chroot

chroot [options] hakemisto [komento]

- “Virtuaalinen levyjärjestelmä”
- Vaihtaa juurihakemiston: annettu hakemisto toimii uutena juurena, sen ulkopuolella olevat eivät näy
- Uuden juuren alla pitää olla kaikki tarvittava, kuten /bin, /etc, /usr (riisuittuina turhista tiedostoista) ja dynaamiset kirjastot (/lib)

chroot

- Käytetään asennusvaiheessa (chroot /target …), etenkin asennuspakettien teossa hakemistonäkymän muuttamiseen
- Käytetään usein myös sovelluksen tietoturvan parantamiseen ajamalla niitä dedikoidussa hakemistopuussa (ei kovin tehokas suoja, root pääsee yleensä ulos), esim. vsftpd

DNS

- Nimipalvelu (Domain Name Service) yhdistää nimet IP-osoitteisiin (ja päinvastoin) ja vähän muuhunkin.
- Globaali hajautettu tietokanta, jota hallinnoi ICANN (Internet Corporation for Assigned Names and Numbers).
- Palomuurin sisäpuolella usein on yksityinen DNS-palvelin, joka tuntee sen privaattiosoitteet. Siten saman koneen IP voi näyttää erilaiselta sen mukaan kysytäänkö sitä palomuurin sisä- vai ulkopuolella (”split DNS”).

DNS

- Nimipalvelimia on kaksi perustyyppiä:
  - *authoritative nameserver* tuntee ja hallitsee itse jonkin (sille delegoidun) domainin nimet; sisäisesti voi olla *master* tai *slave*
  - *recursive nameserver* ei tunne itse nimiä, mutta osaa hakea niitä maailmalta; *caching nameserver* pitää lisäksi vanhoja vastauksia tallessa ja palauttaa uudelleen kysyttäessä ne saman tien hakematta niitä uudestaan (elleivät liian vanhoja ts. TTL ei ylittynyt)
  - sama palvelin voi olla sekä rekursiivinen että autoritatiivinen (ei yleensä suositeltavaa)

DNS

- Yksittäisessä koneessa voi olla vain sen tiedossa olevia nimi-osoite -pareja tiedostossa /etc/hosts. Kaikki palvelut eivät kuitenkaan käytä /etc/hosts'ia vaikka sellainen olisikin, erityisesti sähköpostipalvelimet usein eivät.
- Koneella voi olla myös vain sitä itseään palveleva lokaali nimipalvelinohjelma.
- /etc/resolv.conf määrää mitä nimipalvelimia käytetään. Usein automaattisesti ylläpidetty (resolvconf-paketti).

DNS

- Nimipalvelinohjelmia on useita:
  - **bind** (named), referenssitoteutus
  - **nsd** (vain autoritatiivinen)
  - **dnsmasq** (yleinen pienissä sisäverkoissa)
    - käytössä kurssin sisäverkossa
  - **unbound** (vain rekursiivinen)

DNS records

- DNS-tietokanta muodostuu tietueista (records), joita on useita tyyppejä, tärkeimmät:

  A koneen IPv4 -osoite

  AAAA koneen IPv6 -osoite

  CNAME alias, viittaa toiseen nimeen (ei osoitteeseen)

  MX koneen (tai domainin) sähköpostia välittävä kone
  (A- tai AAAA-tietue, ei CNAME)

DNS records

TXT	vapaa tekstikenttä
PTR	pointer, yhdistää osoitteen nimeen
NS	domainin nimipalvelimen nimi (osoite)
SOA	Start Of Authority, domainin perustiedot
CAA	Certification Authority Authorization, SSL-sertifikaattien toimittaja

- Kaikilla tietueilla on oma voimassaoloaikansa (TTL, Time To Live), tyypillisesti muutamasta tunnista muutamaan vuorokauteen (kannattaa lyhentää jos odotettavissa on muutostarpeita)

DNS zone files

- Zone-tiedosto on yhden (delegoidun) domainin DNS-tiedot sisältävä tiedosto (voi olla oikea tiedosto tai tietokantakin). Esimerkki:

```
; tkvk.org.zone
$TTL            86400
@               IN      SOA     ns1.tkvk.org. hostmaster.tarvainen.info. (1 43200 3600 604800 86400)
                IN      NS      ns1.tkvk.org.
                IN      NS      ns2.tkvk.org.
                IN      MX      10      hauki.tapanitarvainen.fi.
                IN      MX      10      leuka.tarvainen.info.
                IN      MX      90      kannel.tarvainen.info.
                IN      A       80.66.162.88
```

DNS zone files

```
ns1          IN      A        80.68.90.32
ns2          IN      A        64.79.206.244
www          IN      A        80.66.162.88
www2         IN   CNAME  www.tktk.org.
```

- $TTL määrää oletus-TTL:n, @ viittaa domainiin itseensä (”tkvk.org”), domain-kentän puuttuessa se tulkitaan samaksi kuin edellinen.

