Hálózatok építése, konfigurálása és működtetése

Hálózati funkciók a gyakorlatban

Hol tartunk?

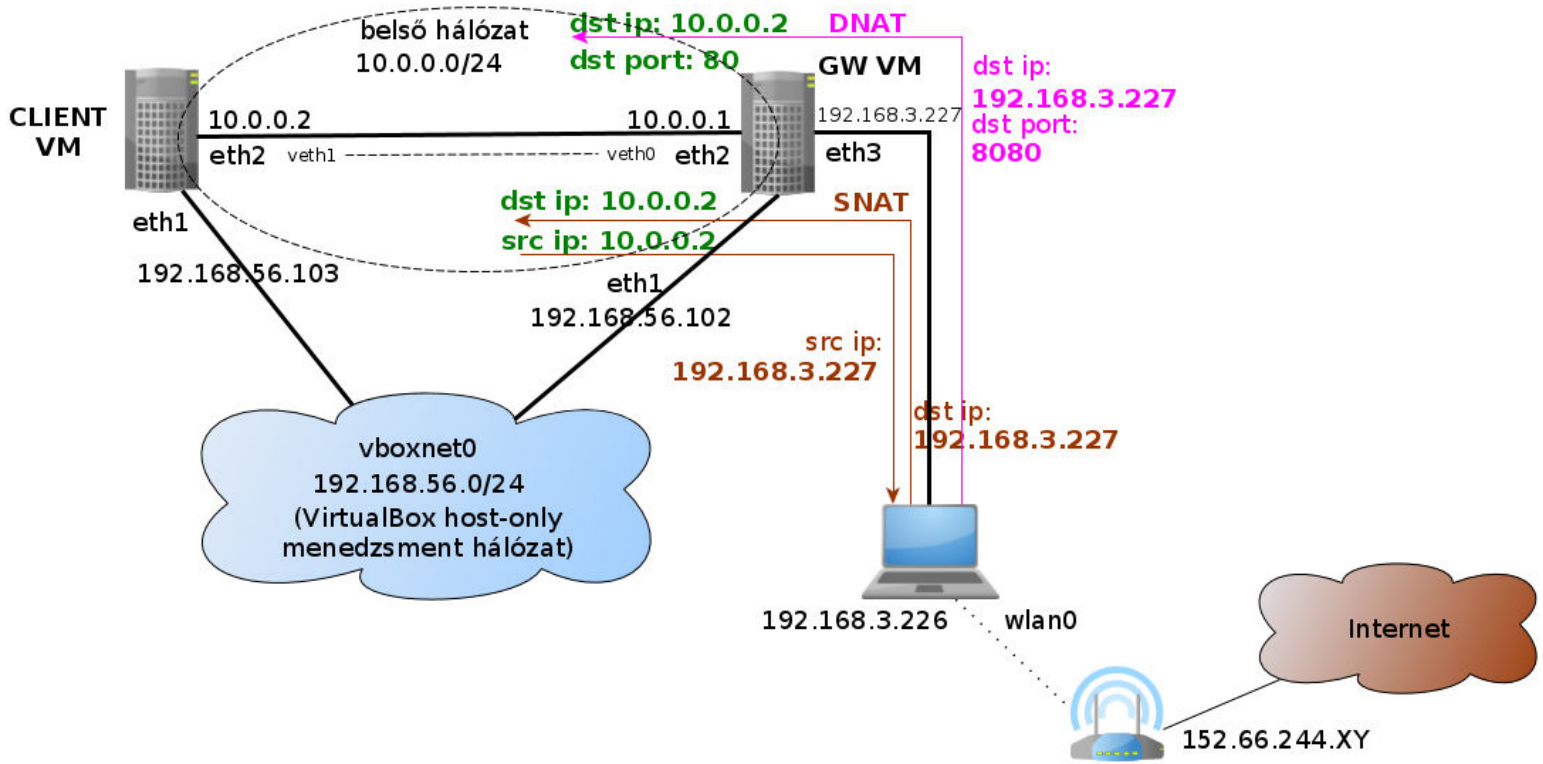
- UNIX/Linux alapok
  - történelem, GNU/Linux rendszerek felépítése, alapvető parancsok, “user mode”, szűrők, Bash alapok…
- Linux hálózatkezelés (és Linux admin alapok)
  - root jogosultság, partíciók, fájlrendszerek, Linux boot folyamata, “service”-ek, csomagok, hálózatkezelés alapok
- Szoftver szerszámok hálózatkezeléshez
  - ifconfig, route, ip (iproute2)
  - ping, traceroute, netstat, tcpdump, wireshark
  - bash, python, scapy
  - python: otthoni feldolgozás (segítség: kiadott összefoglaló!)

Mai téma

- Egyszerű hálózat bekonfigurálása
- egy konkrét példán keresztül
- lépésről-lépésre

A példa hálózatunk

Hálózati elrendezés



Előkészítés

- Virtuális link létrehozása
  - sudo ip link add veth0 type veth peer name veth1
  - sudo ifconfig veth0 up; sudo ifconfig veth1 up
  - virtuális Ethernet pár
    - egyik oldalon BE, másikon KI
- Virtuális gépek konfigurálása (VirtualBox)
  - gateway (GW)
    - eth1: "Host-only Adapter", menedzsment interfész
      - hoszt gépről tudunk belépni egy belső hálózaton
    - eth2: "Bridged Adapter", veth0
      - virtuális link bekötése
    - eth3: "Bridged Adapter", wlan0
      - megkapja a hoszt gép wlan interfészét is bridge módban
  - kliens (CLIENT)
    - eth1: "Host-only Adapter", menedzsment interfész
    - eth2: "Bridged Adapter", veth1

Start!

- Virtuális gépek indítása
  - belépés a menedzsment interfészen
    - ssh -Y mininet@192.168.56.102 (GW)
    - ssh -Y mininet@192.168.56.103 (CLIENT)
  - interfészek manuális konfigurálása
    - (NAT interfész leállítása: sudo ifdown eth0)
    - virtuális összeköttetés a VM-ek között
      - sudo ifconfig eth2 up
    - GW: külső kapcsolat beállítása
      - sudo ifconfig eth3 up
      - sudo dhclient –v eth3
    - közben ellenőrizzük a
      - routing táblát (route -n)
      - névfeloldás beállítását (cat /etc/resolv.conf)

Összeköttetés tesztelése

- eth2 interfészek konfigurálása
  - GW: sudo ifconfig eth2 10.0.0.1/24
  - CLIENT: sudo ifconfig eth2 10.0.0.2/24
  - ping?
    - ping 10.0.0.2 <-> ping 10.0.0.1
  - web?
    - lynx 10.0.0.2
  - ssh?
    - ssh 10.0.0.2

Hogyan tovább?

- Belső hálózatról (CLIENT) szeretnénk "netezni"
- Azért minden forgalmat nem szeretnénk engedélyezni
- Manuális konfiguráció???

Hogyan tovább?

- Belső hálózatról (CLIENT) szeretnénk "netezni"
  - NAT
- Azért minden forgalmat nem szeretnénk engedélyezni

- Manuális konfiguráció???

Hogyan tovább?

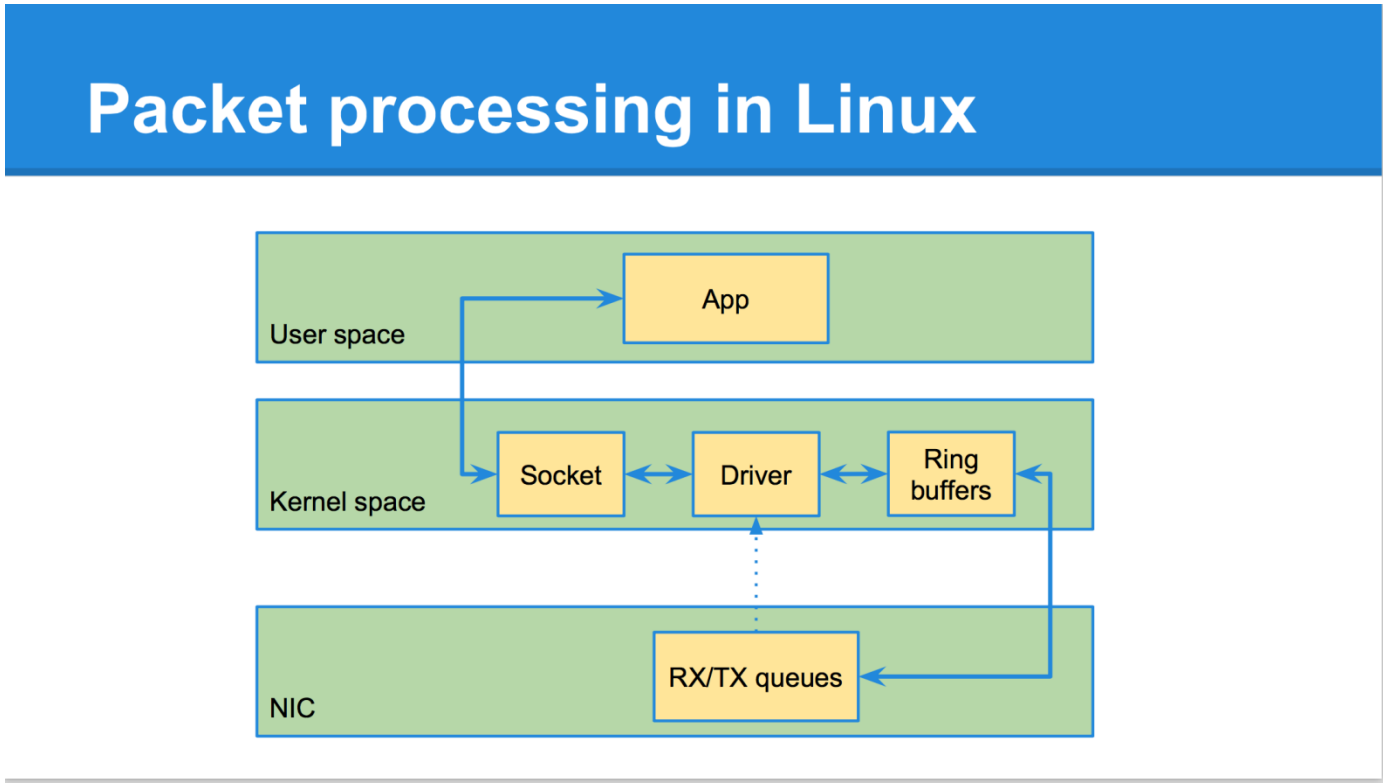
- Belső hálózatról (CLIENT) szeretnénk "netezni"
  - NAT
- Azért minden forgalmat nem szeretnénk engedélyezni
  - Firewall
- Manuális konfiguráció???

