

Hálózatok építése és üzemeltetése

Hálózati funkciók a gyakorlatban

Áttekintés

- ▶ Példa hálózat beüzemelése
- ▶ Egy csomag útja a Linux rendszerben
- ▶ NAT
- ▶ Firewall
- ▶ DHCP
- ▶ DNS

Hol tartunk?

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

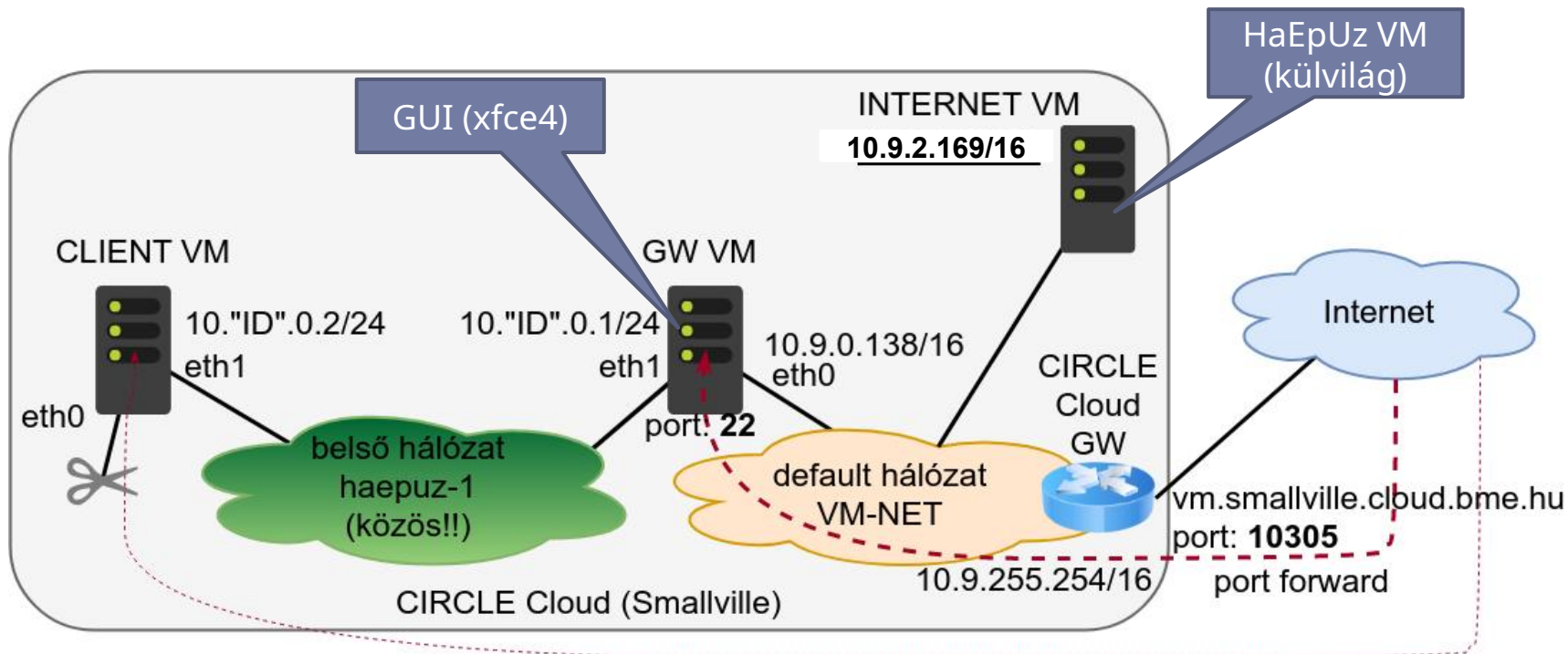
Mai téma

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

