spread factor
0.50

Literatura

1. D. Pleskonjić, N. Maček, B. Đorđević, M. Carić (2007): Sigurnost računarskih sistema i mreža. Mikro knjiga, Beograd.
2. P. Cobbaut (2015): Linux Security.
3. A. Jevremović, M. Veinović, M. Šarac, G. Šimić (2014): Zaštita u računarskim mrežama. Univerzitet Singidunum, Beograd.

Hvala na pažnji

Pitanja su dobrodošla.