Hogyan tovább?

- Belső hálózatról (CLIENT) szeretnénk "netezni"
  - NAT
- Azért minden forgalmat nem szeretnénk engedélyezni
  - Firewall
- Manuális konfiguráció???
  - DHCP, DNS

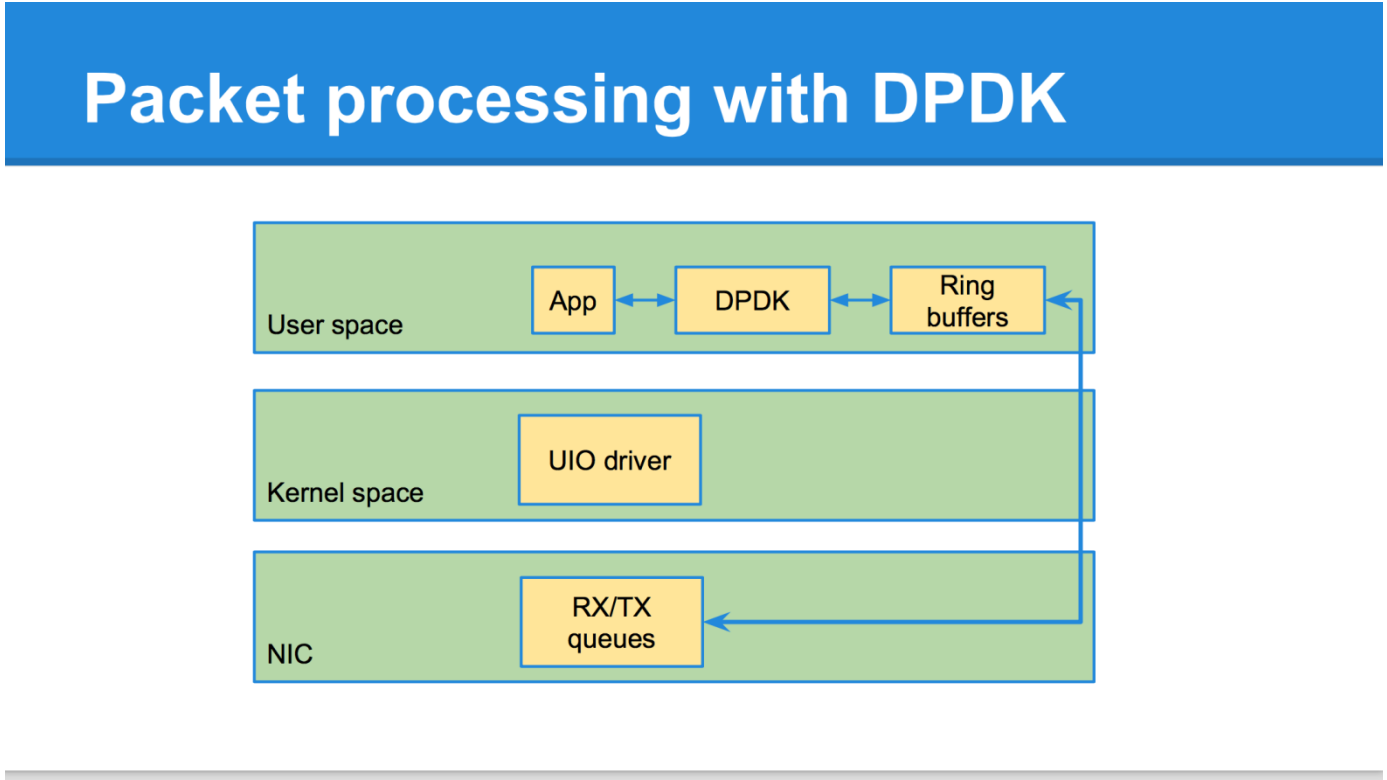
Egy csomag útja a Linux rendszerben

NIC-kernel space-user space



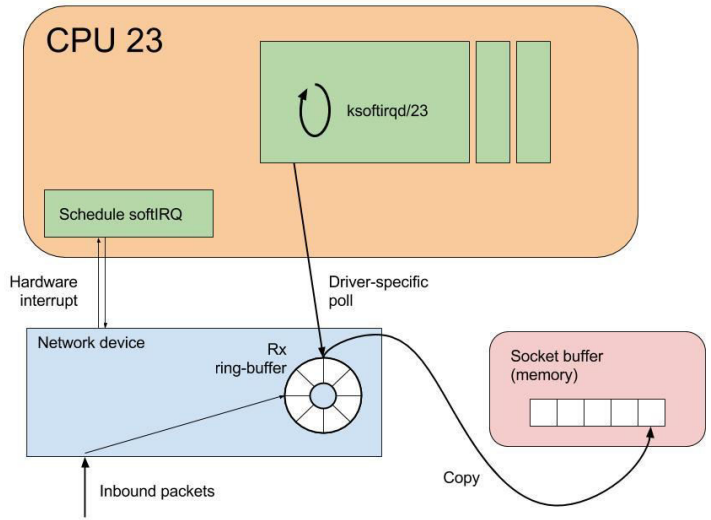
forrás: https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md

(Érdekesség)



forrás: https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md

NAPI – New API (2.6-os kerneltől)



- Régebbi kernel verziókban
  - hardware interrupt alapú működés
  - minden beérkező csomagra IRQ!
  - nem hatékony megoldás
- NAPI
  - poll mode
    - periodikus ellenőrzés
    - egyszerre sok csomag betöltése
  - hw IRQ -> softIRQ ütemezése adott CPU-n
  - feldolgozó processz: ksoftirqd/<cpu-id>
    - megosztható feladatok a core-ok között
  - NIC driver poll függvényét hívja
  - másolás a socket bufferbe
  - ezután jön a teljes network stack

forrás: https://www.lmax.com/blog/staff-blogs/2016/05/06/navigating-linux-kernel-network-stack-receive-path/

Csomagfeldolgozás

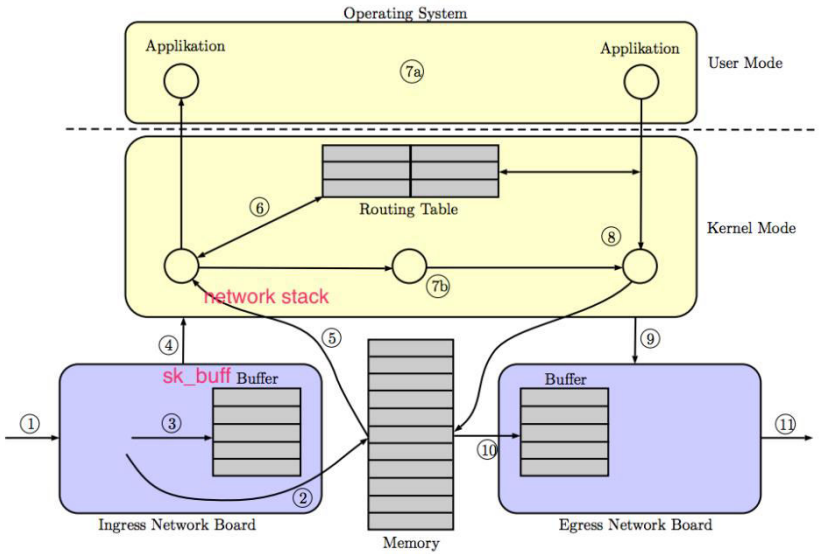


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

- (1) csomag érkezik a hálókártyára
- sk_buff struktúra
  - socket kernel buffer
  - ahogy a kernel reprezentál egy csomagot
- NIC driver ring buffer (3)
  - pointerek az sk_buff struktúrákra
- (2) NIC "be-DMA-zza" a memóriába
- (4) hw IRQ
  - jelzés, hogy van csomag
  - softIRQ ütemezése adott CPU-n
  - NIC hozzáadása a poll_list listához
- (5) ksoftirqd/x:
  - softIRQ handler
  - eszköz "pollozása"
  - minden csomagra a network stack végrehajtása

forrás: https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md

Csomagfeldolgozás

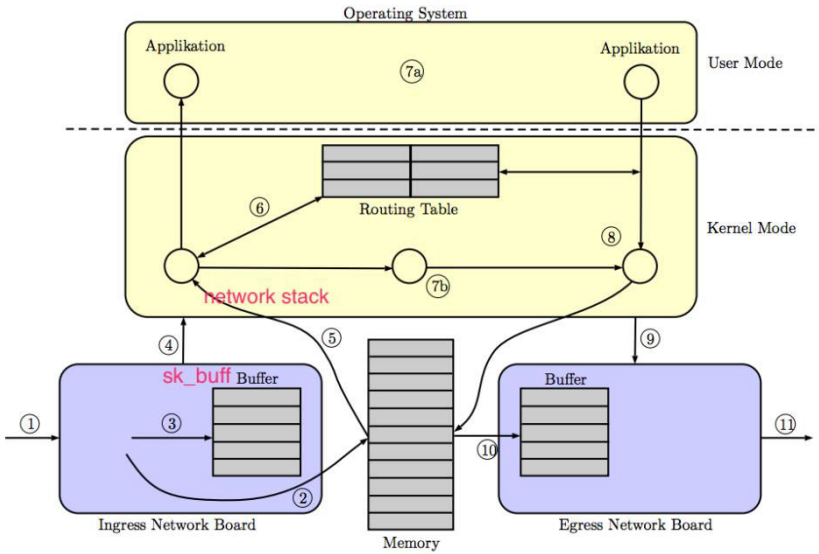


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

- csomagellenőrzés (integrity-verification, cheksum)
  - ha nem OK, eldobja
- **firewall szabályok**
  - routing előtt és után
  - (jön részletesen)
- (6) routing alrendszer
  - ha ide érkezett a csomag
    - tovább a transzport rétegnek
    - (7a) socket API-n keresztül az alkalmazásnak
    - sk_buff -> user space másolás
  - ha nem ide érkezett a csomag
    - (7b) forwarding
    - routing algoritmus, routing tábla alapján