A példa hálózatunk

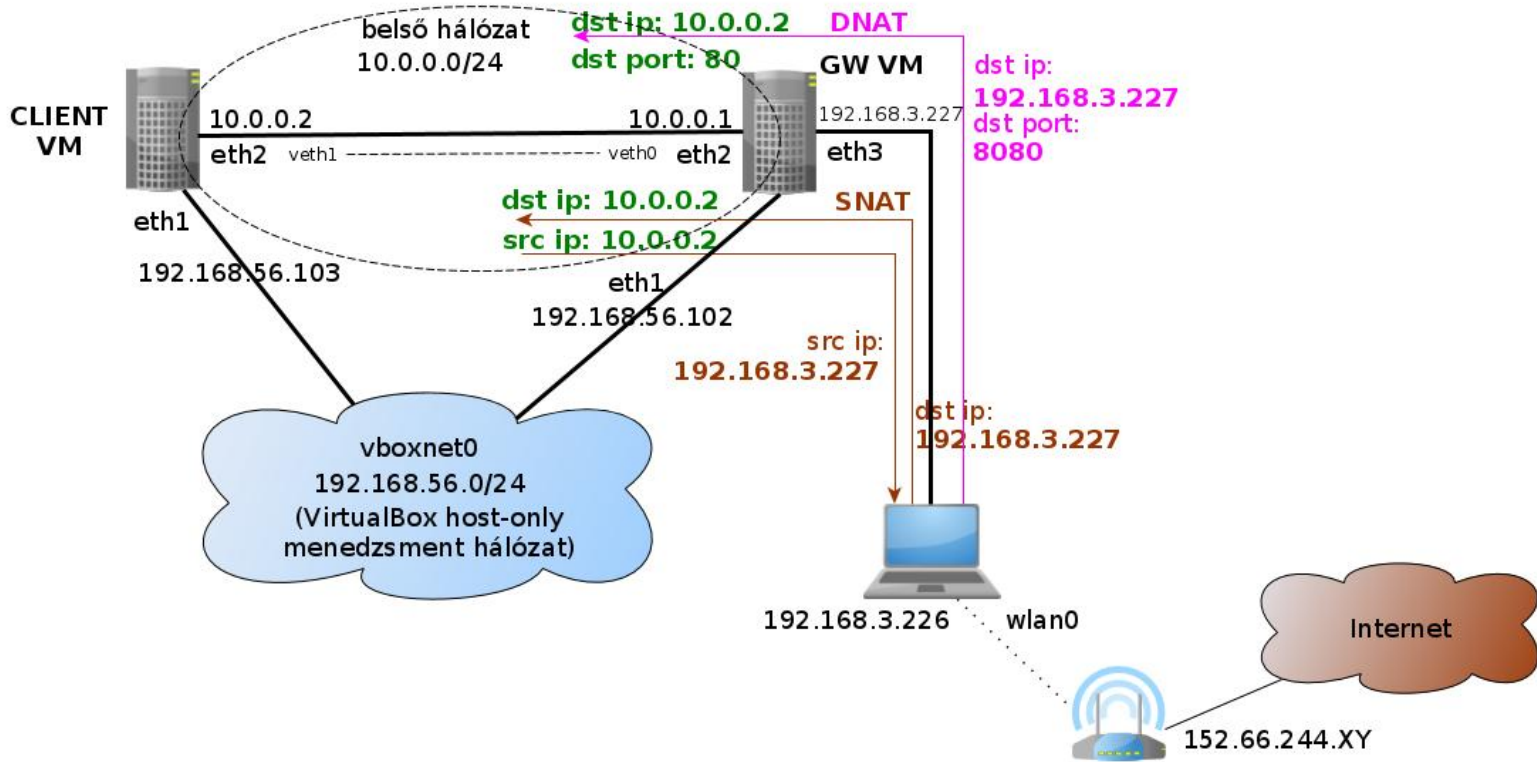


Hálózati elrendezés a gyakorlaton



ID: Teams-ben!

Hálózati elrendezés az előadáson



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`, `resolvectl status`)

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???

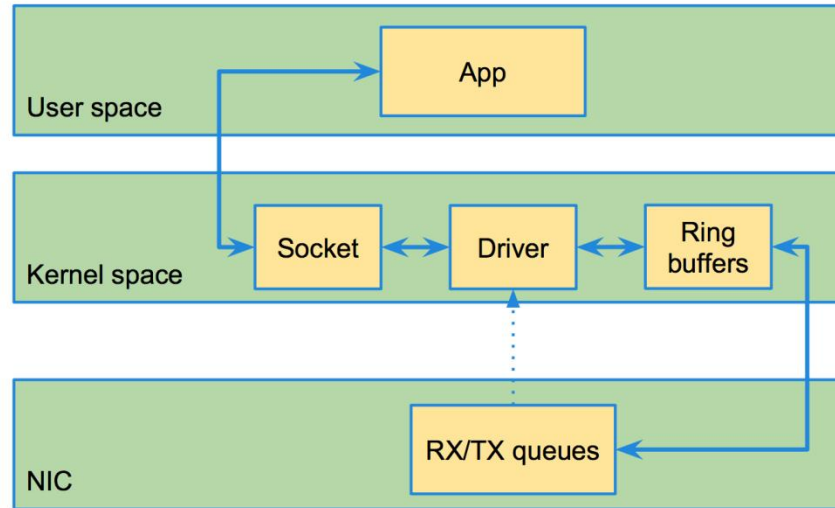
Hogyan tovább?

