

NFS: Network File System

- Keino jakaa tiedostojärjestelmä kokonaan tai osittain (alihakemisto) toiselle koneelle.
- NFSv3:ssa *koneet* luottavat toisiinsa, vähänlaisesti tietoturvaominaisuuksia.

NFSv4 lisää paljon mm. käyttäjäkohtaisen autentikoinnin ja kaikenlaista muutakin.

Seuraavissa esimerkeissä NFSv3

NFS: Network File System

- **Palvelimella** (tunnus2): apt-get install nfs-kernel-server

/etc/exports:

/home tunnus1(rw,sync,no_subtree_check,root_squash)

exportfs -a

jos virhe "... not implemented":

service nfs-kernel-server restart

Ks. /etc/default/nfs-kernel-server

NFS: Network File System

- Yleisiä optioita /etc/exports'issa (ks. man 5 exports):

rw read+write (oletus readonly)

async nopea, vaarallinen jos palvelin kaatuu tms

subtree_check ekstratarkistuksia jos alihakemisto exportattu

[no_]root_squash root → nobody (oletuksena päällä)

all_squash kaikki → nobody (oletuksena pois)

NFS: Network File System

- **Asiakaskoneessa (tunnus1):**

```
apt-get install nfs-common
```

```
mkdir /home2; mount tunnus2:/home /home2
```

```
/etc/fstab:
```

```
tunnus2:/home /home2 nfs defaults 0 0
```

```
Ks. /etc/default/nfs-common
```

Palomureista

- *Palomuurilla* tarkoitetaan sekä
 - (1) Erillistä (fyysistä tai virtuaalista) konetta, joka välittää, suodattaa ja muokkaa kahden tai useamman muun koneen (tai verkon) välistä liikennettä ("rautapalomuuri") että
 - (2) Ohjelmaa tai ohjelmistoa, joka suodattaa ja muokkaa yhden koneen ja ulkomaailman välistä liikennettä ("softapalomuuri").

Palomureista

- Linux-ympäristöissä palomuurien alla on yleensä Netfilter-projekti (www.netfilter.org) ja erityisesti **iptables** (ja ip6tables).
- Korkeamman tason palomuuriratkaisuja, jotka tarjoavat helpomman käyttöliittymän iptables'in päälle, on paljon, esimerkiksi UFW (Ubuntun softapalomuuri) ja IpFire (Linux-pohjainen rautapalomuuriratkaisu).

iptables

- Linuxin (kernelin!) peruspalomuuriratkaisu IPv4-ympäristöön (IPv6:lle on vastaava ip6tables)
- Käsittelee *paketteja*: ei toimi korkeamman tason protokollien kanssa, ts. ei analysoi eikä yhdistele korkeammalla tasolla yhteen kuuluvia paketteja (ilman lisävirityksiä, joita toki on)
- Hyvin monipuolinen ja monimutkainenkin, kaikkien ominaisuuksien tehokas käyttö edellyttää TCP/IP:n ja siihen liittyvien protokollien syvällistä ymmärtämistä, perustilanteet kuitenkin kohtuuhelppoja

iptables

- Täydellinen (...) dokumentaatio <http://www.iptables.info>
- <http://rlworkman.net/howtos/iptables/iptables-tutorial.html>

hiukan vanhentunut mutta helppolukuisempi ja yhä olennaisesti oikea ja hyödyllinen kuvaus iptablesin toiminnasta

iptables: termejä

- drop/deny: paketti "pudotetaan lattialle", hävitetään eikä tehdä muuta
- reject: paketti hävitetään mutta lähettäjälle vastataan ja kerrotaan että näin kävi
- accept: paketti hyväksytään ja päästetään eteenpäin

iptables: termejä

- state: paketin tila suhteessa pakettivirtaan (new, established, related, invalid)
- chain (ketju): paketteihin sovellettava sääntöjoukko; sanotaan että paketti kulkee ketjun läpi (traverses through chain). Joukko vakioketjuja (INPUT jne), minkä lisäksi voi määritellä omia.
- table: taulu, johon palomuurisääntöjä talletetaan (raw, nat, mangle, filter)