DNS-domainit

- Top-Level Domains (TLD):

  Generic TLD (gtld), esim. .com, .org, .website

  Country-Code TLD (ccTLD), esim. .fi, .uk, .eu

  Infrastructure TLD .arpa (Address and Routing Parameter Area)

  Vain sisäiseen käyttöön: .localhost, .example, .invalid, .test

  - Määritelty juuripalvelimissa (root servers)
  - Sopimus suoraan ICANNin kanssa

DNS-domainit

- Second-Level domains

  jyu.fi, **ties478**.website

  - Hankitaan rekisterinpitäjältä/jälleenmyyjältä
  - localhost.localdomain = loopback address (127.0.0.1, ::1)
- Third-Level (&c) domains

  it.jyu.fi, **tt1**.ties478.website

  - Domainin haltija voi tehdä itse
  - Joitakin kohdellaan kuten 2-tasoa (x.co.uk yms)

DNS: PTR

- PTR-tietue (*reverse* DNS) kertoo tiettyä IP-osoitetta vastaavan nimen (A- tai AAAA-tietueen). Se on toteutettu erikoisdomainilla **in-addr.arpa:**

  $ host 130.234.208.16

  16.208.234.130.in-addr.arpa domain name pointer lonka6.it.jyu.fi.

  - IPv6:lle on vastaavasti ip6.arpa:

    $ host 2001:41c9:1:422::228

    8.2.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.2.4.0.1.0.0.0.9.c.1.4.1.0.0.2.ip6.arpa domain name pointer kuha.tapanitarvainen.fi.

DNS: PTR

- PTR-tietuetta ei määritellä samassa zone-tiedostossa eikä yleensä edes samalla palvelimella kuin muut, nimiin liittyvät tietueet, vaan ao. ip-osoiteavaruuden haltijan palvelimella. Käytännössä PTR-tietueen muutokset pitää yleensä pyytää erikseen palveluntarjoajalta, vaikka muuten ylläpitäisi omaa nimipalveluaan.
- Erityisesti sähköpostipalvelimilla pitäisi A-tietueen ja PTR:n vastata toisiaan, muuten postikulussa voi tulla ongelmia.
- Testi esim. http://users.jyu.fi/~tt/misc/iptest.php

Dynaaminen DNS

- Joissakin ympäristöissä (erityisesti kotiliittymät ja mobiililiittymät) koneiden IP-osoitteet voivat muuttua automaattisesti. Dynaaminen DNS tarjoaa mahdollisuuden päivittää niiden nimitieto (lähinnä A record) automaattisesti osoitteen muuttuessa.
- DNS-palvelimen ominaisuus, riippumaton osoitteen määräävästä ISP:stä

Dynaaminen DNS

- Yksinkertainen mekanismi: asiakaskone ottaa yhteyden nimipalvelimeen, autentikoi itsensä jollakin tavalla ja ilmoittaa palvelimelle uuden osoitteensa
- Sisäänrakennettuna useissa palomuurilaitteissa (WLAN-tukiasemissa, ADSL-modeemeissa jne)

DNS: kyselytyökaluja

DNS-tietojen tutkimiseen on useita työkaluja:

host [*options*] [*name*] [*server*]

- Tärkeimmät optiot:

  -t *type* minkätyyppisiä tietueita haetaan (ANY = kaikki)

  -v verböösimpi tulostus

  -a sama kuin ”-v -t ANY”
- *server* on halutun nimipalvelimen osoite (nimi)
- *name* voi olla myös osoite

DNS: kyselytyökaluja

dig [*@server*] [*options*] [[-q] *name*] [[-t] *type*] ...

- optioita on paljon, mm.