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

Egy csomag útja a Linux rendszerben

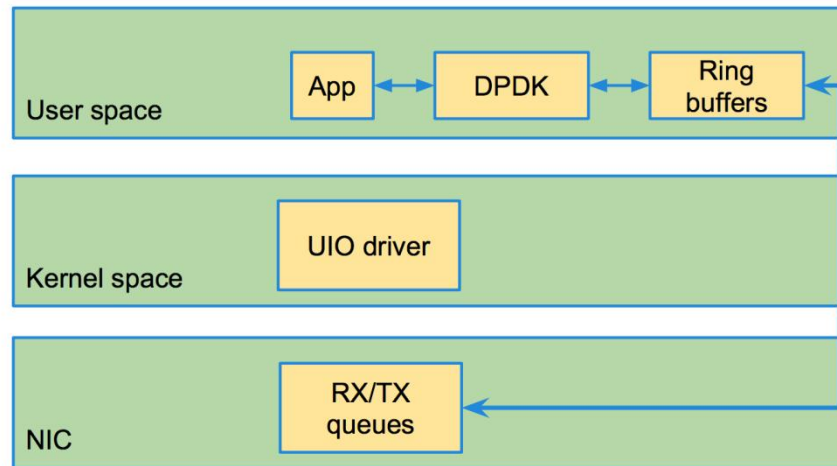
NIC-kernel space-user space

Packet processing in Linux



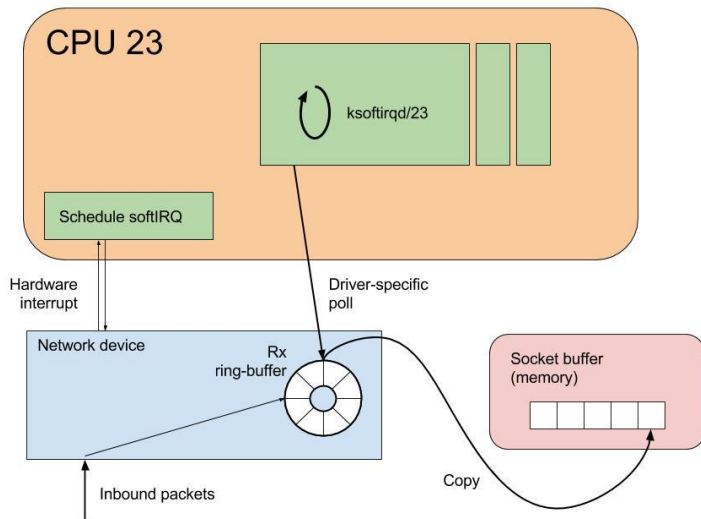
forrás: <https://github.com/rkuo/NetworkOS/blob/master/dpdk-summit-2015.md>

Packet processing with DPDK



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

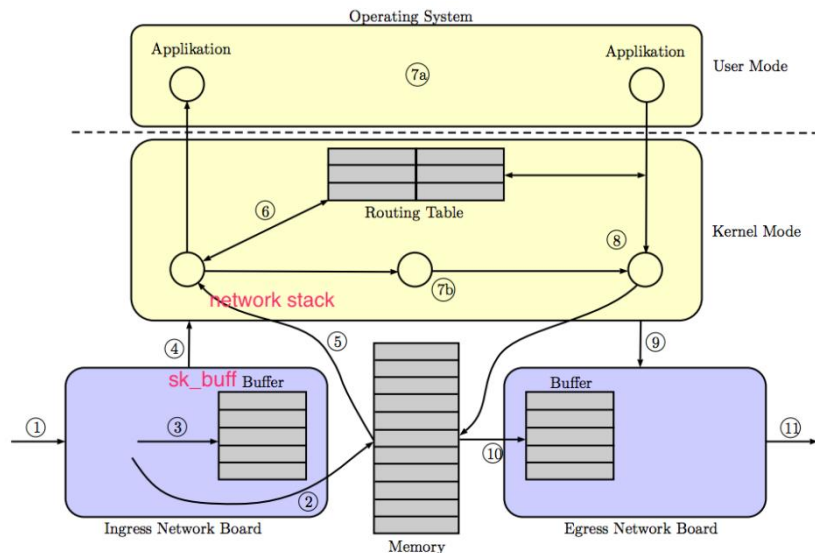


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

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

- ▶ (1) csomag érkezik a hálókártyára
- ▶ sk_buff struktúra
 - ▶ socket kernel buffer
 - ▶ ahogy a kernel reprezentál egy csomagot
- ▶ NIC driver ring buffer (3)
 - ▶ pointerok 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

Csomagfeldolgozás

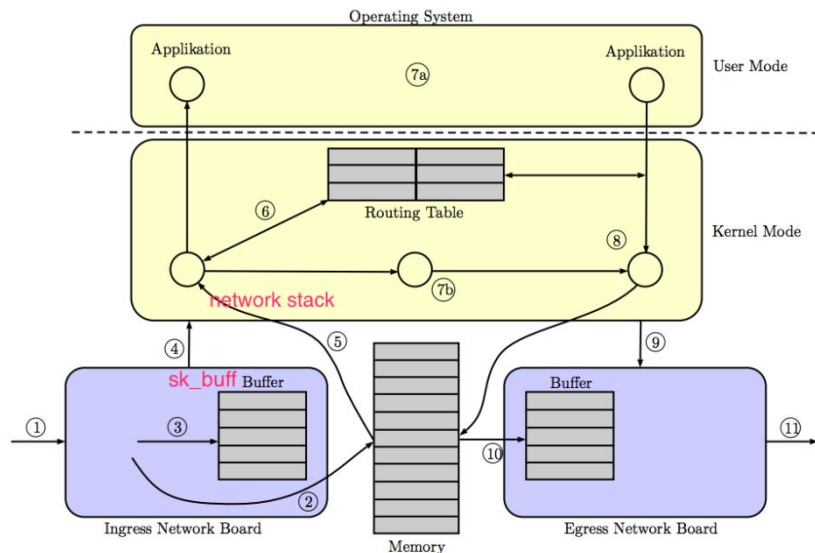


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

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

- ▶ csomagellenőrzés (integrity-verification, checksum)
 - ▶ 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

