Hálózatok építése és üzemeltetése

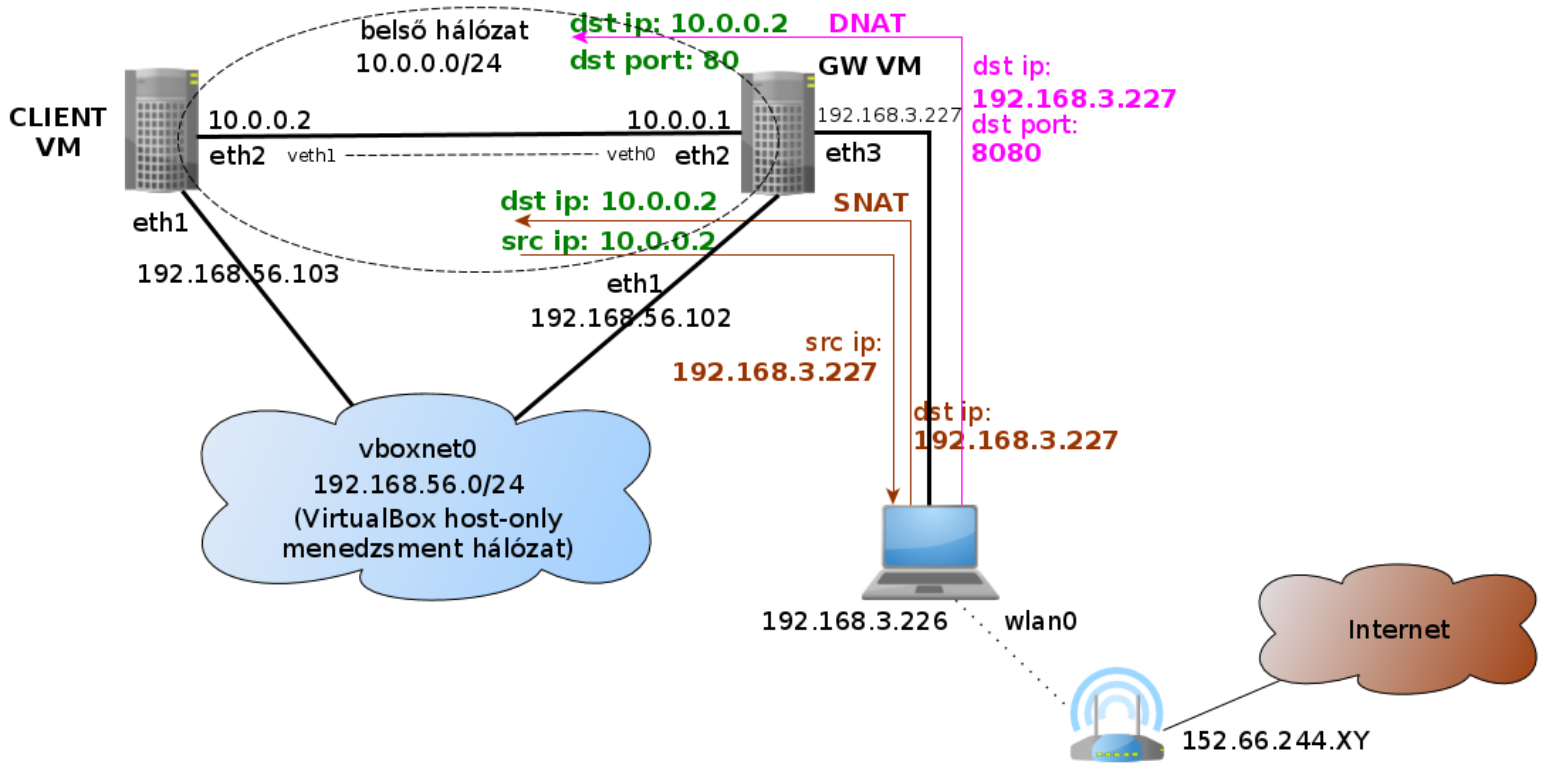
Hálózati funkciók a gyakorlatban

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

Firewall

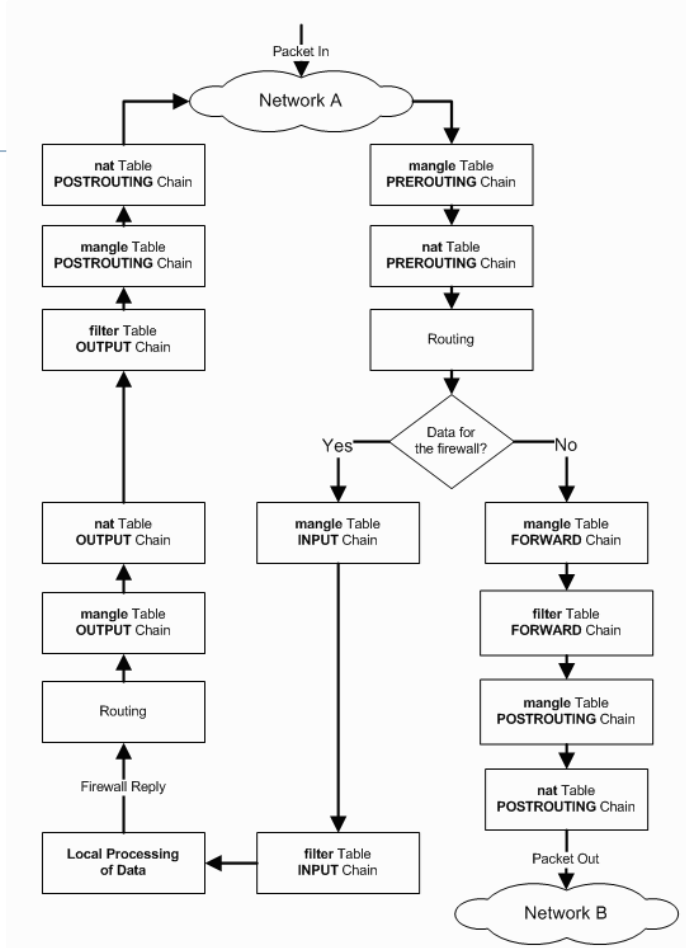
iptables

Firewall, tűzfal

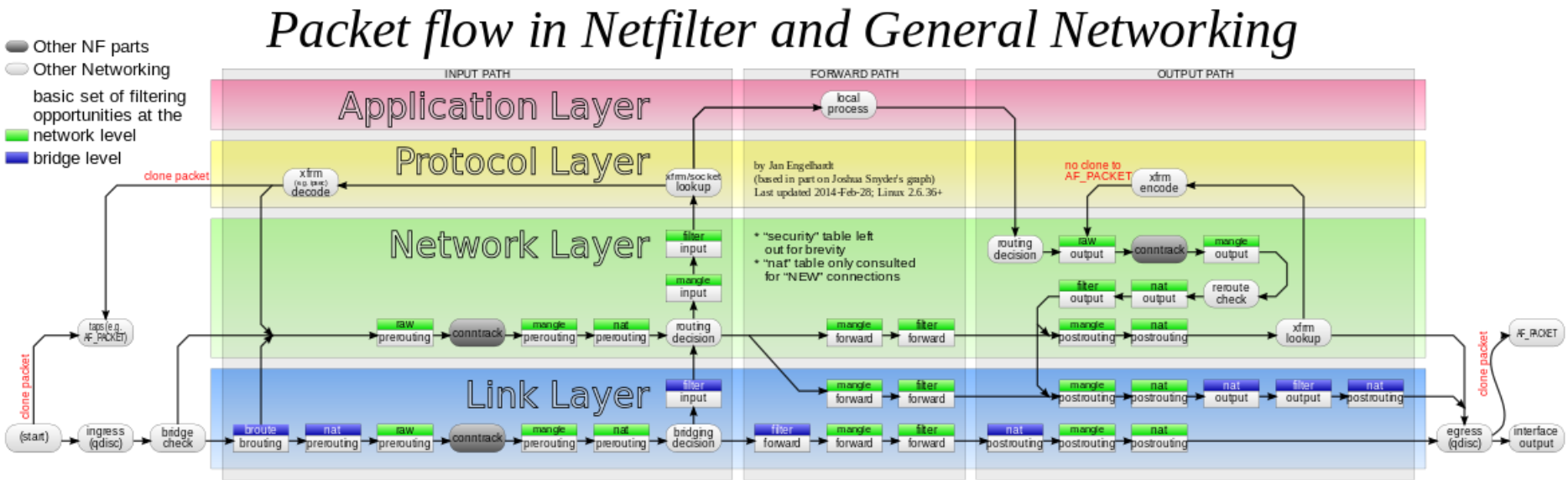
- Tűzfal: alapvető fontosságú eleme a hálózatnak
- Routing funkció mellett
  - döntést hoz, hogy adott forgalom mehet-e egyik hálózatból a másikba
  - sok (egyre több) tényezőt vehet figyelembe
  - csomag tartalmát is változtathatja
- Fajtái
  - stateless packet filters
    - egyenként vizsgálja a csomagokat (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