+[no]tcp	käytetäänkö tcp:tä udp:n asemesta
+[no]trace	näytetäänkö rekursiivinen hakuketju
+[no]showsearch	näytetäänkö välituloksia
+[no]recurse	rekursiivinen haku
+[no]short	lyhyt tulostusmuoto
+[no]stats	statistiikkaa hausta

Split DNS

Yhteyttä kahden palomuurin sisäpuolella olevan koneen välillä ei haluta reitittää palomuurin ulkopuolelta. Kuitenkin halutaan käyttää samaa nimeä.

Ratkaisu: palomuurin sisäpuolella on oma nimipalvelin, joka palauttaa ao. nimistä niiden privaattiosoitteen, joka toimii vain palomuurin sisäpuolella (ja palomuuri kieltäytyy reitittämästä liikennettä ko. koneiden julkisiin osoitteisiin sisäpuolelta).

Split DNS: esimerkki

Palomuurin ulkopuolelta:

[tt@jalava ~]$ host tt1.student.it.jyu.fi

tt1.student.it.jyu.fi is an alias for s019.vm.it.jyu.fi.

s019.vm.it.jyu.fi has address 130.234.209.19

[tt@jalava ~]$ host s019.vm.it.jyu.fi

s019.vm.it.jyu.fi has address 130.234.209.19

Split DNS: esimerkki

Palomuurin sisäpuolelta:

tt@lonka5:~$ host tt1.student.it.jyu.fi

tt1.student.it.jyu.fi has address 172.20.209.19

tt@lonka5:~$ host s019.vm.it.jyu.fi

s019.vm.it.jyu.fi has address 130.234.209.19 # ei toimi!

vertaa:

tt@lonka5:~$ host s019.vm.it.jyu.fi 130.234.4.30

WHOIS

- Globaali tietokanta, jossa on domainien haltijoiden yhteystiedot, ja protokolla niiden hakemiseen.
- Nimipalvelurekisterien ylläpitämä, luotettavuus vaihtelee, usein julkinen tieto on proxy.
- Komentoriviclient ”whois”, esim:

```
$ whois jyu.fi
domain:   jyu.fi
descr:    Jyväskylän yliopisto
…
```

WHOIS

- Aina whois ei automaattisesti löydä oikeaa palvelinta. Tällöin sitä voi joutua kutsumaan useaan kertaan: kunkin top-level domainin whois-palvelimen pitäisi aina löytyä IANA:n (Internet Assigned Numbers Authority) palvelimelta, ja sieltä edelleen ao. TLD:n domainit:

  ```
  $ whois -h whois.iana.org fi

  …

  whois: whois.fi

  $ whois -h whois.fi jyu.fi
  ```

- Whois-järjestelmän tulevaisuus on epävarma (laiton EU:ssa!), todennäköisesti korvataan kokonaan eri järjestelmällä (RDS).

regular expressions

- Regular expression: yleinen merkkijonomalli, jossa erikoismerkeillä esitetään vaihtoehtoja
- Useita variantteja, yleisimmät:

  glob pattern

  basic regular expression

  extended regular expression

  perl regular expression

Glob pattern

- Glob pattern: yksinkertaistettu regexp, shellin yms jokerimerkkinotaatio:

  ? yksi mielivaltainen merkki

  * mielivaltainen määrä mitä tahansa merkkejä

  [acx1-9] yksi kirjain a, c tai x tai numero 1-9

  [!0-9] yksi merkki joka ei ole numero # tai [\!0-9]

  - Eri shelleissä erilaisia laajennuksia

Basic regexp

- Basic regular expression:
  - . (piste) yksi mielivaltainen merkki
  - * nolla tai useampia edeltäviä merkkejä
  - \{*m,n*\} edellistä merkkiä tai osalauseketta m-n kappaletta
  - ^ rivin alku
  - $ rivin loppu
  - [...] kuten glob patternissa, paitsi negaatio [^…] ja [[:space:]] ym
  - \(...\) rajoittaa osan lausekkeesta viitattavaksi myöhemmin
  - *n* *viittaa n*:nteen \(...\) -osalausekkeeseen
  - \ suojaa erikoismerkin, esim. \. = piste

grep

(”g/re/p”: get regexp and print)

- grep [options] {[-e] 'malli'|-f mallifile} [file...]