Csomagfeldolgozás

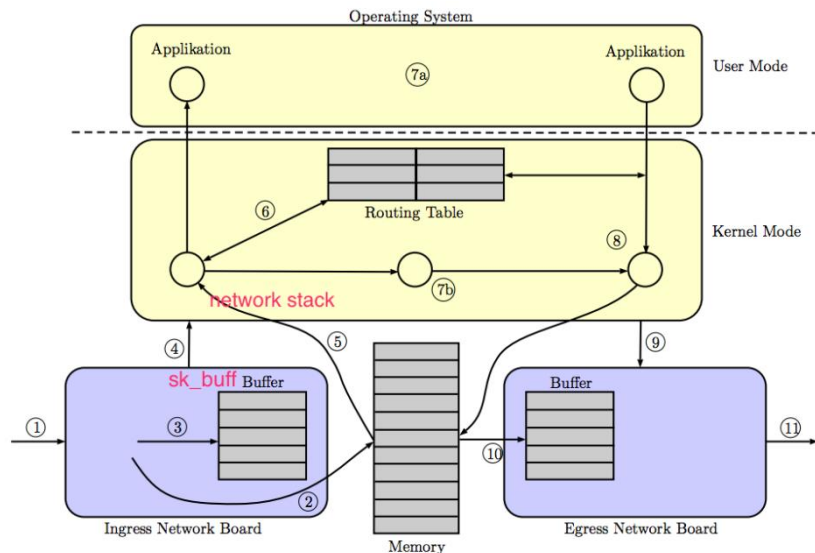
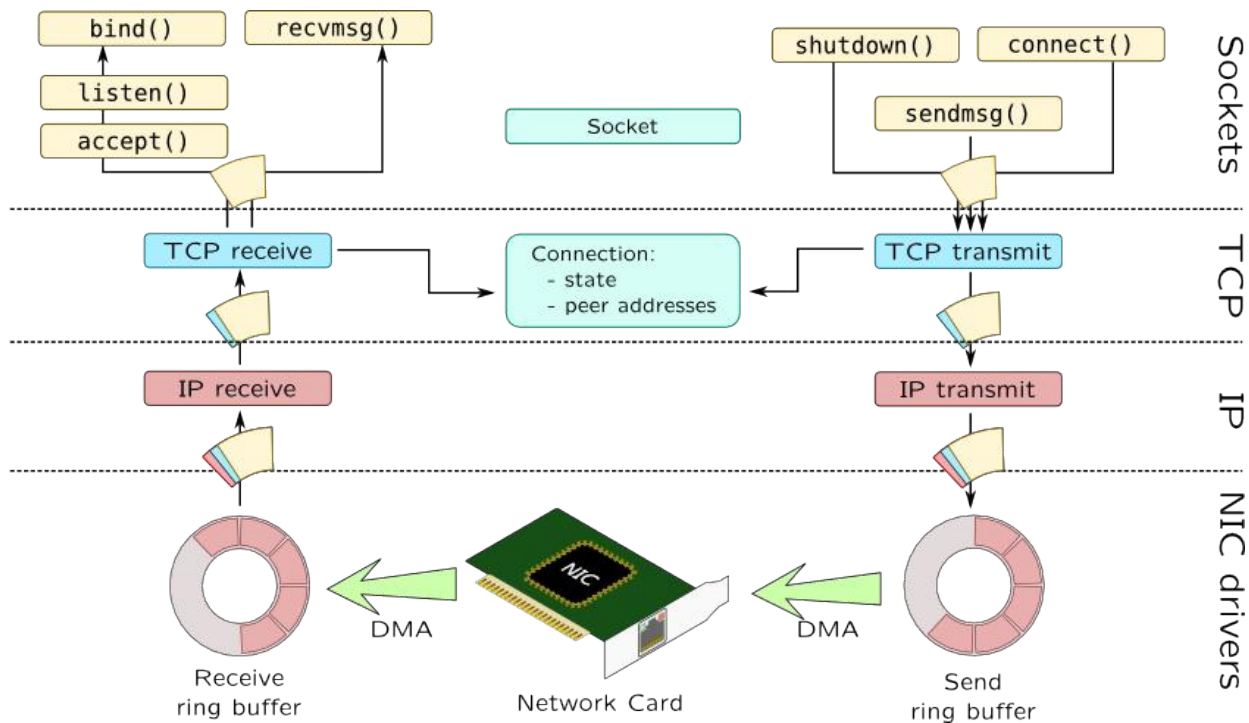


Figure 1: Abstract model of the packet processing steps in Unix-based software routers

