# telnet

- Telnet on alunperin tehty pääteyhteyden muodostamiseksi verkon yli, mutta koska siinä ei ole minkäänlaista salausta, siitä on luovuttu eikä telnet-demonia enää juuri missään käytetä. Telnet-client on kuitenkin yhä hyödyllinen avointen tcp-porttien testaamiseen ja tekstipohjaisten protokollien kokeilemiseen käsin:

  telnet *kone portti*

# telnet

- Esim. kokeillaan postipalvelimen toimintaa:

```
$ telnet smtp.example.org 25
Trying 10.23.4.25...
Connected to smtp.example.org.
Escape character is '^]'.
220 mail1.example.org ESMTP; Wed, 6 May 2015 07:51:39 +0300
quit
221 2.0.0 mail1.example.org closing connection
Connection closed by foreign host.
```

# traceroute

- Reititys- ja palomuuriongelmien selvittämisessä yksi perustyökalu on traceroute, joka yrittää selvittää mitä kautta paketti kulkee:

  traceroute [options] kone

- Usein mitään optioita ei tarvita, mutta jos jokin palomuuri välissä estää normaalin toiminnan, voi kokeilla vaihtoehtoisia menetelmiä:

  -I käytä ICMP ECHOa (ping)

  -T käytä TCP SYN-paketteja

- Muitakin optioita erikoistarpeisiin on, mm.

  -n älä hae koneiden nimiä DNS:stä

  -m *max_ttl* aseta yläraja hyppymäärälle (oletus 30)

  -w *waittime* kuinka monta sekuntia vastausta odotetaan (oletus 5)

# nmap

- Erityisesti palomuurien toiminnan testaamiseen hyödyllinen työkalu: kokeilee mitkä portit kohdekoneissa ovat auki. Optioita on eri tarkoituksiin paljon, esim.

  -Pn (aikaisemmin -P0): älä pingaa ensin (jos ping blokattu)

  -p22,1024-2047  kokeile portteja 22 ja 1024-2047

  -v  verböösimpi tulostus (-vv vielä enemmän)

# nmap

- Kohteena voi olla kone (nimi tai IP) tai verkkoalue tai useita, esim.

```
nmap 172.20.208.16  # etsi avoimet portit lonka6:sta
nmap -p80,443 172.20.209.0/24
# etsi www-palvelimet verkkoalueelta
```

- Erilaisia skannaustapoja löytyy man-sivulta lisää, jos kone ei vastaa vaikka pitäisi
- Älä skannaa toisten koneita tai verkkoja ilman lupaa!

# iptables: sekalaista

- Yleensä halutaan sallia kaikki liikenne localhostiin:

  iptables -A INPUT -i lo -j ACCEPT

  iptables -A OUTPUT -o lo -j ACCEPT
- Kerberos-PAM-autentikointia varten pitää sallia yhteys ulospäin Kerberos-porttiin (88):

  iptables -A OUTPUT -p tcp --dport 88 -j ACCEPT
- LDAP vaatii vastaavasti portin 389 (SSL:n kanssa 636):

  iptables -A OUTPUT -p tcp --dport 389 -j ACCEPT

  iptables -A OUTPUT -p tcp --dport 636 -j ACCEPT
- Yleensä jos jokin palvelu ei toimi, sen tarvitsemia porttinumeroita voi etsiä tiedostosta /etc/services (siellä voi olla lokaaleja lisäyksiä, joista Googlekaan ei tiedä)

# Palomuurin tyhjennys

- iptables ei tarjoa mitään keinoa kaikkien sääntöjen poistamiseen kerralla vaan kaikkien taulujen policyt, säännöt ja omat ketjut pitää poistaa erikseen. Yksinkertaisessa tilanteessa riittää:

  iptables -P INPUT ACCEPT

  iptables -P OUTPUT ACCEPT

  iptables -P FORWARD ACCEPT

  iptables -F

# Palomuurin tyhjennys

- Jos iptables halutaan tyhjentää perusteellisesti, se voidaan tehdä tähän tapaan (tarvittaessa samat ip6tables'illa):

```
for table in filter nat mangle raw ; do
    for chain in INPUT OUTPUT FORWARD POSTROUTING PREROUTING
    do iptables -t $table -P $chain ACCEPT 2>/dev/null || true ;  done
    iptables -t $table -F  # taulut tyhjiksi
    iptables -t $table -X  # omat ketjut pois
done
```

# Palomuurin tyhjennys

- Sääntöjen poistaminen ei poista iptables-moduleita kernelistä. Sekin voidaan haluttaessa tehdä modprobe -r -komennolla, mutta se ei yleensä ole tarpeen (modulit voi selvittää lsmod -komennolla, ne pitää poistaa riippuvuusjärjestyksessä, viimeisinä ip_tables, ip6_tables ja x_tables).
- Jos iptables halutaan kokonaan hävittää, senkin voi tehdä:

  apt-get purge iptables libxtables11

  Tätä ei yleensä tehdä kuin embedded-ympäristöissä tms.

