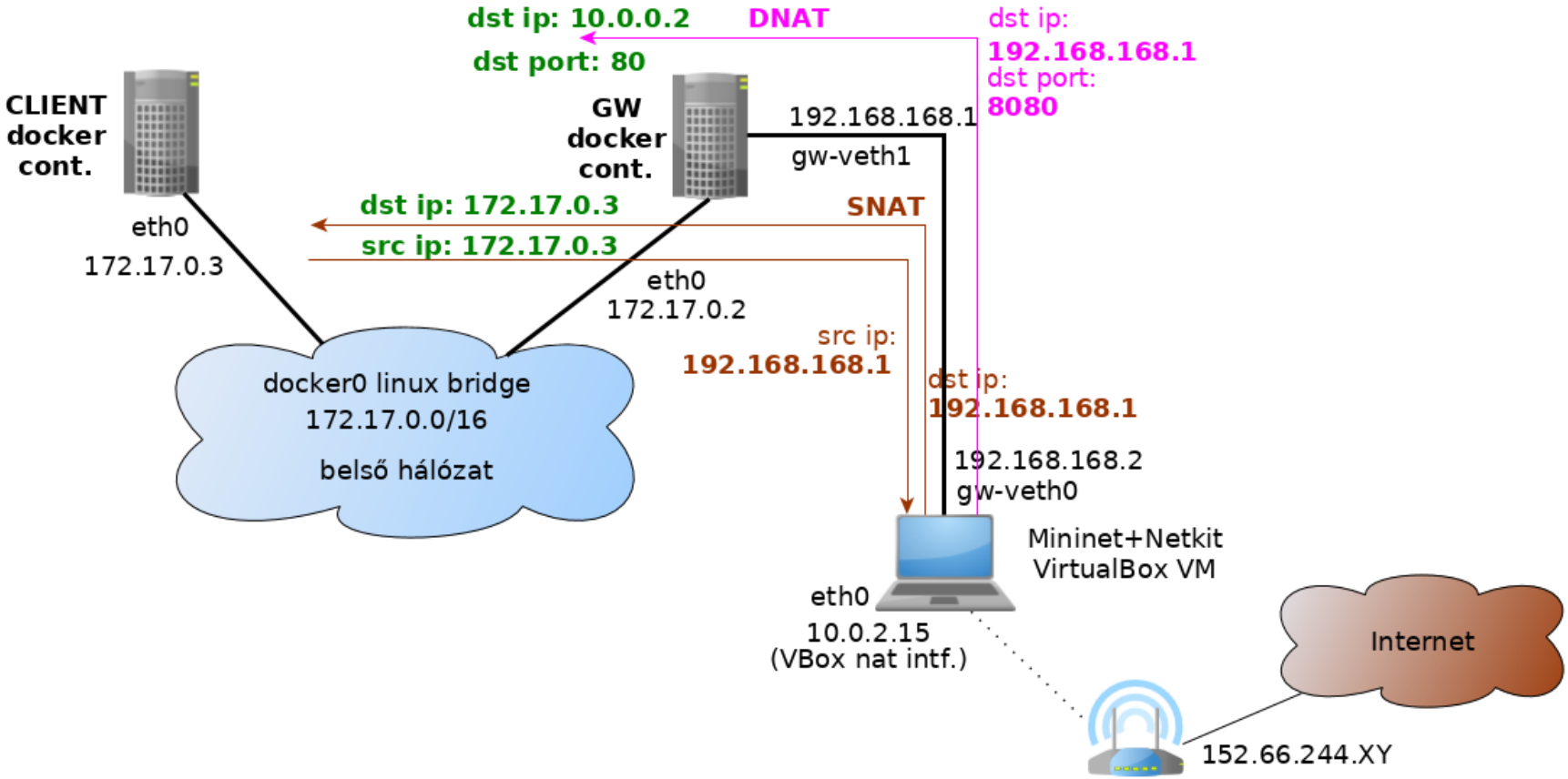
Hálózatok építése és üzemeltetése

Hálózati funkciók a gyakorlatban gyakorlat…

A példa hálózatunk

VirtualBox + Docker

Hálózati elrendezés



Előkészítés

- Virtuális gépben scriptek futtatása
  - wget http://152.66.244.65/~sonkoly/haepuz-gyak3.tgz
  - tar xvzf haepuz-gyak3.tgz
  - cd haepuz-gyak3
  - ./install-docker.sh
  - ./build-gw.sh
    - GW konténer létrehozása
    - 2 interfész
  - ./build-client.sh
    - CLIENT konténer létrehozása
    - csak a default interfész
  - (Dockerfile-ok, indító scriptek, akit érdekel…)

Ellenőrzés

- Image-ek, konténerek ellenőrzése
  - sudo docker images
  - sudo docker ps

Belépés

- Konténerkben van ssh szerver, kulcsok feltöltve
  - belépés a belső hálózatról
  - docker0 linux bridge: 172.17.0.1
    - ssh -i ~/haepuz-gyak3/lab_id_rsa root@172.17.0.2 (GW)
    - ssh -i ~/haepuz-gyak3/lab_id_rsa root@172.17.0.3 (CLIENT)
  - ha nincs ssh szerver