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

- ▶ (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

Mindez felülről



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

NAT

iptables

Network Address Translation

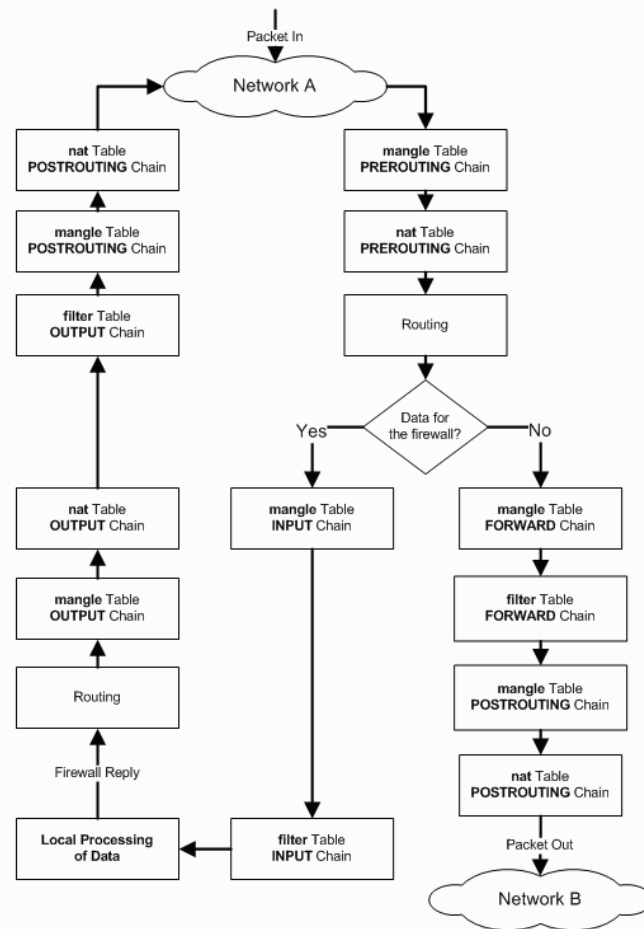
- ▶ NAT
 - ▶ olyan router, ami megváltoztatja a forrás vagy/és cél IP címet egy csomagban
 - ▶ leggyakrabban privát IP alhálózatot kapcsol a publikus internethez
- ▶ PAT (Port Address Translation)
 - ▶ a forrás vagy/és cél TCP/UDP port számot módosítja
 - ▶ általában beleértjük a NAT-ba
- ▶ SNAT (Source 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
 - ▶ kívüljár 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 láncához)
 - ▶ `-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)
 - ▶ `-j MASQUERADE` (source IP cím fordítása dinamikusan)
(output interfésztől függően)

2 opció: SNAT vs
MASQUERADE

SNAT konfigurálása (CLIENT)

- ▶ default gateway beállítása
 - ▶ `sudo route add default gw 10.0.0.1 dev eth2`
- ▶ tesztelés
 - ▶ `ping 8.8.8.8`
 - ▶ `ping telex.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