  tulostaa mallia vastaavat rivit
- Optioita paljon, mm:
  - -i ignore case (isot ja pienet kirjaimet samanarvoisia)
  - -v tulosta rivit joilla mallia *ei* esiinny
  - -F fixed strings (fgrep): vakiomerkkijono (* jne esittävät vain itseään)
  - -B *n* / -A *n* tulostaa *n* riviä ennen/jälkeen löydettyä
  - -c tulostaa vain löydettyjen rivien määrän
  - -q ei tulosta mitään (skriptissä testaamiseen)

grep-esimerkkejä

```
grep kala lajit.txt
grep -e kala -e lintu file # sekä kala- että linturivit
grep -i -B3 -A 1 error /var/log/syslog
grep '^tt:' /etc/passwd
sudo grep -F '*' /etc/shadow
sudo mount | grep -c ext4
```

grep-esimerkkejä

```
grep -v '^#' /etc/default/grub # ei-kommenttivit (myös tyhjät)
grep '^[^#]' /etc/default/grub # ei-tyhjät ei-kommenttirivit
grep '[0-9][0-9]*\.[0-9][0-9]*' # kahden luvun välissä piste
grep '^\([0-9][0-9]*\)[^0-9].*\1$' # alussa ja lopussa sama luku
grep '^kala$' # sama kuin grep -x 'kala': rivillä ei muuta
if grep -q 'sudo:.*[:,]tt' /etc/group ; then ...
```

Extended regexp

- egrep = grep -E, käyttää extended regexp'ejä:
  - * edeltävä merkki nolla tai useampia kertoja
  - + edeltävä merkki vähintään kerran
  - ? edeltävän merkki nolla tai yhden kerran
  - {*m,n*} edeltävän merkin toisto m-n kertaa (ei kenoviivoja)
  - ^ ja $ toimivat myös osalausekkeissa
  - ( ) rajaavat osalausekkeen ilman kenoviivoja
  - | ”tai” (myös osalausekkeen sisällä)
  - osalausekeviittauksia \1 jne ei ole (standardiversiossa)

Extended regexp

- Usein lyhyempi kuin basic regexp, esim.

```
egrep '[-+]?[0-9]+'          # vrt. grep '[-+]\{0,1\}[0-9][0-9]*'
```

- Seuraavat eivät onnistu basic regexpillä:

```
egrep '((kissa|koira)[ ;,])+'  # paljon kissoja tai koiria
egrep 'iso (lintu|kala) (lensi|ui)'
```

- Toisaalta esim. '\([a-z]\)=\1' ei onnistu extended regexpillä
- Erityisesti Gnu grep tuntee kaikenlaisia laajennuksia...

sed

- ”stream editor”: komentorivieditori, sopii käytettäväksi skripteissä yksinkertaisiin tiedoston muutoksiin ja komentoriviltä, etenkin jos interaktiiviset editorit eivät toimi. Komennot muistuttavat vi:tä.
- Käsittelee tiedostoa rivi kerrallaan (monen rivin muutoksia voi tehdä, mutta se on hieman vaikeaa)
- Nopea ja tehokas, tarvitsee hyvin vähän muistia

sed: syntaksi

- Syntaksi:

  sed [optiot] {-f scriptfile|[-e] 'script'...} [file...] [>outfile]

- Oletuksena kirjoittaa stdout'iin, optiolla -n ei tulosta muuta kuin erillisillä tulostuskomennoilla, optiolla -i muuttaa tiedostoa paikalla (-i.bak tallettaa alkuperäisen tarkenteella .bak)
- Editointikomennot voi antaa joko komentorivillä (optio -e, ei tarvita jos vain yksi) tai tiedostossa (-f), useita komentoja voi erotella puolipisteellä tai rivinvaihdolla
- Basic regular expressions

sed

- Komentojen yleinen syntaksi:

  [<osoite>[,<osoite2>]]komento[argumentit][liput]

  - Osoitteet rivinumeroita tai malleja (/regexp/), $=loppu (viimeinen rivi), !=negaatio
  - Erikoismerkkejä (usein myös /) voi suojata \:lla
  - Joillekin komennoille voi antaa vain yhden osoitteen, joillekin myös kaksi (= väli)
  - Komennot yksikirjaimisia, enimmäkseen samoja kuin vi (vim) editorissa

sed: s

- s: substitute, korvaa merkkijono toisella

  s/malli/korvaus/[liput]