    - sudo docker exec -ti gw /bin/bash
    - egy bash shell indítása
  - konténeren belül nézzünk körül
    - ps auxf
    - ifconfig, route -n, resolv.conf, stb.

Összeköttetés tesztelése

- CLIENT <-> GW
- GW <-> külvilág (VBox host)
- CLIENT <-> külvilág (VBox host)

- mi működik, mi nem?

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

NAT

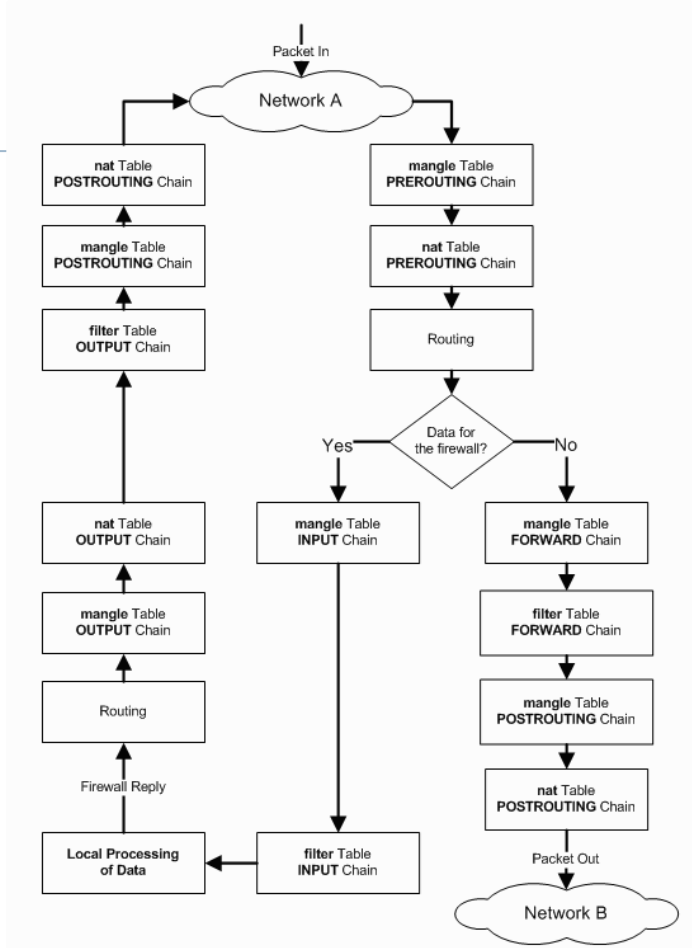
iptables

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

- 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

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 172.17.0.0/16 (ha ez a source IP)`
  - `-o gw-veth1  (ha ez az output interfész)`
  - `-j MASQUERADE  (akkor source IP fordítása dinamikusan)`
  - `(output interfésztől függően)`

SNAT konfigurálása (CLIENT)

- default gateway beállítása
  - `sudo route add default gw 172.17.0.2 [dev eth0]`
- tesztelés
  - ping 8.8.8.8
  - ping index.hu ???