# iptables: debugging

- Palomuurisääntöjen syntaksivirheet saa helpoiten kiinni, kun ne kirjoittaa skriptiin ja ajaa sitä -x ja ehkä myös -e optioilla (alkuun ”#! /bin/bash -ex” tai suoritettaessa bash -ex ...)
- Uudet säännöstöt kannattaa testata ensin **iptables-apply** -komennolla, jos konetta säätää ssh:n tms verkkoyhteyden yli. (Jos konsoliyhteydenkin saa, se kannattaa silti varmuuden vuoksi avata.)

# iptables: debugging

- Epäilyttävien DROP- ja REJECT-sääntöjen eteen voi laittaa LOG-säännön samalla ehdolla ja katsoa mitä lokissa näkyy. Joskus voi lokittaa myös läpi päästettyjä paketteja.
- Palomuurin toimintaa voi testata mm. komennoilla telnet (tcp-portit), ping (icmp), nmap (kaikki...) ja traceroute (etenkin useamman palomuurin läpi mentäessä).

# iptables: anti-spoofing

- IP-osoitteiden väärentäjien ja verkkosotkujen varalta kannattaa blokata ulkopuolelta tulevat privaatti- yms osoitteet, joita ei tiedetä omiksi tai muuten tarpeellisiksi:

```
iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP
iptables -A INPUT -i eth0 -s 172.16.0.0/12  -j DROP
iptables -A INPUT -i eth0 -s 192.168.0.0/16  -j DROP
iptables -A INPUT -i eth0 -s 224.0.0.0/4      -j DROP
iptables -A INPUT -i eth0 -s 240.0.0.0/5      -j DROP
iptables -A INPUT -i eth0 -s 127.0.0.0/8  -j DROP
iptables -A INPUT -i eth0 -s 169.254.0.0/16 -j DROP
iptables -A INPUT -i eth0 -s 0.0.0.0/8        -j DROP
iptables -A INPUT -i eth0 -s 239.255.255.0/24 -j DROP
```

# iptables: sekalaisia puolustuksia

- Fragmentit ovat nykyisin vain hyökkäyksiä

  iptables -A INPUT -f -j DROP

- NEW-paketit joissa ei ole SYN-bittiä

  iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

- ”Joulupaketit” (kaikki tcp-liput yhtaikaa päällä)

  iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

- Rikkinäiset NULL-paketit (ei mitään tcp-lippua päällä)

  iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

# iptables: limit

- Testillä --limit voidaan rajoittaa tietynlaisen liikenteen määrää:

  ... -m limit --limit *n/aikayksikkö* [--limit-burst *m*]

  Optio --limit rajoittaa pakettien määrän per aikayksikkö (second, minute, hour, day) ja --limit-burst kuinka monta pakettia saa ensin tulla peräkkäin ennen kuin niitä aletaan rajoittaa.

- Esim. sallitaan vain 5 ssh-yhteyttä peräkkäin ja sitten 3/minuutti:

  ... -p tcp --dport 22 -m limit --limit 3/minute --limit-burst 5 -j ACCEPT

# iptables: limit

- Esimerkki: estetään lokin täyttyminen, esim.

  -N LognDrop

  -A LognDrop -m limit --limit-burst 5 --limit 2/min -j LOG --log-prefix ”evil packet”

  -A LognDrop -j DROP

- Käytettävissä on myös --hashlimit, jolla voidaan tehdä monimutkaisempia rajoituksia tyyliin ”enintään 100 yhteydenottoa per portti-ja-palvelin” jne.

# iptables: connlimit

- connlimit -ehdolla voidaan rajoittaa yhtaikaisten yhteyksien määrää, esim.

  ... -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 --connlimit-mask 32 -j DROP

  rajoittaa porttiin 80 (http) samasta osoitteesta tulevien samanaikaisten yhteyksien määrän 20:een. Jos halutaan tehdä vastaava rajoitus yhteydenottajan aliverkolle, se voidaan tehdä --connlimit-mask -optiolla. Optio --syn rajaa testin uusiin yhteydenottoihin.