  - Oletuksena vain ensimmäinen per rivi, lippu g = kaikki, numero = korvaa n:s esiintymä
  - p = tulosta jos korvattiin, w *file* = kirjoita tiedostoon jos korvattiin
  - Malli basic regular expression (vähemmän laajennuksia kuin grepissä)
  - Korvausmerkkijonossa & = korvattava jono, *n* = *n*:s \(...\)

    sed '5,6s/lintu/kala/g' # korvaa kaikki linnut kaloilla riveillä 5-6

    sed 's+.+&/+g' # lisää jokaisen merkin perään /

    sed '/kala/s/^\(.\)\(.\)/\2\1/' # vaihda kaksi ens. merkkiä kalariveillä

sed: d

- d: delete
  - Ei optioita eikä argumentteja, mutta yleensä aina osoite

    sed -i 5d ~/.ssh/known_hosts # poista rivi 5

    sed /kala/d # poista kalarivit (vrt. grep -v kala)

    sed '/kala/!d' # poista kalattomat rivit (vrt. grep kala)

    sed '1,5d;9,15d' # poista rivit 1-5 ja 9-15

    sed '/kala/,$d' # poista rivit ensimmäisestä kalarivistä loppuun

    sed -e '2,\+/+d' # poista rivit riviltä 2 seuraavaan ”/”:n sisältävään riviin saakka

sed: p

- p: print
  - Tulostaa rivin (senhetkisessä muodossaan); ei optioita eikä argumentteja mutta yleensä aina osoite
  - Usein option -n kanssa

```
sed -n 15,20p       # tulosta vain rivit 15-20
sed -n /kala/p      # sama kuin sed '/kala/!d'
sed -n '/\//p'      # sama kuin grep /
sed /kala/p         # tulosta kalarivit kahdesti
sed '/kala/p;s/kala/lintu/g'
```

sed: w, r

- w: write to file

  sed '/kala/w kalafile.txt' infile >outfile

  # vrt: grep kala infile > kalafile.txt

  sed '/[/]/w kauttaviivarivit' # tai '\x/xw …' tai '/\//w …'

  sed -e '/kala/w kalat' -e '/lintu/w linnut'

  - Tiedostonimen edessä oltava tasan yksi välilyönti
- r: read from file

  sed '5r file' infile # lisää tiedoston ”file” sisältö rivin 5 jälkeen

  - vain yksi osoite

sed: { }

- { }: ryhmittely, esim.

```
sed '1,10{/^#/d;}' # poistetaan riveistä 1-10 #:lla alkavat
sed '1,50{s/kala/hauki/g;s/lintu/kana/g;}'
    # riveiltä 1-50 vaihdetaan kalat hauiksi ja linnut kanoiksi
sed '1,/kala/{/hauki/s/a/b/2;}'
    # alusta ensimmäiseen kalariviin sanan hauki-sisältäviltä
    # riveiltä vaihdetaan toinen a b:ksi
```

- Paljon muitakin komentoja ja optioita, ks.

  http://www.mit.jyu.fi/opiskelu/kurssit/unixshell01/sed.html

chmod: ”change mode bits”

- Muuttaa tiedoston oikeuksia
- Vanha numeerinen muoto:

  yksi oktaali numero/3 bittiä, kolme numeroa user-group-other -järjestyksessä: 4=r, 2=w, 1=x → 7=rwx, 5=rx jne

  jos nelinumeroinen, ensimmäinen numero 4=suid, 2=sgid, 1=sticky bit
- Symbolinen notaatio:

  [ugoa][+-=][rwxXst]

  u=user, g=group, o=others, a=all

  rwx=read,write,execute/search, s=suid tai sgid, t=sticky

  X = ehdollinen x (vain jos hakemisto tai jos jollakulla jo x)

umask

- Shellin (bash) sisäinen komento: asettaa oletusarvon luotavien tiedostojen oikeuksille (rwx)
- Vanha syntaksi oktaalinen bittimaski **poistettaville** oikeuksille, esim.

  umask 022 # ryhmältä ja maailmalta w-oikeus pois

  umask 067 # ryhmältä rw pois, maailmalta kaikki pois
- Uusi syntaksi **sallitut** oikeudet kuten chmod'issa, esim. ylläolevat toisin esitettynä:

  umask u=rwx,g=rx,o=rx

  umask u=rwx,g=x,o=
- Ilman argumentteja tulostaa voimassaolevan asetuksen, oletuksena oktaalimuodossa, optiolla -S symbolisena