forrás: https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md

Csomagfeldolgozás

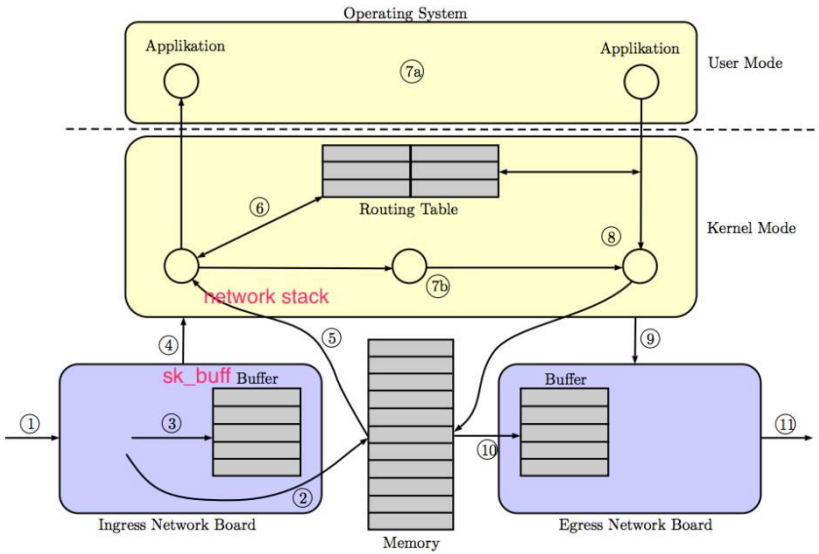
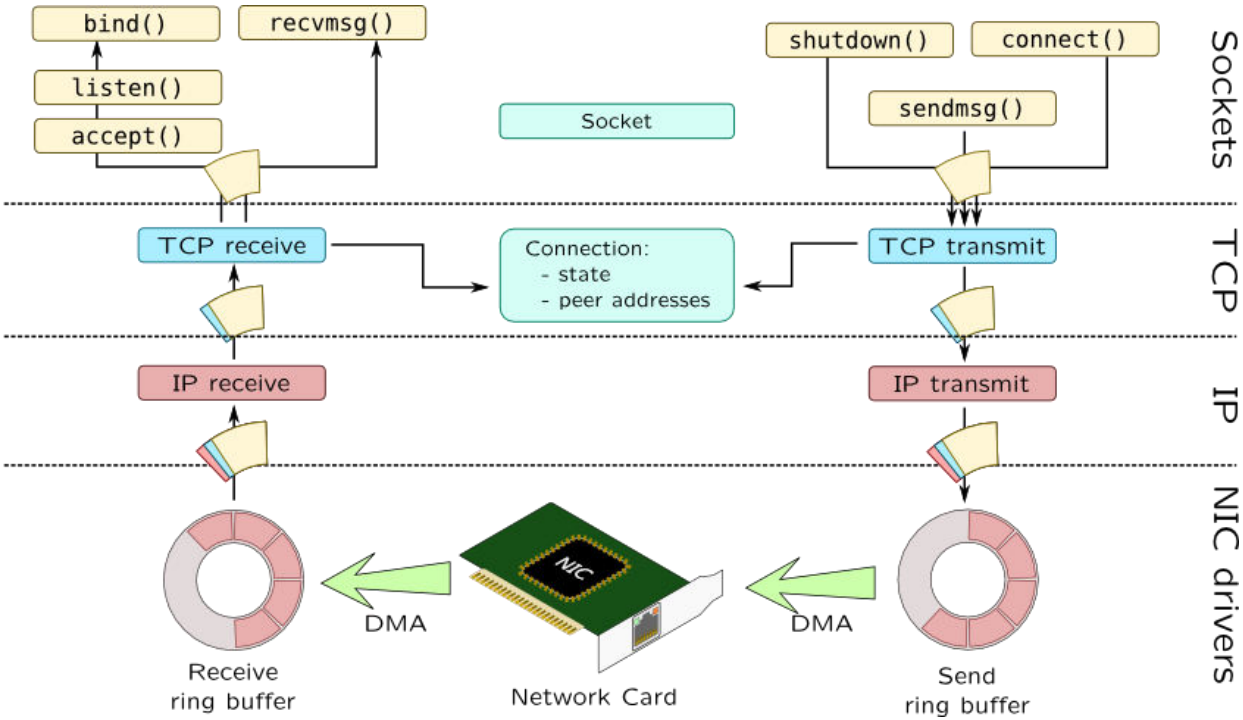


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

- (8) csomag küldése
  - lokális alkalmazás generálta
  - vagy továbbítandó csomag
- next hop mac címét ki kell találni
  - ARP
- (9) NIC driver csomagküldés függvénye
  - (10) csomag leíró betöltése a ring bufferbe
    - sk_buff pointer
  - (11) csomag küldésre készen
- ha sikeres küldés
  - interrupt a CPU-nak
  - sk_buff felszabadítható

forrás: https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md

Mindez felülről



forrás: http://myaut.github.io/dtrace-stap-book/kernel/net.html

NAT

iptables

Network Address Translation

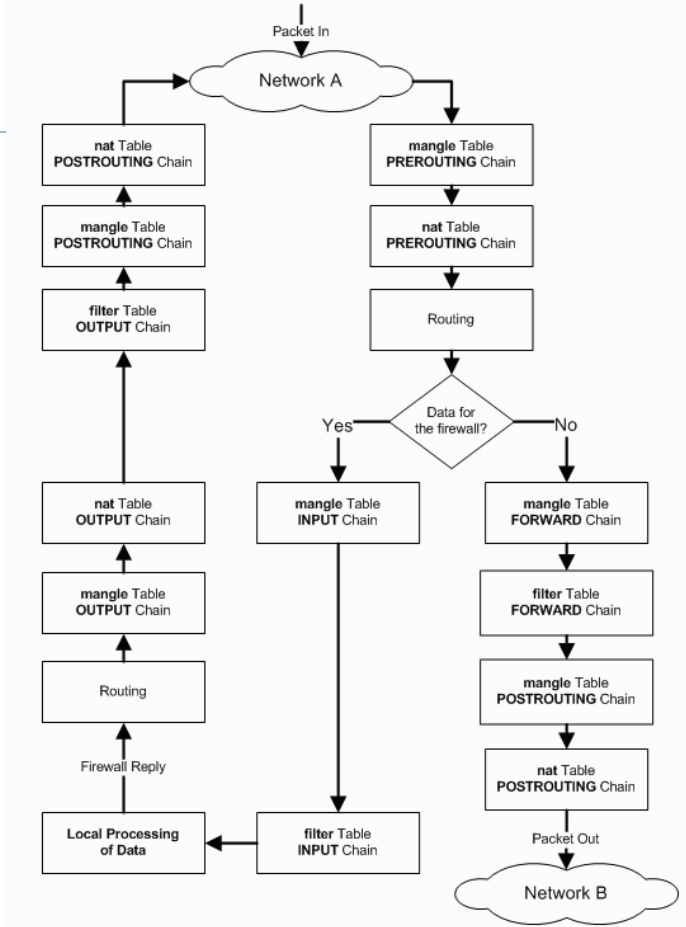
- NAT
  - olyan router, ami megváltoztatja a forrás vagy/és cél IP címet egy csomagban
  - leggyakrabban privát IP alhálózatot kapcsol a publikus internethez
- PAT (Port Address Translation)
  - a forrás vagy/és cél TCP/UDP port számot módosítja
  - általában beleértjük a NAT-ba
- SNAT (Source NAT)
  - forrás címet cserél a (kimenő) csomagokon egy fix címre
- Masquerading
  - forrás címet cserél a kimenő csomagokon dinamikus címre
- DNAT (Destination NAT)
  - cél címet cserél
- Port forwarding
  - DNAT, amikor külső hálózatról engedünk forgalmat a privát alhálózatba
  - külvilág számára látható ip:port számot kell a belső tartományra fordítani

iptables

- először "NAT"-olásra fogjuk használni
- általános célú csomagszűrő
- Linux alatt a csomagszűrés a kernel része (1.1 verziótól)
  - vagy teljesen bele van fordítva
  - vagy modulként tölthető be
- Rövid története
  - első változat
    - BSD UNIX ipfw programjára épült (Alan Cox portolta 1994 végén)
  - Linux 2.0
    - ipfwadm parancs kontrollálta a csomagszűrést
  - Linux 2.2
    - nagymértékben újraírták a kódot
    - új kezelőoldali alkalmazás: ipchains
  - Linux 2.4
    - iptables és a hozzá kapcsolódó kernelrészek újraírása

iptables

- Alapelemek: szabályok, láncok, táblák
  - láncokban tárolt szabályokon "megy" végig a csomag
  - ha illeszkedés van, target végrehajtása
- Három beépített tábla
  - filter: csomagok szűrése, szortírozása
  - **nat: címfordítási feladatok**
  - mangle: általános csomagmódosítások
- Hivatkozás
  - -t nat
  - -t filter (ez a default)
  - -t mangle
- Például
  - nat tábla listázása:
  - sudo iptables -t nat -nvL



SNAT konfigurálása (GW)

- Első lépés: forwarding engedélyezése
  - alapból nem tudjuk routerként használni a gépünket
    - `cat /proc/sys/net/ipv4/ip_forward`
  - engedélyezés
    - `sudo sysctl -w net.ipv4.ip_forward=1`
- címfordítás beállítása
  - `iptables -t nat`
  - `-A POSTROUTING  (append, új szabály hozzáfűzése a POSTROUTING lánchoz)`
  - `-s 10.0.0.0/24  (ha ez a source IP)`
  - `-o eth3  (ha ez az output interfész)`
  - `-j SNAT  (akkor source IP fordítása)`
  - `--to-source 192.168.3.227  (erre a címre)`

SNAT konfigurálása (CLIENT)

- default gateway beállítása
  - `sudo route add default gw 10.0.0.1 dev eth2`
- tesztelés
  - ping 8.8.8.8
  - ping index.hu ???