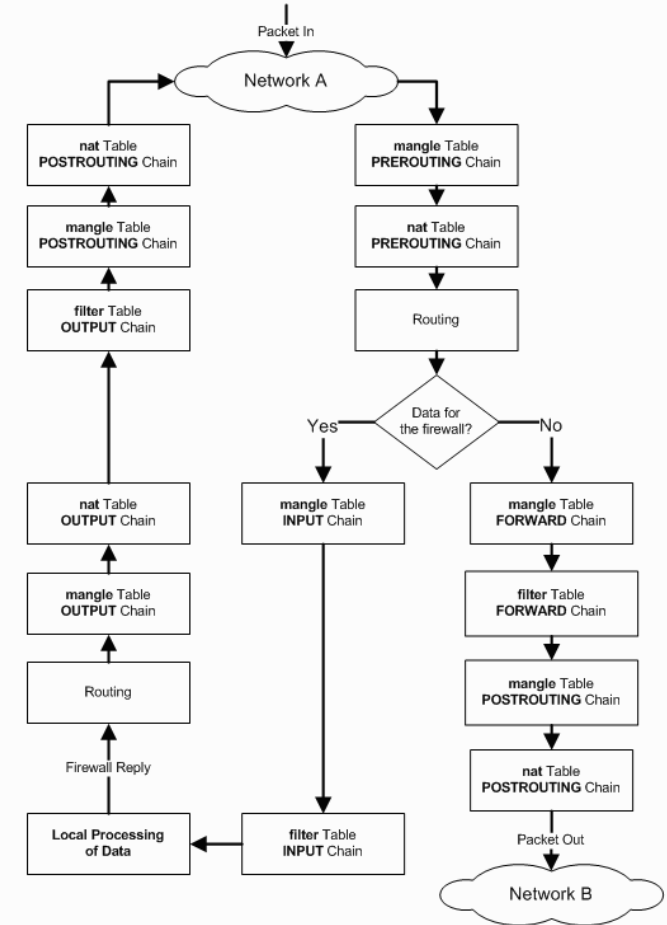
iptables

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

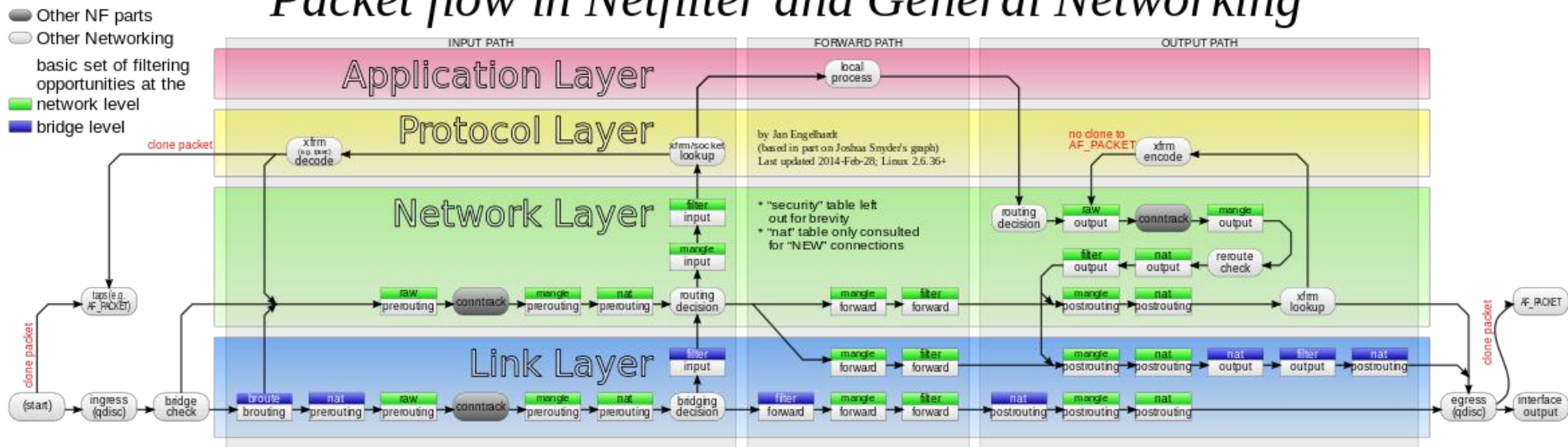
iptables

- ▶ 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...)

Packet flow in Netfilter and General Networking



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 láncan 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 láncan 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 előtt a szabályt nem illeszkedőnek vennénk
- ▶ korlátozás token bucket segítségével
 - ha van a vödörben token, akkor a bejövő csomagot elfogadjuk
 - ha nincs, akkor nem illeszkedőnek minősítjük
 - elfogadásnál a vödörből kiveszünk egy token
 - 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 az eldobás előtt 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 (check lo interface!)
 - ▶ **iptables -A INPUT -p udp --sport 53 -j ACCEPT**
 - ▶ **iptables -A INPUT -p udp --dport 53 -j ACCEPT**
 - ▶ CLIENT->GW
- ▶ internet->GW

Firewall konfigurálása (GW)

Átmenő forgalom szűrése
(FORWARD lánc)

```
(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
```

Tesztelés

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

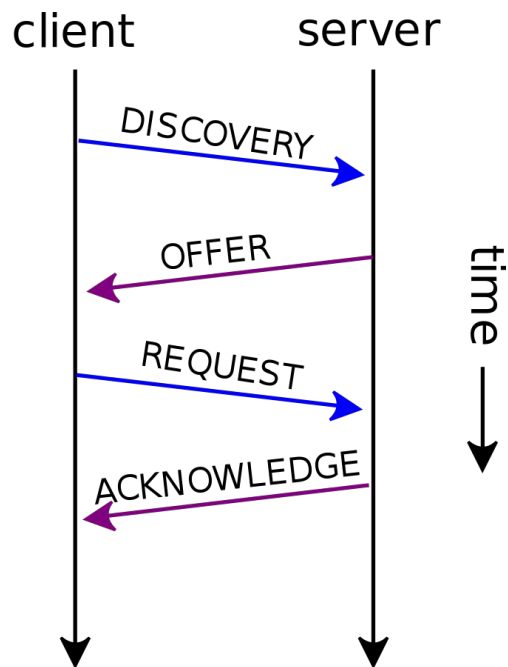
- ▶ internet -> CLIENT

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!

No.	Time	Source	Destination	Protocol	Info
40383	1687.343978	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transac
40385	1687.647466	192.168.1.200	255.255.255.255	DHCP	DHCP Offer - Transac
40386	1687.647535	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transac
40387	1687.653918	192.168.1.200	255.255.255.255	DHCP	DHCP ACK - Transac

Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 192.168.1.158 (192.168.1.158)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: CadmusCo_5e:38:76 (08:00:27:5e:38:76)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) DHCP Server Identifier = 192.168.1.200
Option: (t=51,l=4) IP Address Lease Time = 6 hours
Option: (t=81,l=24) Client Fully Qualified Domain Name
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=15,l=15) Domain Name = "classdemo.local"
Option: (t=3,l=4) Router = 192.168.1.1
Option: (t=6,l=4) Domain Name Server = 192.168.1.1
End Option

0120 a8 01 c8 33 04 00 00 54 60 51 18 03 02 02 77 32 ...3...I Q...w2
0130 30 30 33 2e 63 6c 61 73 73 64 65 6d 6f 2e 6c 6f 003.clas sdemo.lo
0140 63 61 6c 01 04 ff ff ff 00 0f 0f 63 6c 61 73 73 cal..... ..class
0150 64 65 6d 6f 2e 6c 6f 63 61 6c 03 04 c0 a8 01 01 demo.loc al.....
0160 06 04 c0 a8 01 01 ff

Installáljunk dhcp szervert!

- ▶ `sudo apt-get install isc-dhcp-server`
 - ▶ (Ubuntu-t vagy Debian-t feltételezzük)
 - ▶ Internet Software Consortium implementációja
 - ▶ korábban `dhcp3-server`
 - ▶ alapból nem indul, státusz ellenőrzés
 - ▶ `sudo systemctl status isc-dhcp-server`
 - ▶ régebben: `sudo service isc-dhcp-server status`
 - ▶ `isc-dhcp-server stop/waiting`

Konfiguráljunk dhcp szerveret!