DNAT konfigurálása (GW)

- Adott porton tegyük elérhetővé kívülről a belső gép
  - web szerverét (8080)
  - de előtte installáljuk! (apache2)
- címfordítás beállítása

DNAT konfigurálása (GW)

- Adott porton tegyük elérhetővé kívülről a belső gép
  - web szerverét (8080)
  - de előtte installáljuk! (apache2):
    - apt-get install apache2; service apache2 start
- címfordítás beállítása
  - `iptables -t nat`
  - `-A PREROUTING  (append, új szabály hozzáfűzése PREROUTING-hoz)`
  - `-d 192.168.168.1 (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 172.17.0.3:80  (a belső web szerverre)`

DNAT tesztelése

- VBox Host gépről
  - web browser
    - http://192.168.168.1
    - http://192.168.168.1:8080

Firewall

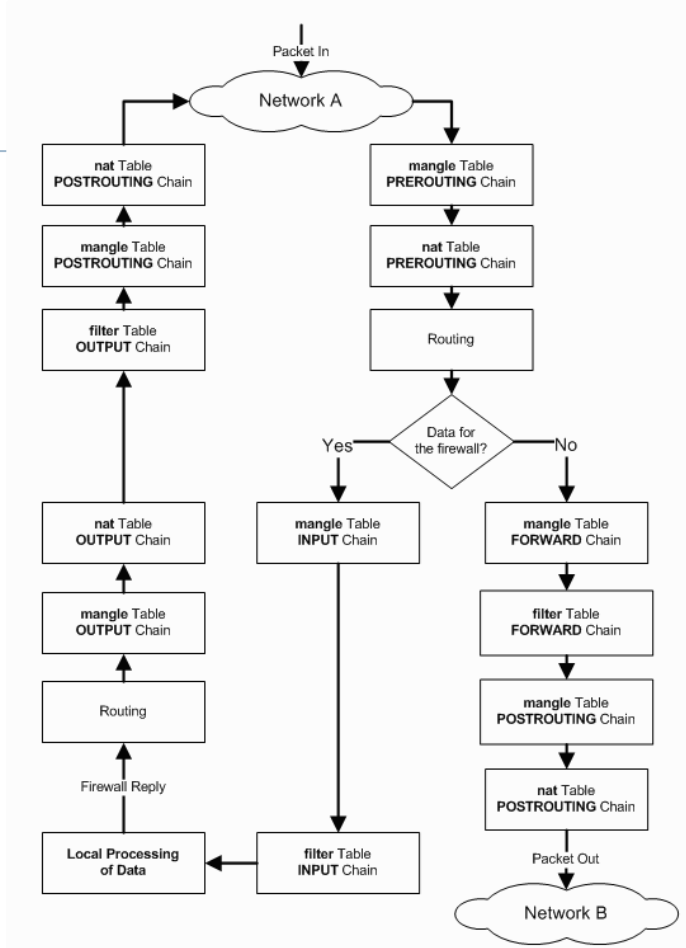
iptables

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



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)

- Feladat
  - az előadáson látott példa alapján konfiguráljuk fel a GW konténer FORWARD láncát hasonló funkcionalitásra
  - de most az ssh-val nem kell foglalkozni, csak a web forgalommal
  - szerkesztéshez pl. nano

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) előadáson bemutatott példa