DNAT konfigurálása (GW)

- Adott portokon tegyük elérhetővé kívülről a belső gép
  - web szerverét (8080)
  - ssh szerverét (2222)
- címfordítás beállítása
  - `iptables -t nat`
  - `-A PREROUTING` (append, új szabály hozzáfűzése PREROUTING-hoz)
  - `-d 192.168.3.227` (ha ez a destination IP)
  - `-p tcp` (ha TCP protokoll)
  - `--dport 8080` (és 8080-as TCP destination port)
  - `-j DNAT` (akkor destination IP:port fordítása)
  - `--to-destination 10.0.0.2:80` (a belső web szerverre)
- hasonlóan
  - `iptables -t nat -A PREROUTING -d 192.168.3.227 -p tcp --dport 2222 -j DNAT --to-destination 10.0.0.2:22`

DNAT tesztelése

- Hoszt gépről
  - web browser
    - http://192.168.3.227
    - http://192.168.3.227:8080
  - ssh
    - ssh mininet@192.168.3.227
    - ssh mininet@192.168.3.227 -p 2222

Firewall

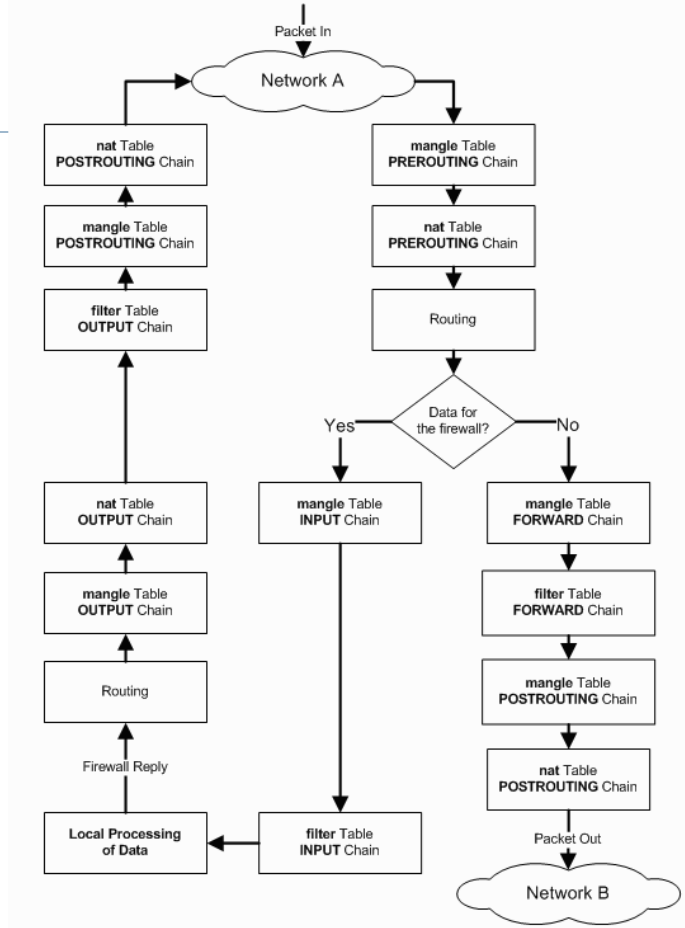
iptables

Firewall, tűzfal

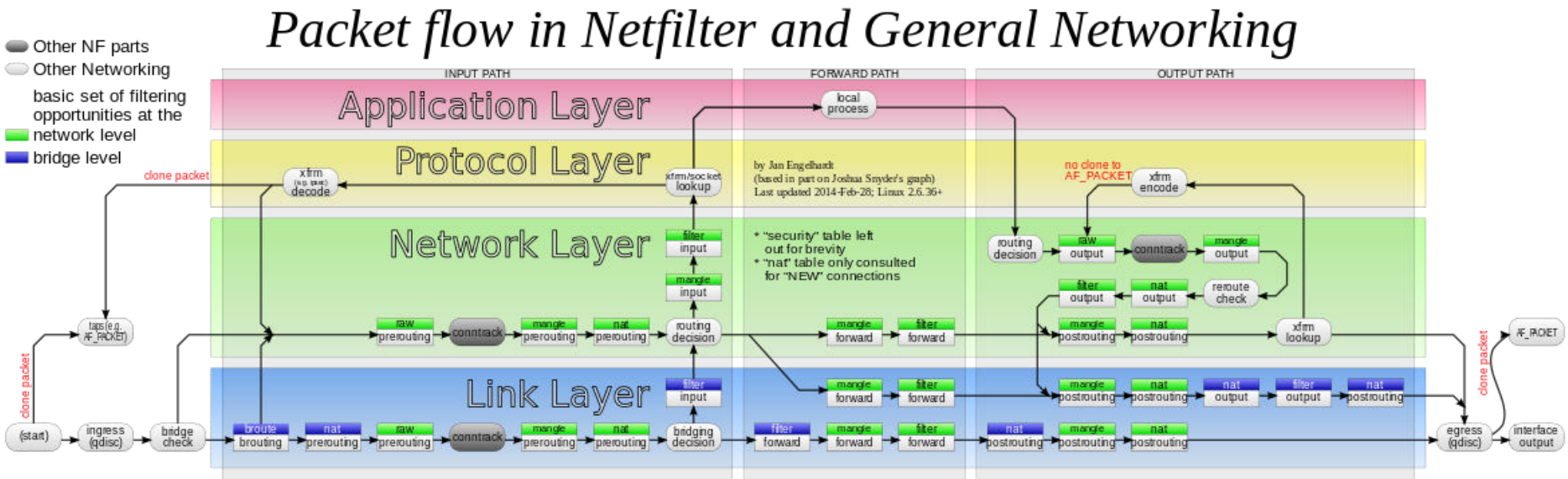
- Tűzfal: alapvető fontosságú eleme a hálózatnak
- Routing funkció mellett
  - döntést hoz, hogy adott forgalom mehet-e egyik hálózatból a másikba
  - sok (egyre több) tényezőt vehet figyelembe
  - csomag tartalmát is változtathatja
- Fajtái
  - stateless packet filters
    - egyenként vizsgálja a csomagokat (fejrészt)
  - stateful packet filters
    - kapcsolatokat követ, csomag fejrészt vizsgál
  - application level firewall
    - payloadot is néz
- Kapcsolódó hálózati funkciók
  - DPI (deep packet inspection), IPS (intrusion prevention system), IDS (intrusion detection system)
  - pl: DPI jelzésére új bejegyzés felvétele a tűzfalba

iptables

- iptables: user space alkalmazás, amivel a Linux kernel firewall (Netfilter modulok) konfigurálható
- Beérkező, kimenő és átmenő csomagok vizsgálata és szűrése
- rugalmasan konfigurálható, széles körben alkalmazzák
- Alapelemek: szabályok, láncok, táblák
  - láncokban tárolt szabályokon "megy" végig a csomag, ha illeszkedés van, target végrehajtása
- target:
  - elfogadás (ACCEPT), kilépünk a láncból
  - eldobás (DROP), nincs visszajelzés
  - visszautasítás (REJECT), van visszajelzés (port unreachable)
  - egy másik lánc, másik láncon folytatjuk
- Három beépített tábla
  - **filter: csomagok szűrése, szortírozása**
  - nat: címfordítási feladatok
  - mangle: általános csomagmódosítások
- Három előre definiált lánc a filter táblában
  - INPUT, OUTPUT, FORWARD



(Kicsit részletesebben…)



Műveletek láncokkal

- Előre definiált láncokat nem lehet törölni, alaphelyzetben üresek
- Tetszőleges láncok létrehozhatók, meglévő láncokhoz kell kapcsolni szabályok segítségével
- Lánckezelő parancsok:
  - -N Új lánc létrehozása
  - -X Üres lánc törlése
  - -P Default policy megváltoztatása beépített láncon
  - -L Adott lánc szabályainak listázása
  - -F Adott lánc összes szabályának törlése
  - -Z A csomag és byte számlálók nullázása egy adott lánc valamennyi szabályában.
- Pl: filter tábla teljes tartalmának lekérdezése
  - sudo iptables -nvL

Műveletek szabályokkal

- Szabályok létrehozása és törlése:
  - -A Új szabály hozzáfűzése a lánchoz
  - -I Szabály beszúrása az adott pozícióra
  - -R Az adott pozíciójú szabály cseréje új szabályra
  - -D Az adott pozíción lévő, vagy az első illeszkedő szabály törlése

Szűrési feltételek megadása

- Inverzió: "!"
- Forrás és célcím
  - -s, --source
    - a forrás IP címének meghatározása
  - -d, --destination
    - a cél IP címének meghatározása
  - például:
    - `iptables -A INPUT -s 10.0.0.0/8 -j DROP`
- Protokoll megadása
  - -p, --protocol
  - például:
    - `iptables -A INPUT -p icmp -j ACCEPT`

Szűrési feltételek megadása

- Interfész meghatározása
  - -i, --in-interface
    - a bejövő interfész definiálása
    - (OUTPUT láncon kimenő csomagoknak nincs bemenő interfésze, itt egy csomag sem illeszkedik)
    - például:
      - `iptables -A INPUT -i eth2 -j DROP`
  - -o, --out-interface
    - a kimenő interfész definiálása
    - (INPUT láncon bejövő csomagoknak nincs kimenő interfésze, itt egy csomag sem illeszkedik)