# iptables: recent

- recent -ehdolla voidaan virittää aikaisemmista tapahtumista riippuvia ehtoja. Esimerkiksi ssh-massahyökkäysten torjunta:

```
iptables -N SshTest # tehdään uusi ketju

# kutsutaan ketjua aina kun joku kolkuttaa ssh-porttia

iptables -A INPUT -i eth0 -p tcp -m tcp --dport 22 -m state --state NEW -j SshTest

# --set tallettaa tapahtuman joukkoon ”Ssh”

iptables -A SshTest -m recent --set --name Ssh --rsource
```

# iptables: recent

# jos yli 5 yritystä 10 minuutissa, lokitetaan ...

iptables -A SshTest -m recent --update --seconds 600 --hitcount 5 --name Ssh --rsource -j LOG --log-prefix "ssh-attack" --log-level warn

# ja blokataan ne

iptables -A SshTest -m recent --update --seconds 600 --hitcount 5 --name Ssh --rsource -j DROP

# muussa tapauksessa hyväksytään

iptables -A SshTest -j ACCEPT

# iptables: NFS

- NFS:n toiminta edellyttää isoa joukkoa erillisiä palveluita, jotka tarvitsevat omat porttinsa auki palomuurissa. Asiaa vaikeuttaa se, että osa niistä arpoo portin satunnaisesti, ellei sitä erikseen kiinnitä. Tarvittavat palvelut ovat (suluissa portti jos se on kiinteä): portmapper (111), nfsd (2049), lockd, mountd ja statd, sekä levykiintiöitä käytettäessä rquotad.

# iptables: NFS

- Muuttuvat portit voidaan kiinnittää näin:

  statd tiedostossa /etc/default/nfs-common:

  ```
  STATDOPTS="--port 4000 --outgoing-port 4001"
  ```

  mountd tiedostossa /etc/default/nfs-kernel-server:

  ```
  RPCMOUNTDOPTS="--manage-gids --port 4002"
  ```

  lockd tiedostossa (esim.) /etc/modprobe.d/lockd.conf:

  ```
  options lockd nlm_udpport=4003 nlm_tcpport=4003
  options nfs callback_tcpport=4004
  ```

  rquotad tiedostossa /etc/defaults/quota:

  ```
  RPCRQUOTADOPTS="-p 4005"
  ```

# iptables: NFS

- Portit pitää sitten avata palvelimen palomuurissa tyyliin

  iptables -A INPUT -s  ... -p tcp --dport 111 -j ACCEPT

  iptables -A INPUT -s  ... -p tcp --dport 2049 -j ACCEPT

  iptables -A INPUT -s  ... -p tcp --dport 4000:4005 -j ACCEPT

  ja vastaavat OUTPUT-säännöt asiakkaisiin ja mahdolliseen välissä olevaan palomuuriin FORWARDit.

- Valitut portit kannattaa lisätä tiedostoon /etc/services.

# Automounter (autofs)

- mounttaa nfs- tai muita verkkolevyjä tarpeen mukaan, ja umounttaa kun tarve poistuu
- asennus: apt-get install autofs
- mounttauksesta huolehtii demoni automount
- konfiguraatiotiedostot: /etc/auto.master ja siellä viitatut ns. kuvaustiedostot, map files (useimmiten /etc/auto.jotakin)

# Automounter (autofs)

- suorat kuvaukset (direct maps): absoluuttinen polku kuvaustiedostossa, harvoin käytetty
- epäsuorat kuvaukset (indirect maps): kuvaustiedostossa suhteellinen polku, yhdistetään master-tiedoston polkumäärityksen
- säätömahdollisuuksia monimutkaisempiin ympäristöihin (LDAP jne)

# autofs: esimerkki

- /etc/auto.master:

  /koti  /etc/auto.koti  --timeout=60

  /-  /etc/auto.suora  --timeout=180

- /etc/auto.koti:

  oma  kotiserver:/home

  naapuri tulppu:/home

# autofs: esimerkki

- /etc/auto.suora:

  /jako/peli peliserver:/pelijako

- Näkyvät hakemistot:

  /koti/oma /koti/naapuri /jako/peli

# autofs: yleistä

- automountatut hakemistot tulevat näkyviin kun niitä käytetään; edellisessä esimerkissä komento “ls /koti” voi näyttää tyhjää, mutta “cd /koti/oma/tt” toimii kuitenkin (ja sen jälkeen ls toimii myös)
- automounter luo tarvittavat välihakemistot (edellä /koti jne) itse, niitä ei saa tehdä valmiiksi
- master-tiedoston muuttamisen jälkeen tarvitaan

  service autofs reload

# autofs: yleistä

- umount tapahtuu automaattisesti määräajan (oletus 5min) kuluttua siitä kun hakemistoja on viimeksi käytetty (tarvittaessa fuser tai lsof tai ps auttavat selvittämään käyttäjän)
- oletusasetukset tiedostossa /etc/default/autofs

# autofs: wildcards, scripts

- kuvaustiedostossa voi käyttää hakemistonimessä jokerimerkkejä * ja &:
  - kotipalvelin:/home/&
- kuvaustiedosto voi olla skripti, joka tavalla tai toisella tuottaa halutun mounttikuvauksen (argumenttina levypalvelimen nimi); esimerkkinä /etc/auto.net, joka mounttaa mitä vain verkosta löytyy polkuun /net/kone/hakemisto

# autofs: debugging

- debuggausta varten automount-demonia voi ajaa verbose-optiolla etualalla omassa ikkunassaan:

```
service autofs stop
/usr/bin/automount -f -v
```