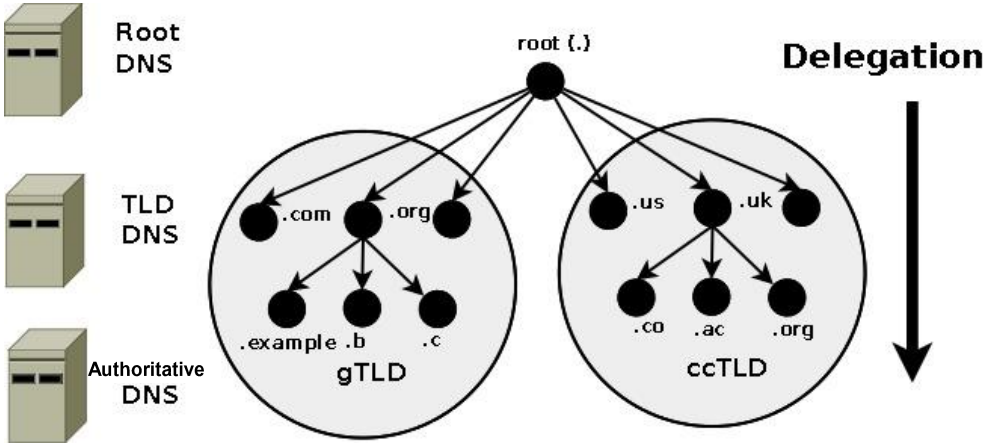
Tesztelés

- ping, web, közben logok figyelése
  - CLIENT->internet
  - internet (VBox Host) -> CLIENT

DNS

bind9

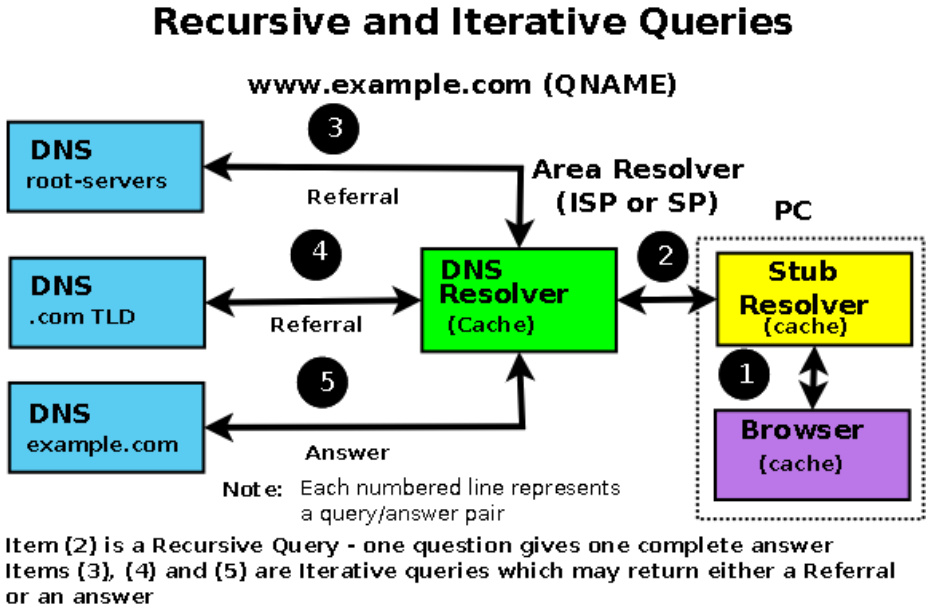
DNS szerverek, domain name delegáció



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

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/

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, tesztelés

- bind9 indítása
  - service bind9 start
  - service bind9 status
  - netstat -aputne | grep -i listen

- forgalom rögzítése
  - wireshark helyett tcpdump
  - tcpdump -i any -ne port 53 [-vvv]

CLIENT konfigurálása, tesztelés

- /etc/resolv.conf
  - nameserver 172.17.0.2

- tesztelés a kliens gépről (közben GW-en tcpdump)
  - dig stanford.edu +norecurse +short
  - dig stanford.edu +short
  - dig stanford.edu +norecurse +short

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
```

Illusztráció!

Összefoglalás

- Egyszerű teszthálózat kialakítása
  - VirtualBox host + Docker konténerek
  - GW, CLIENT, docker0 belső hálózat
- Hálózati funkciók vizsgálata, konigurálása
- NAT
  - SNAT, DNAT
  - iptables
- Firewall
  - iptables
- (DHCP most kimaradt)
  - isc-dhcp-server
- DNS
  - bind9