TCP kiterjesztés

- -p tcp hatására töltődik be, elérhető opciók:
- --source-port, --sport
  - forrás portra vagy port-tartományra illeszkedik
  - Például:
    - --sport 23 csak a 23-as portra illeszkedik
    - --sport 2000:3000 a 2000 és a 3000 közötti portokra illeszkedik (zárt intervallum)
    - --sport 2000: a 1999-nél nagyobb portokra illeszkedik
    - --sport :3000 a 3001-nél kisebb portokra illeszkedik
- --destination-port, --dport
  - célportra vagy port-tartományra illeszkedik
- --tcp-option
  - egy TCP opciót határoz meg, melyet a számával kell definiálnunk

TCP kiterjesztés

- --tcp-flags
  - TCP kapcsolók (flag-ek) vizsgálatát teszi lehetővé
  - két kötelező paramétere van
    - első: egy maszk, mely kapcsolókat vizsgáljuk (SYN,ACK, FIN, RST, URG, PSH,ALL)
    - második: mely kapcsolóknak kell aktívnak lenniük (NONE is érvényes)
- --syn
  - ugyanaz, mint
    - --tcp-flags SYN,RST,ACK SYN
- például: bejövő kapcsolatok tiltása:
  - `iptables -A INPUT -p tcp --syn -j DROP`

state modul

- A kapcsolat állapota alapján végezhetünk szűréseket
  - az ip_conntrack modul a kapcsolatkövető és analizáló részét implementálja
  - stateful packet inspection
- -m state paranccsal aktiválhatjuk
- --state
  - a kapcsolat állapotát vizsgálhatjuk
  - paraméterként az állapotok vesszővel elválasztott listája
- Lehetséges állapotok:
  - NEW — Új kapcsolatot létesítő csomag
  - ESTABLISHED — Egy már felépített, létező kapcsolathoz tartozó csomag
  - RELATED — Egy kapcsolathoz tartozó, de annak részét nem képező csomag, például ICMP hibaüzenet
  - INVALID — Azonosítatlan csomag, mely nem rendelhető egyetlen kapcsolathoz sem
- Például:
  - `iptables -A INPUT -m state --state NEW,INVALID -j DROP`

Más hasznos modulok

- Limit modul
  - -m limit, --match limit hatására töltődik be
    - korlátozhatjuk az illeszkedések számát, naplózás csökkentésére, vagy DoS támadások ellen
      - DoS támadás: nagyszámú csomag árasztja el a számítógépet, így az képtelen lesz válaszolni a bejövő kérésekre
  - --limit
    - adott időintervallumon belüli maximális illeszkedések száma (pl: 2/second)
  - --limit-burst
    - maximális csomagszám mielőtt a szabályt nem illeszkedőnek vennénk
    - korlátozás tocken bucket segítségével
      - ha van a vödörben token, akkor a bejövő csomagot elfogadjuk
      - ha nincs, akkor nem illeszkedőnek minősítjük
      - elfogadásnál a vödörből kiveszünk egy tokent
      - periodikus újratöltés, a vödör mérete maximalizálva van
      - --limit : milyen gyorsan töltjük újra a vödröt tokenekkel
      - --limit-burst: a vödör mérete

Más hasznos modulok

- Limit modul
  - példák:
    - Syn-flood elleni védelem:
      - `iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT`
    - Portscan elleni védelem:
      - `iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT`
    - Ping-flood elleni védelem:
      - `iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT`

Más hasznos modulok

- LOG modul
  - -j LOG illeszkedő csomagok kernel szintű naplózása (syslog)
  - például:
    - `iptables -A INPUT -p TCP --log-prefix Tcp_ -j LOG`
    - `iptables -A FORWARD -m limit --limit-burst 5 --limit 2/s -j LOG --log-prefix 'FIREWALL: ' --log-level 7`
- UDP kiterjesztés
  - -p udp hatására töltődik be, elérhető opciók:
  - --source-port, --sport
    - mint TCP-nél
  - --destination-port, --dport
    - mint TCP-nél

Firewall konfigurálása (GW)

- Jó tanácsok:
  - default policy legyen DROP (vagy REJECT)
  - a kívánt forgalmakat külön-külön, explicit módon engedélyezzük (ACCEPT)
  - amelyik csomag végigmegy minden láncon és nincs illeszkedés, eldobásra kerül
  - a végére betehetünk egy loggoló szabályt, ami a drop helyett egy log bejegyzést készít

Firewall konfigurálása (GW)

(mininet) 192.168.56.102 — Konsole

File Edit View Bookmarks Settings Help

File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash

# delete chains
iptables -F INPUT
iptables -F tcpfilter
iptables -X  # delete all user-specified chains
iptables -Z  # reset counters
# set default policies
iptables -P INPUT DROP
# create tcpfilter chain
iptables -N tcpfilter

# allow mgmt traffic
iptables -A INPUT -s 192.168.56.0/24 -p tcp -j ACCEPT
# allow icmp traffic
iptables -A INPUT -p icmp -j ACCEPT
# allow internal tcp traffic
iptables -A tcpfilter -s 10.0.0.0/24 -p tcp -j ACCEPT
# allow external tcp traffic if it relates to a connection
iptables -A tcpfilter ! -s 10.0.0.0/24 -p tcp -m state \
    --state ESTABLISHED,RELATED -j ACCEPT
# deny external tcp connection request (eth3 is the external interface)
iptables -A tcpfilter -i eth3 -p tcp -m state --state NEW -j REJECT
# connect tcpfilter to INPUT chain
iptables -A INPUT -j tcpfilter
```

Bejövő forgalom szűrése (INPUT lánc)

Tesztelés

- ping, web, ssh
  - GW->internet
  - CLIENT->GW
  - internet->GW

Tesztelés

- ping, web, ssh
  - GW->internet
    - enable DNS
    - **`iptables -A INPUT -p udp --sport 53 -j ACCEPT`**
  - CLIENT->GW
  - internet->GW

Firewall konfigurálása (GW)

(mininet) 192.168.56.102 — Konsole

File Edit View Bookmarks Settings Help

File Edit Options Buffers Tools Sh-Script Help

```
#!/bin/bash

# delete chains
iptables -F FORWARD
iptables -X  # delete all user-specified chains
iptables -Z  # reset counters
# set default policies
iptables -P FORWARD DROP

# allow icmp traffic
iptables -A FORWARD -p icmp -j ACCEPT
# enable outgoing traffic
iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
# enable backward direction if it was initiated from the internal domain
iptables -A FORWARD -d 10.0.0.0/24 -p tcp \
    -m state --state ESTABLISHED,RELATED -j ACCEPT
# enable DNAT ports from the external net
iptables -A FORWARD ! -s 10.0.0.0/24 -p tcp --dport 80 \
    -m state --state NEW -j ACCEPT
iptables -A FORWARD ! -s 10.0.0.0/24 -p tcp --dport 22 \
    -m state --state NEW -j ACCEPT
# enable DNS
iptables -A FORWARD -p udp --sport 53 -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -j ACCEPT
# log dropped packets
iptables -A FORWARD -m limit --limit-burst 5 --limit 2/s \
    -j LOG --log-prefix 'FIREWALL: ' --log-level 7
```

Átmenő forgalom szűrése (FORWARD lánc)

Tesztelés

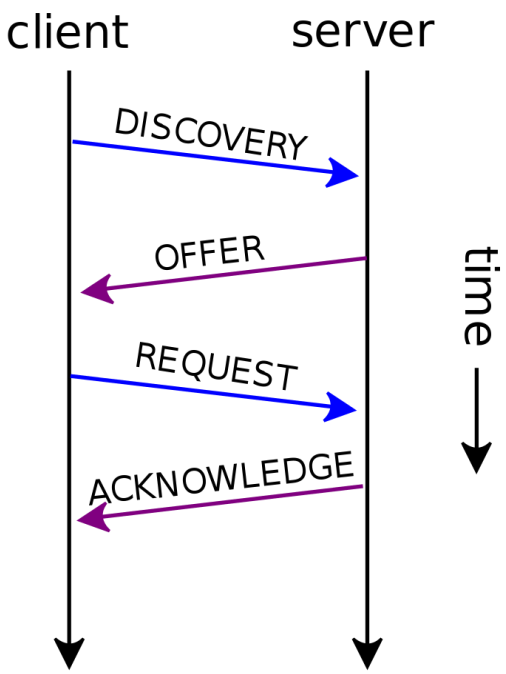
- ping, web, ssh, közben logok figyelése
  - CLIENT->internet
  - internet -> CLIENT

DHCP

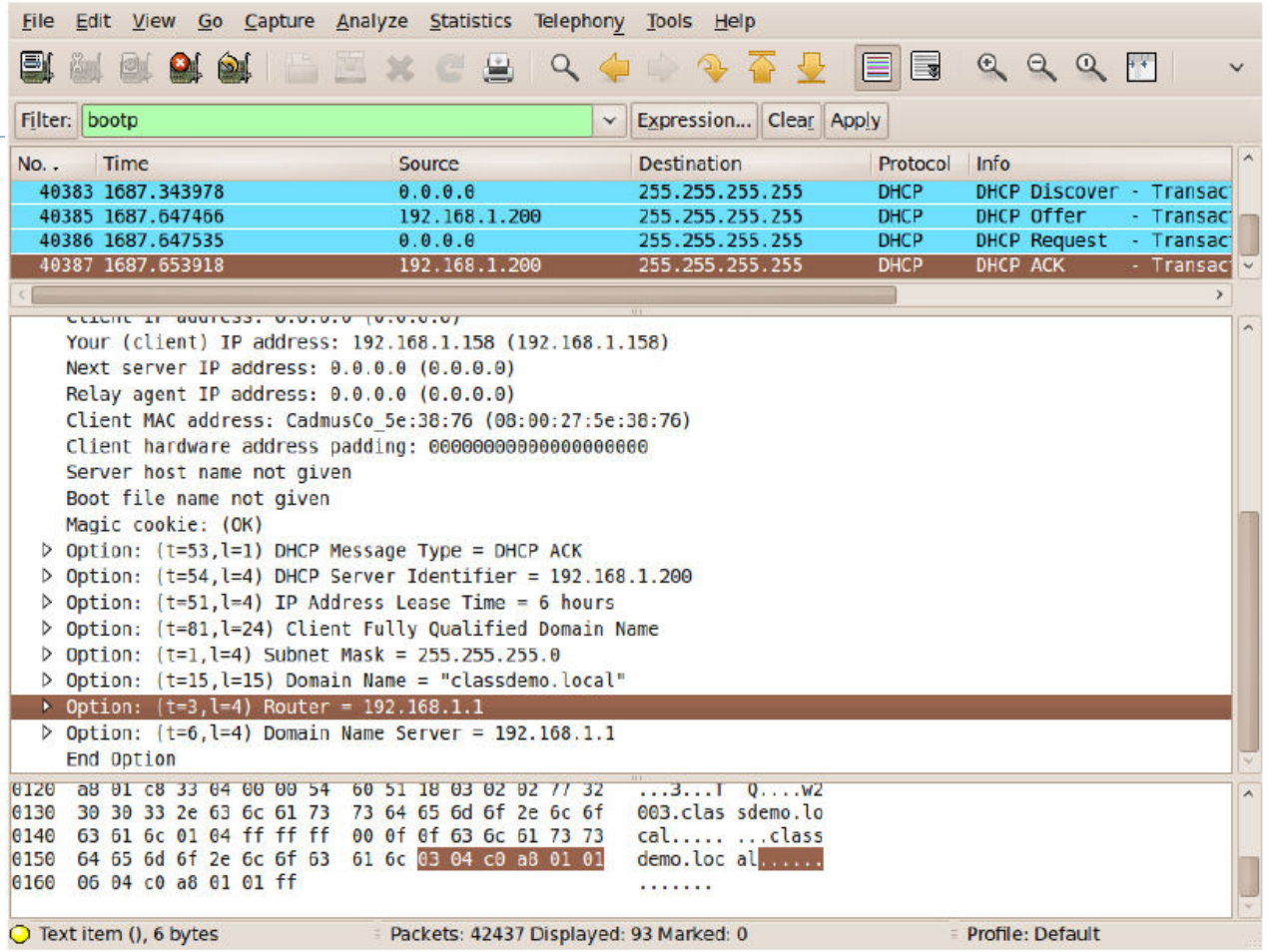
DHCP

- Dynamic Host Configuration Protocol
  - szabványos (RFC) protokoll hálózati interfészek dinamikus konfigurálására
  - UDP felett megy
- Rövid történelem
  - 1984: RARP (Reverse Address Resolution Protocol, RFC 903)
    - diszk nélküli munkaállomások IP konfigurálása
    - layer 2-es működés
    - egy hálózaton kellett lenni a szerverrel
  - 1985: BOOTP (Bootstrap Protocol, RFC 951)
    - hálózati bootolás (netboot)
    - relay agent: IP alhálózatok között is átmennek a BOOTP csomagok
    - egy szerver több alhálót is kiszolgál
  - 1993: DHCP (RFC 1531, RFC 1541, RFC 2131)
    - BOOTP-n alapul
    - dinamikus IP cím kiosztás és felszabadítás
    - számos egyéb konfigurációs paraméter (pl. nameserver)

DHCP



Több szerver is lehet!



Installáljunk dhcp szervert!

- sudo apt-get install isc-dhcp-server
  - (Ubuntu-t vagy Debiant feltételezünk)
  - Internet Software Consortium implementációja
  - korábban dhcp3-server
  - alapból nem indul
  - sudo service isc-dhcp-server status:
    - isc-dhcp-server stop/waiting

Konfiguráljunk dhcp szervert!

- 1. lépés
  - beállítjuk az interfészeket, ahol DHCP kéréseket kezelünk
  - /etc/default/isc-dhcp-server fájlban:
    - INTERFACES="eth2“
- 2. lépés
  - konfiguráljuk a szervert
  - /etc/dhcpd/dhcpd.conf