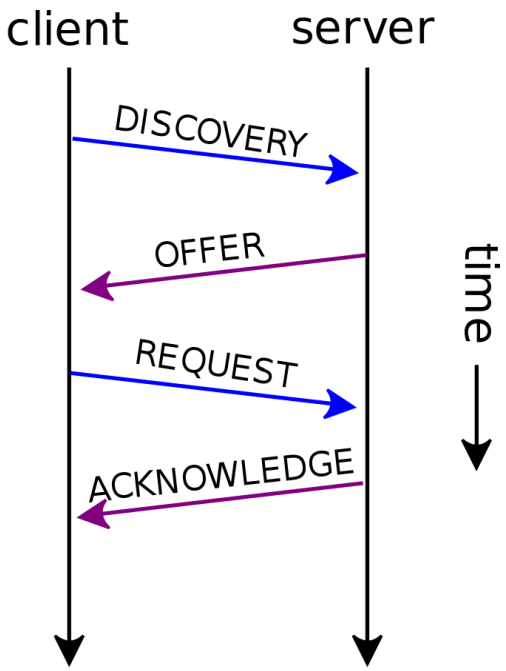
- ping, web, ssh, közben logok figyelése
  - CLIENT->internet
  - internet -> CLIENT

DHCP

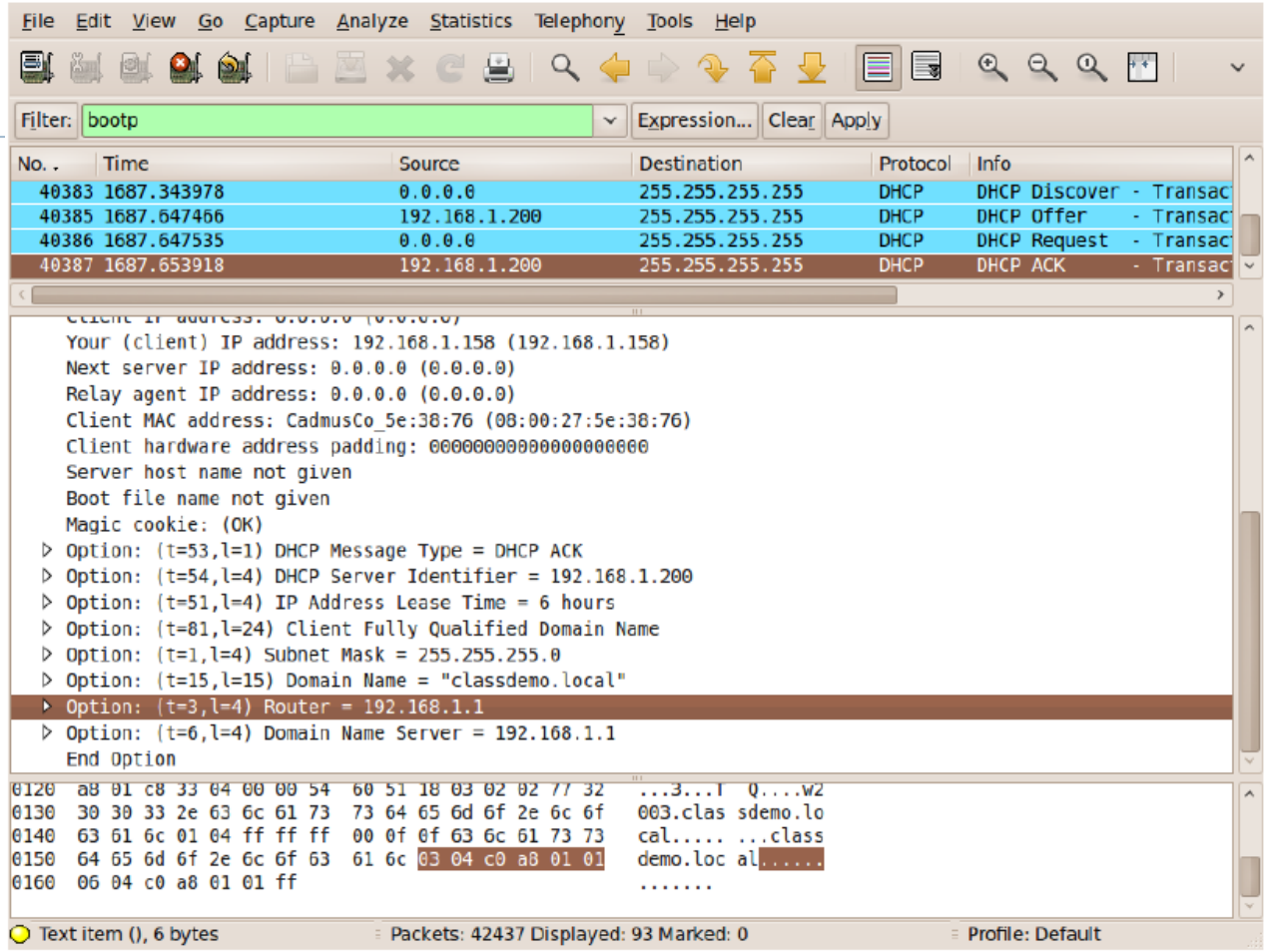
DHCP

- 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;}
```