▶ 1. lépés

- ▶ beállítjuk az interfészeket, ahol DHCP kéréseket kezelünk
- ▶ /etc/default/isc-dhcp-server fájlban:
 - ▶ INTERFACES="eth2"

▶ 2. lépés

- ▶ konfiguráljuk a szerveret
- ▶ /etc/dhcpd/dhcpd.conf

```
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.102 10.0.0.150;  
    option domain-name-servers 8.8.8.8, 152.66.115.1;  
    # option domain-name-servers 10.0.0.1;  
    option domain-name "haepuz.hu";  
    option subnet-mask 255.255.255.0;  
    option routers 10.0.0.1;  
    option broadcast-address 10.0.0.255;  
    default-lease-time 600;  
    max-lease-time 7200;  
    host client1 {  
        hardware ethernet 08:00:27:54:9f:81;  
        fixed-address 10.0.0.101;  
    }  
    deny unknown-clients;  
}
```

▶ 3. lépés

- ▶ isc-dhcp-server service indítása:
 - ▶ sudo systemctl start isc-dhcp-server (régebben: sudo service isc-dhcp-server start)
 - ▶ sudo systemctl status isc-dhcp-server (régebben: 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.
```

▶ GW-en

- ▶ Wireshark
- ▶ `journalctl -b -u isc-dhcp-server -f`

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
```

- ma már: `resolvectl status`
- vagy: `systemd-resolve --status`

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

dhcpcd.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

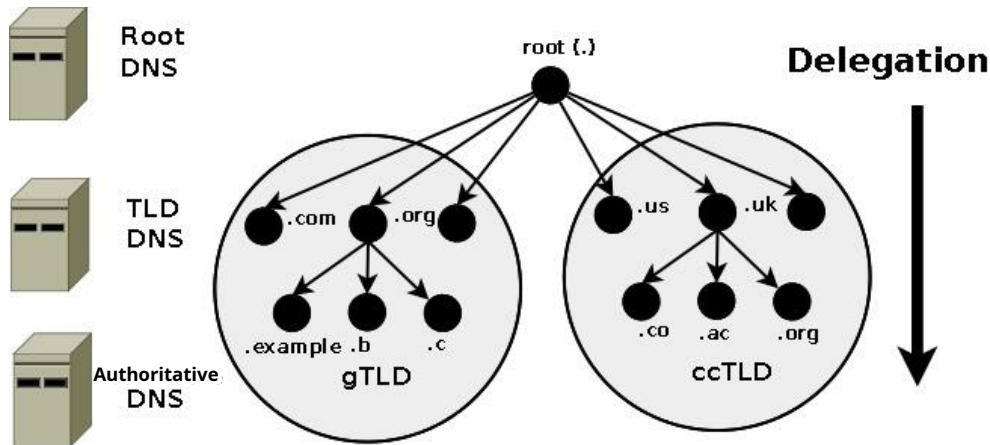
▸ Domain Name System

- 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
 - ma már persze minden systemd-ben... ()
- 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

► Komponentek

► 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:

- ▶ host, nslookup, dig parancsok
- ▶ közben wireshark

```
sonkoly : bash — Konsole
File Edit View Bookmarks Settings Help
sonkoly@notty:~$ dig -t A @8.8.8.8 tmit.bme.hu

; <<>> DiG 9.10.3-P4-Debian <<>> -t A @8.8.8.8 tmit.bme.hu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31147
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 512
;; QUESTION SECTION:
;tmit.bme.hu.                                IN      A

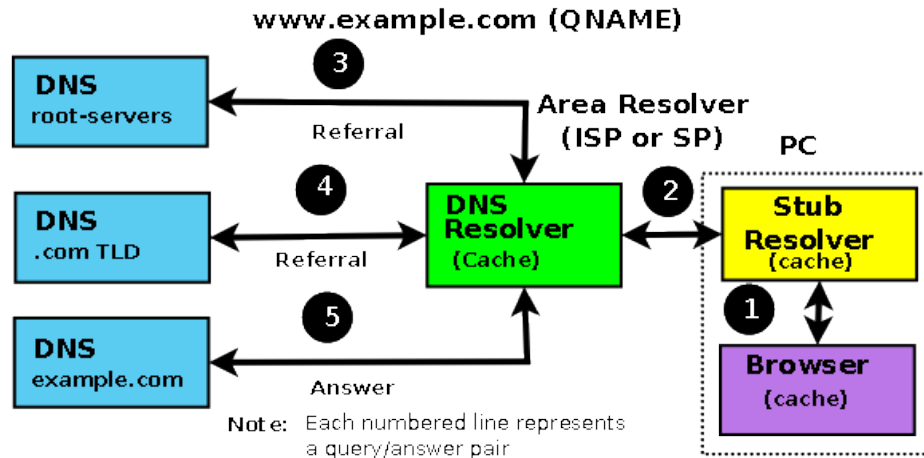
;; ANSWER SECTION:
tmit.bme.hu.                                21406   IN      A      152.66.244.17

;; Query time: 4 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Oct 02 18:48:48 CEST 2016
;; MSG SIZE rcvd: 56

sonkoly@notty:~$ █
```

DNS Query

Recursive and Iterative Queries



Item {2} is a Recursive Query - one question gives one complete answer
Items {3}, {4} and {5} are Iterative queries which may return either a Referral or an answer