```
# HeEpUz internal subnet.
subnet 10.0.0.0 netmask 255.255.255.0 {
 range 10.0.0.101 10.0.0.150;
 option domain-name-servers 152.66.115.1, 8.8.8.8;
 option domain-name "haepuz.hu";
 option routers 10.0.0.1;
 option broadcast-address 10.0.0.255;
 default-lease-time 600;
 max-lease-time 7200;
}
```

- 3. lépés
  - isc-dhcp-server service indítása:
    - sudo service isc-dhcp-server start
    - sudo service isc-dhcp-server status

Teszteljük a kliens gépről!

- CLIENT
  - manuálisan konfigurált cím törlése
    - sudo ip addr del 10.0.0.2/24 dev eth2
  - cím kérése dhcp-vel (közben wireshark capture):

```
mininet@CLIENT:~$ sudo dhclient -v eth2
Internet Systems Consortium DHCP Client 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth2/08:00:27:53:0b:a9
Sending on   LPF/eth2/08:00:27:53:0b:a9
Sending on   Socket/fallback
DHCPDISCOVER on eth2 to 255.255.255.255 port 67 interval 3 (xid=0xb97b96a)
DHCPREQUEST of 10.0.0.101 on eth2 to 255.255.255.255 port 67 (xid=0xb97b96a)
DHCPOFFER of 10.0.0.101 from 10.0.0.1
DHCPACK of 10.0.0.101 from 10.0.0.1
bound to 10.0.0.101 -- renewal in 251 seconds.
```

Teszteljük a kliens gépről!

- CLIENT
  - névfeloldás (resolv.conf fájl) ellenőrzése:

```
mininet@CLIENT:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 152.66.115.1
nameserver 8.8.8.8
search haepuz.hu
```

Teszteljük a kliens gépről!

- CLIENT
  - cím felszabadítása (közben wireshark capture):

```
mininet@CLIENT:~$ sudo dhclient -v -r eth2
Internet Systems Consortium DHCP Client 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth2/08:00:27:53:0b:a9
Sending on   LPF/eth2/08:00:27:53:0b:a9
Sending on   Socket/fallback
DHCPRELEASE on eth2 to 10.0.0.1 port 67 (xid=0x6e762b08)
```

Kliens gép konfigurálása

- Ha nem akarjuk kézzel kérni a címet
  - eth2 interfész konfigurálása az /etc/network/interfaces fájlban:
    - auto eth2
    - iface eth2 inet dhcp
  - ezután használhatók a következő parancsok
    - sudo ifup eth2
    - sudo ifdown eth2
  - “auto” esetén indulásnál felkonfigurálódik

dhcpd.conf: további lehetőségek

```
subnet 192.168.213.0 netmask 255.255.255.0 {
  range 192.168.213.160 192.168.213.199;
  filename "/grldr";
  next-server 192.168.213.1;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.213.255;
  option routers 192.168.213.1;
}
#IB.213
#A sor
host 0 {hardware ethernet 1c:6f:65:3d:e5:44; fixed-address 192.168.213.100; option host-name lab0;}
host 1 {hardware ethernet d8:50:e6:41:f7:bd; fixed-address 192.168.213.101; option host-name lab1;}
host 2 {hardware ethernet d8:50:e6:41:f7:0b; fixed-address 192.168.213.102; option host-name lab2;}
host 3 {hardware ethernet d8:50:e6:3c:46:9d; fixed-address 192.168.213.103; option host-name lab3;}
host 4 {hardware ethernet d8:50:e6:41:f7:6c; fixed-address 192.168.213.104; option host-name lab4;}
host 5 {hardware ethernet d8:50:e6:41:f5:29; fixed-address 192.168.213.105; option host-name lab5;}
host 6 {hardware ethernet d8:50:e6:3c:48:ab; fixed-address 192.168.213.106; option host-name lab6;}
host 7 {hardware ethernet d8:50:e6:41:f8:20; fixed-address 192.168.213.107; option host-name lab7;}
host 8 {hardware ethernet d8:50:e6:3c:49:7f; fixed-address 192.168.213.108; option host-name lab8;}
```

DNS

bind9

DNS

- Domain Name System
  - fundamentális része a nagyobb számítógép-hálózatoknak, internetnek
  - hierarchikus, decentralizált/elosztott "naming system" (azonosító rendszer)
  - internetre kapcsolt erőforrások (számítógép, szolgáltatások) azonosítására, lokalizálására
  - "worldwide, distributed directory service"
  - és a protokoll, amin keresztül lekérdezhető (UDP 53-as port)
  - leggyakrabban: domain name -> IP cím fordítás
  - korábban már láttuk, hogy nélküle nehéz az élet…
  - (most megnézzük, hogy vele is…)

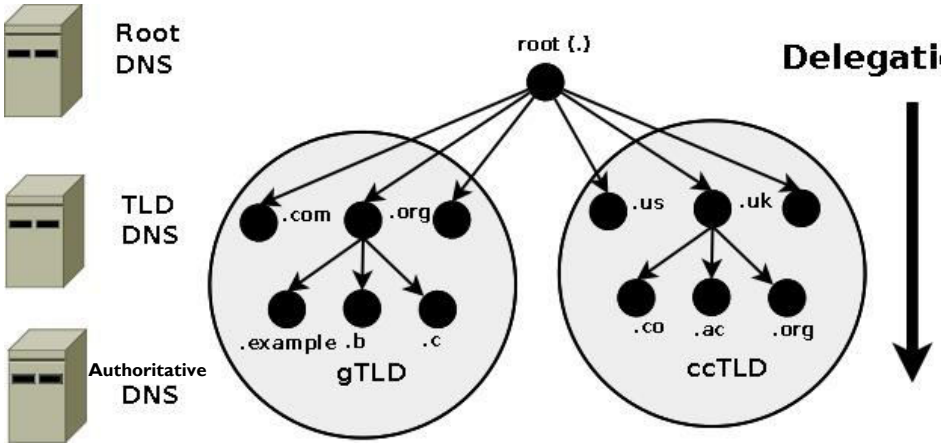
DNS

- Domain Name System
  - név és erőforrás összerendelése
    - nem kell foglalkoznunk vele, ha változik egy gép helye, IP címe
    - pl. terheléselosztásnál dinamikusan változhat a feloldás
  - szerver oldali elemek: névszerverek (name server)
    - authoritative name server
      - egy teljes domain felelőse
    - hierarckikus rendszerben tovább delegálható
      - sub-domain delegált kiszolgálóval
  - kliens oldali elem: névfeloldó (resolver)
    - névfeloldás különböző információk alapján
      - pl: /etc/hosts fájl: statikus összerendelések
      - pl: resolvconf csomag dinamikusan frissíti az /etc/resolv.conf fájlt
  - bonyolult rendszer, itt csak az alap dolgokat nézzük meg

DNS rövid történet

- ARPANET idején
  - hosts.txt fájl
    - Stanford Research Institute tartotta karban
    - statikus név->IP cím összerendelések
    - operátorok másolatot kaptak
- 1983: Domain Name System
  - Paul Mockapetris (University of California, Irvine) és Jon Postel (ISI, University of Southern California)
  - RFC 882, 883
  - ez az alapkoncepció ma is
- 1984: első UNIX implementáció
  - UC Berkeley hallgatók
    - Douglas Terry, Mark Painter, David Riggle, Songnian Zhou
  - Berkeley Internet Name Domain (BIND)
- 1985: revised BIND implementáció
  - Kevin Dunlap, DEC
- 1990-es évek: portolás Windows NT-re
- BIND: a legelterjedtebb DNS szoftver
  - mi a bind9 csomagot fogjuk használni

DNS szerverek, domain name delegáció



http://www.zytrax.com/books/dns/ch2/

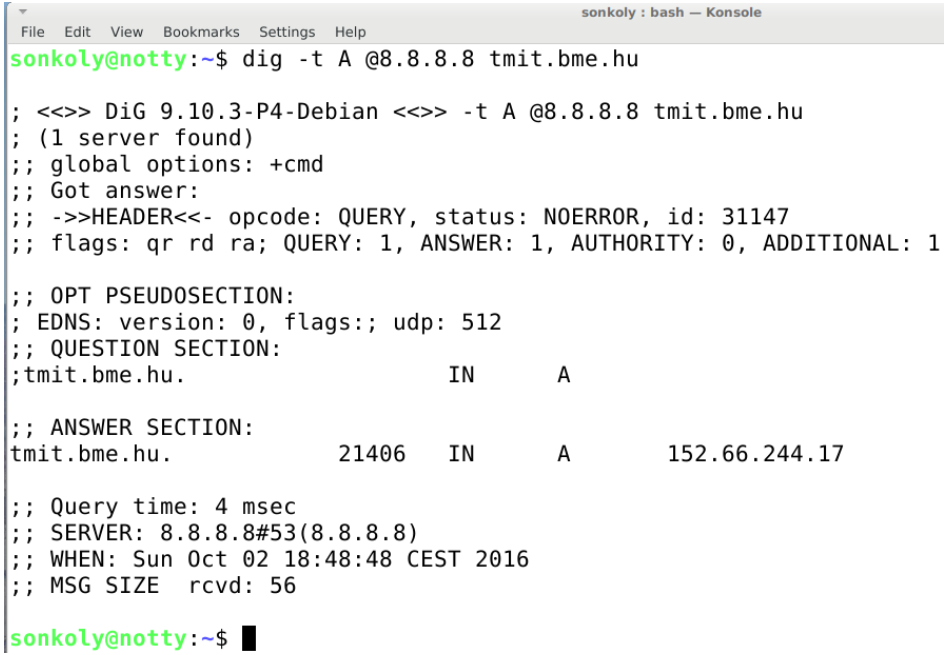
- hierarchikus fa struktúra
- 13 db root szerver: "." (a-m)
  - 12 különböző szervezet üzemelteti
  - több példány (megbízhatóság)
  - ugyanaz az IP címük
    - IP anycast
    - legközelebbi válaszol
- Top Level Domain (TLD)
  - Generic TLD (gTLD)
    - Sponsored TLD (sTLD)
  - Country Code TLD (ccTLD)
- Kulcs: cache-elés

Névfeloldás

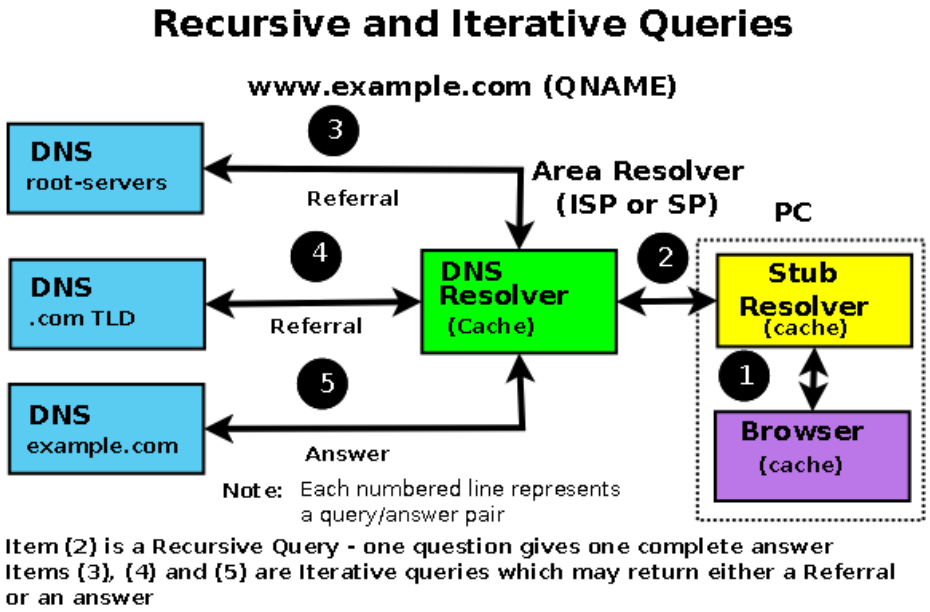
- Komponensek
  - Resource Records (Zone Files)
    - name, [TTL], [class], type, rdata
    - name
      - domain name
    - TTL
      - Time To Live
    - class
      - IN (Internet)
    - type
      - A (IPv4), AAAA (IPv6), NS (name server), glue (NS<->A), CNAME (alias), MX (mail server), SOA (Start of Authority) …
    - rdata
      - típustól függő adat
    - például A record:
      - `tmit.bme.hu.  1200  IN  A  152.66.244.17`
  - Name Server programok
  - Resolver program, library (kliens oldalon)

Névfeloldás

- DNS Query fajtái
  - recursive query (nem kötelező támogatni)
    - mindig teljes válasz
  - iterative / non-recursive query (kötelező támogatni)
    - teljes válasz (ha elérhető)
    - “referral” egy másik szerverre
  - inverse query
    - IP -> név
- Példa



DNS Query



- (1) pl. browser névfeloldást kér (call standard lib function, pl. `gethostbyname()`)
- (2) lokális resolver recursive query-t küld a közvetlen DNS szerverének
- (3) ott nincs cache-elve, ezért (iterative) query a root DNS szervernek
- root nem tudja feloldani, de ismeri a .com-hoz tartozó TLD szervert, referral-t küld
- (4) újabb (iterative) query, most a TLD-nek
- válasz: referral
- (5) (iterative) query eample.com szerverének
- válasz: "A" (IPv4) record
- (2) válasz küldése a kliensnek
- + információ cache-elése
- (1) válasz küldése a programnak (browser)
- + információ cache-elése

http://www.zytrax.com/books/dns/ch2/

DNS Query példák

sonkoly : bash — Konsole

File Edit View Bookmarks Settings Help

```
sonkoly@notty:~$ dig tmit.bme.hu +norecurse +short
sonkoly@notty:~$
sonkoly@notty:~$ dig tmit.bme.hu @152.66.246.10 +norecurse

; <<>> DiG 9.10.3-P4-Debian <<>> tmit.bme.hu @152.66.246.10 +norecurse
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47941
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;tmit.bme.hu.                   IN      A

;; ANSWER SECTION:
tmit.bme.hu.            86400   IN      A       152.66.244.17

;; AUTHORITY SECTION:
tmit.bme.hu.            86400   IN      NS      anubis.tmit.bme.hu.
tmit.bme.hu.            86400   IN      NS      nic.bme.hu.
tmit.bme.hu.            86400   IN      NS      ehnaton.tmit.bme.hu.
tmit.bme.hu.            86400   IN      NS      corto.tmit.bme.hu.
tmit.bme.hu.            86400   IN      NS      ns2.tmit.bme.hu.

;; ADDITIONAL SECTION:
ns2.tmit.bme.hu.        86400   IN      A       152.66.246.170
corto.tmit.bme.hu.      86400   IN      A       152.66.246.10
anubis.tmit.bme.hu.     86400   IN      A       152.66.245.198
ehnaton.tmit.bme.hu.    86400   IN      A       152.66.245.197

;; Query time: 6 msec
;; SERVER: 152.66.246.10#53(152.66.246.10)
;; WHEN: Sun Oct 02 18:57:08 CEST 2016
;; MSG SIZE  rcvd: 219
```

sonkoly : bash — Konsole

File Edit View Bookmarks Settings Help