- ▶ (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 szerveret, referral-t küld
- ▶ (4) újabb (iterative) query, most a TLD-nek
- ▶ válasz: referral
- ▶ (5) (iterative) query example.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/qSeWNF90Tm+e3PtCaAAGB5sDvwGLR5q+JtSW5 HhInM/4npW05Izj4giml05Qx7vHBzERiYPrfu+6ZiBrBV2HR8RaUly
W 60MgissUuN3vISPMJlNUlTofCMVztWk1YoI+hweyXJvf0K/j7l7iPX1 41Nb6K1sRV7tfc6C8mIwceVbVd7N8Uitfz0tHeHfLneoTXSS
ttrzaUcMj CevDvZb1Gp90Gh92qmpq2MnBz8muRZiF3YsKats6dT/NHhbkDTR0L5 D0/ggIn/xkTAwCYF5GUAp38YdnizM5WbTAUUne
p1UqMD0b0shCqJnz LIzJ6w==
;; Received 525 bytes from 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 93FDCE134B52B1B8FDCADDAD9152B0F2CA6E90EE091500F6
24661C6B 9A5B74DA
hu. 86400 IN RRSIG DS 8 1 86400 20161015050000 20161002040000 39291 . ZnbOLc4
8018nWZ6vyV/T06cim4/d/QuX+c2osRJNc98Ic6+KET9oftMU iyexts+dy8qUN3pH5EmrLvlLmLPkvdBAtI1H0LX5b1XghCR0TWEfh4z+
/jicRmWJANjWfs8Lu+VX6xvQdKl0cKFNTqtNmC711PzVdCl0R/RmL6N d1rL3M+wxqJ/b0UMNB8rVPMxc2jbutTLMP6p1Z0kb8MA5AD6
IKWu1V5b tnxrF10/vrHtmOnnMyEoHhGke+XopRaytTvJZuk7TT6X6XmSVRE0zR5Bn E7gn4aIYy2xoiNGDCL1cU79veFbHrS0Qah8PZw7
nQekF6pua8VulhCR QwEcwb==
;; 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.
8ab2d3lrnel6sukrtn88nvihc9glv88.hu. 3600 IN NSEC3 1 1 5 09A45AA33A023A65 8ABU575811BQ3JAKRV90NOIv879FR3
NS SOA TXT RRSIG DNSKEY NSEC3PARAM
8ab2d3lrnel6sukrtn88nvihc9glv88.hu. 3600 IN RRSIG NSEC3 8 2 3600 20161029101812 20161001163105 23295 hu.
h14EJIKAYTE48dkjhXtjvBEj83zswQVPLrrhwV982zb6VUIp61XUg0M5 1dlF2ZLud67LntXHTeg8FWC3qJcCmA061tTh7yVUrn1Cgewu
UN9pAyR YxEaBmz+4hnfBRkCoJ8IJ+87PeMTYLfXrkd9wCrlrpEy/LoB27416vRY LNQ=
2r70vs0gsot7p8na739qe3o3ebeb54nm.hu. 3600 IN NSEC3 1 1 5 09A45AA33A023A65 2R80BPpANG52BCJDLBMQCUIKCJA21L4B
NS DS RRSIG
2r70vs0gsot7p8na739qe3o3ebeb54nm.hu. 3600 IN RRSIG NSEC3 8 2 3600 20161028140039 20160930223100 23295 hu.
CvoDhb0/f0y+ob80yuBFJ9byhm+l9qXFYU0HynTKLoX4VZIG8G1Ej5gc tXoXpPTB4b999RHaXxscEnxKc6GHFTKS59LZfN5EPJf5Cjnet
ZH2XKxN cgnR+jfw+PFkq6Z+6m53HB1L5IUsq07vn2Q/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)

```
(mininet) 192.168.
File Edit View Bookmarks Settings Help
File Edit Options Buffers Tools Conf Help
# HeEpUz
acl goodclients {
    10.0.0.0/24;
    localhost;
};

options {
    directory "/var/cache/bind";

    # HaEpUz
    recursion yes;
    allow-query { goodclients; };
    forwarders {
        8.8.8.8;
    };
    forward only;

    dnssec-validation auto;

    auth-nxdomain no;      # conform to RFC1035
    listen-on-v6 { any; };
};
```


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

► Wireshark

- Kliensen (lo, belső hálózat)
- GW-en (lo, belső/külső hálózat)

```
haepuz@haepuz-client:~$ host stanford.edu  
stanford.edu has address 171.67.215.200  
stanford.edu has IPv6 address 2607:f6d0:0:925a::ab43:d7c8  
stanford.edu mail is handled by 10 mxa-00000d07.gslb.pphosted.com.  
stanford.edu mail is handled by 10 mxb-00000d07.gslb.pphosted.com.  
haepuz@haepuz-client:~$  
haepuz@haepuz-client:~$ nslookup stanford.edu  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   stanford.edu  
Address: 171.67.215.200  
Name:   stanford.edu  
Address: 2607:f6d0:0:925a::ab43:d7c8  
  
haepuz@haepuz-client:~$
```

```
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 teszhálózat kialakítása
 - ▶ gateway
 - ▶ kliens (belső hálózat)
- ▶ Hálózati funkciók vizsgálata, konfigurálása
- ▶ NAT
 - ▶ SNAT, DNAT
 - ▶ iptables
- ▶ Firewall
 - ▶ iptables
- ▶ DHCP
 - ▶ isc-dhcp-server
- ▶ DNS
 - ▶ bind9