```
sonkoly@notty:~$ dig tmit.bme.hu @8.8.8.8 +trace

; <<>> DiG 9.10.3-P4-Debian <<>> tmit.bme.hu @8.8.8.8 +trace
;; global options: +cmd
.                       16834   IN      NS      h.root-servers.net.
.                       16834   IN      NS      e.root-servers.net.
.                       16834   IN      NS      b.root-servers.net.
.                       16834   IN      NS      c.root-servers.net.
.                       16834   IN      NS      d.root-servers.net.
.                       16834   IN      NS      f.root-servers.net.
.                       16834   IN      NS      g.root-servers.net.
.                       16834   IN      NS      k.root-servers.net.
.                       16834   IN      NS      m.root-servers.net.
.                       16834   IN      NS      l.root-servers.net.
.                       16834   IN      NS      i.root-servers.net.
.                       16834   IN      NS      j.root-servers.net.
.                       16834   IN      NS      a.root-servers.net.
.                       16834   IN      RRSIG   NS 8 0 518400 20161015050000 20161002040000 39291 . SZtC3M
j9ETTblMfF4g/q5eWNF9DTm+e3PtCaAAGB5sDvwGlR5q+JtSWS Hh1nM/4npWO5Izj4gimi05Qx7vHBzERiyPRfu+6ZIbrBV2HR8RaRulY
W 6QMgissUuN3viSMMljNU1TofCMVztWk1YoI+hweyXJvf0K/j7lr7iPX1 41Nb6KlsRV7tf6C8mIwcevBbvD7N8Utfz0toHefLneoTXS5
ttrzaUcmJ CewDVZb1GgP9OgH92qmpq2MnBzBmuRZiF3SyKats6dT/NHhbkXDTR0l5 DO/ggIN/xkTAWcYF5GcUaP38YdnizM5W8uTAUNe
p1UqMDOwboShCqJnz LIZJ6w==
;; Received 525 bytes from 8.8.8.8#53(8.8.8.8) in 6 ms

hu.                     172800  IN      NS      e.hu.
hu.                     172800  IN      NS      d.hu.
hu.                     172800  IN      NS      c.hu.
hu.                     172800  IN      NS      b.hu.
hu.                     172800  IN      NS      a.hu.
hu.                     172800  IN      NS      ns-com.nic.hu.
hu.                     172800  IN      NS      ns2.nic.fr.
hu.                     86400   IN      DS      20056 8 2 93FDCE134B52B1BBFDCADDAD9152B0F2CA6E9DEE09150DF6
24661C6B 9A5B74DA
hu.                     86400   IN      RRSIG   DS 8 1 86400 20161015050000 20161002040000 39291 . ZnbOLc4
8OI8nWZ6vyV/T06cim4/d/QuX+c2osRJNc98Ic6+KEt9oftMU iyexts+dY8qUN3pHP5EmR1vlmIPkvdbATiIH0LX5b1XghCR0TWEfh4z+
 /jcRmWJANjWfs8Lu+VX6xVqUdkloCkFNTqtNmC7I1PzVDclc0R/RmL6N dirL3M+wxqU/b0UMN8rVMXmc2jbUtTLtMP6p1Z0kb8MAsAD6
IKWuiV5b tmxFi0/vrHtmOnNMyEoHhGke+XopRAytTvJZuk7TT6X6XmSVrEOzR5bN E7gn4aIYy2xoiNGDCLIlcuT9veFbHrSQ0ah8PZw7
nQekF6pua8VuLhCR OwEcbw==
;; Received 756 bytes from 202.12.27.33#53(m.root-servers.net) in 294 ms

bme.hu.                 86400   IN      NS      ns.bme.hu.
bme.hu.                 86400   IN      NS      nic.bme.hu.
bme.hu.                 86400   IN      NS      ns2.pantel.net.
8ab2d3lrnel6sukrtkn88nvihc9glv88.hu. 3600 IN NSEC3 1 1 5 09A45AA33A023A65 8ABU5758I1BQ3JAKRV9ONOIV8P79FRR3
 NS SOA TXT RRSIG DNSKEY NSEC3PARAM
8ab2d3lrnel6sukrtkn88nvihc9glv88.hu. 3600 IN RRSIG NSEC3 8 2 3600 20161029101812 20161001163105 23295 hu.
h14EJIKAYTE48dkjhXtjvBEj83zsWQVPLrrhwV982zb6VUIp61XUgOM5 1dlF2Zleud67lntXHTeg8FWC3qJcCmAo61tTh7yVUrn1Cgewu
UN9pAyR YxEaBMz+4hnfBRkCoJ8IJ+87PeMTYLfXrkd9WcElrpEy/LoB27416vRy LNQ=
2r70vs0gsot7p8na739qe3o3ebeb54nm.hu. 3600 IN NSEC3 1 1 5 09A45AA33A023A65 2R80BPGANG52BCJDLBMCQU1KCJA21L4B
 NS DS RRSIG
2r70vs0gsot7p8na739qe3o3ebeb54nm.hu. 3600 IN RRSIG NSEC3 8 2 3600 20161028140039 20160930223100 23295 hu.
CvoDhb0/f0y+ob8QyuBFJ9byhm+l9qXFYUOHynTKloX4VZIG8G1Ej5gc tXoXPpTB4b99RHaXXsxEnxKc6GHkFTSs59lZfN5EPJF5cjneT
ZHZXKxN cgnR+jfw+PFkq6Z+6m53HBl5UIBsq07vn2Q/HqvUZPnzaG+oMghug476 YbU=
;; Received 706 bytes from 194.0.1.12#53(ns-com.nic.hu) in 3 ms

tmit.bme.hu.            14400   IN      NS      ehnaton.tmit.bme.hu.
tmit.bme.hu.            14400   IN      NS      nic.bme.hu.
tmit.bme.hu.            14400   IN      NS      corto.tmit.bme.hu.
tmit.bme.hu.            14400   IN      NS      ns2.tmit.bme.hu.
;; Received 210 bytes from 152.66.116.1#53(ns.bme.hu) in 3 ms

tmit.bme.hu.            86400   IN      A       152.66.244.17
;; Received 56 bytes from 152.66.245.197#53(ehnaton.tmit.bme.hu) in 4 ms
```

bind9

- sudo apt-get install bind9
- sudo netstat -aputne | grep -i listen
- konfigurációs fájlok
  - /etc/bind könyvtár alatt
    - named.conf.default-zones
      - zone entry-k megadása
      - zone db hivatkozás (pl. db.local)

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
```

(mininet) 192.168.56.102 — Konsole

File Edit View Bookmarks Settings Help

```
File Edit Options Buffers Tools Conf Help
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

bind9 (db.root)

```
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000      AAAA  2001:503:BA3E::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     192.228.79.201
;
; FORMERLY C.PSI.NET
;
.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;
; FORMERLY TERP.UMD.EDU
;
.                       3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000      A     199.7.91.13
D.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:2D::D
;
; FORMERLY NS.NASA.GOV
;
.                       3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000      A     192.203.230.10
;
; FORMERLY NS.ISC.ORG
;
.                       3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000      A     192.5.5.241
F.ROOT-SERVERS.NET.     3600000      AAAA  2001:500:2F::F
```

(named.conf.options)

GW konfigurálása

- dhcpd.conf frissítése

```
# HeEpUz internal subnet.
subnet 10.0.0.0 netmask 255.255.255.0 {
 range 10.0.0.101 10.0.0.150;
 option domain-name-servers 10.0.0.1;
 option domain-name "haepuz.hu";
 option routers 10.0.0.1;
 option broadcast-address 10.0.0.255;
 default-lease-time 600;
 max-lease-time 7200;
}
```

- tesztelés a kliens gépről
  - sudo dhclient –r –v eth2
  - sudo dhclient –v eth2
  - dig stanford.edu +trace

```
mininet@CLIENT:~$ dig stanford.edu @10.0.0.1 +norecurse +short
mininet@CLIENT:~$
mininet@CLIENT:~$
mininet@CLIENT:~$ dig stanford.edu @10.0.0.1 +short
171.67.215.200
mininet@CLIENT:~$
mininet@CLIENT:~$ dig stanford.edu @10.0.0.1 +norecurse +short
171.67.215.200
mininet@CLIENT:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
479	8.760786000	10.0.0.2	10.0.0.1	DNS	85	Standard query 0xa142 A stanford.edu
480	8.761045000	10.0.0.1	10.0.0.2	DNS	296	Standard query response 0xa142
593	14.263840000	10.0.0.2	10.0.0.1	DNS	85	Standard query 0xea0b A stanford.edu
594	14.265489000	192.168.1.227	8.8.8.8	DNS	85	Standard query 0x33fb A stanford.edu
595	14.294314000	8.8.8.8	192.168.1.227	DNS	273	Standard query response 0x33fb A 171.67.215.200 RRSIG
596	14.294839000	192.168.1.227	8.8.8.8	DNS	85	Standard query 0x6c22 DNSKEY stanford.edu
597	14.332023000	8.8.8.8	192.168.1.227	DNS	1277	Standard query response 0x6c22 DNSKEY DNSKEY DNSKEY DNSKEY RRSIG RRSIG
598	14.332850000	192.168.1.227	8.8.8.8	DNS	85	Standard query 0xc8d2 DS stanford.edu
599	14.359516000	8.8.8.8	192.168.1.227	DNS	296	Standard query response 0xc8d2 DS RRSIG
600	14.360162000	192.168.1.227	8.8.8.8	DNS	76	Standard query 0x9520 DNSKEY edu
601	14.389508000	8.8.8.8	192.168.1.227	DNS	787	Standard query response 0x9520 DNSKEY DNSKEY RRSIG
602	14.390065000	192.168.1.227	8.8.8.8	DNS	76	Standard query 0xe3bc DS edu
603	14.393985000	8.8.8.8	192.168.1.227	DNS	411	Standard query response 0xe3bc DS RRSIG
604	14.394333000	192.168.1.227	8.8.8.8	DNS	72	Standard query 0xb69a DNSKEY <Root>
605	14.405294000	8.8.8.8	192.168.1.227	DNS	1055	Standard query response 0xb69a DNSKEY DNSKEY DNSKEY RRSIG
606	14.406420000	10.0.0.1	10.0.0.2	DNS	101	Standard query response 0xea0b A 171.67.215.200
1816	25.660057000	10.0.0.2	10.0.0.1	DNS	85	Standard query 0xff9d A stanford.edu
1817	25.660272000	10.0.0.1	10.0.0.2	DNS	101	Standard query response 0xff9d A 171.67.215.200

```
Transaction ID: 0xff9d
▷ Flags: 0x80a0 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 1
▷ Queries
▽ Answers
  ▽ stanford.edu: type A, class IN, addr 171.67.215.200
      Name: stanford.edu
      Type: A (Host address)
      Class: IN (0x0001)
      Time to live: 21 minutes, 12 seconds
      Data length: 4
      Addr: 171.67.215.200 (171.67.215.200)
▷ Additional records
```

Összefoglalás

- Egyszerű teszthálózat kialakítása
  - gateway
  - kliens (belső hálózat)
- Hálózati funkciók vizsgálata, konigurálása
- NAT
  - SNAT, DNAT
  - iptables
- Firewall
  - iptables
- DHCP
  - isc-dhcp-server
- DNS
  - bind9