iptables: termejä

- rule (sääntö): kokoelma ehtoja ja niiden toteutuessa suoritettava kohde
- match (ehto): säännössä ehto jonka pitää toteutua; käytetään myös koko säännöstä, jos kaikki sen ehdot toteutuvat
- target (kohde): säännössä kertoo mitä tehdään jos ehdot toteutuvat

iptables: termejä

- jump (hyppy): kohde, joka määrää hyppäämään toiseen ketjuun
- connection tracking: yhteyksien seuranta
- policy: oletustoimenpide sääntöjen loppuessa (drop, reject, accept)

Ketjut

- Vakioketjut ovat:

INPUT: paketit, joiden määränpää on (palomuuuri)kone itse

OUTPUT: koneesta ulos lähtevät (siellä syntyneet) paketit

PREROUTING: pakettien käsittely ennen reititystä
(päättöstä määränpäästä)

POSTROUTING: pakettien käsittely reitityksen (ja yleensä
suodatuspäättösten) jälkeen

FORWARD: paketit, joiden lähde ja määränpää
palomuurin ulkopuolella

Ketjut

- Kunkin ketjun sisällä käydään siihen kuuluvat taulut läpi järjestyksessä

raw → mangle → nat → filter

(kaikissa ketjuissa ei ole kaikkia tauluja)

- Mahdolliset omat ketjut toimivat ikäänkuin aliohjelmina muille ketjuille

Taulut

RAW: lähinnä haluttaessa ohittaa yhteysseuranta joillekin paketeille (kohde NOTRACK)

MANGLE: paketin ohjauskenttien muuttamiseen (kohteet TOS, TTL, MARK, SECMARK, CONNSECMARK)

FILTER: pakettien suodattamiseen, ts. hyväksymiseen tai hylkäämiseen niitä muuttamatta (kohteet DROP, REJECT, ACCEPT)

NAT: pakettien osoitekenttien (IP ja porttinumero) muuttamiseen (kohteet DNAT, SNAT, MASQUERADE, REDIRECT)

- Useimmissa ketjuissa ei ole kaikkia tauluja, ja useimmiten käytetään joka tapauksessa vain FILTER- ja NAT-tauluja

Kohteet

- Joukko sisäänrakennettuja vakiokohteita, yleisimmät:
 - ACCEPT: paketti hyväksytään, ketjun käsittely lopetetaan
 - DROP: paketti hylätään, ketjun käsittely lopetetaan
 - REJECT: paketti torjutaan, ketjun käsittely lopetetaan
 - LOG: paketti lokitetaan, ketjun käsittely jatkuu

Kohteet

- Vakiokohteita on enemmänkin, ks.

man 8 iptables-extensions

<http://www.iptables.info/en/iptables-targets-and-jumps.html>

– Vakiokohteiden kutsu aina *-j kohde*

Kohteet

- Omia ketjuja voi myös käyttää kohteina
 - Kaksi tapaa kutsua:
 - j *ketju*: palaa nykyiseen ketjuun kutsutun ketjun päätyttyä (vrt. aliohjelmakutsu)
 - g *ketju*: lopettaa nykyisen ketjun kutsutun päätyttyä ja palaa ylemmälle tasolle (vrt. goto)

Taulut ja ketjut

- Kolme perustilannetta:
 - Saapuva paketti (kohde palomuurikone itse; huom. tässä "palomuurikone" voi olla myös vain omia pakettejaan käsittelevä palvelin):
PREROUTING → INPUT
 - Lähtevä paketti (luotu palomuurikoneessa):
OUTPUT → POSTROUTING
 - Välitettävä paketti:
PREROUTING → FORWARD → POSTROUTING

Taulut ja ketjut

- Ketjujen käsittelyn välissä tehdään reitityspäätökset
- Kaikissa tilanteissa ei käytetä kaikkia tauluja, mutta niitä joita käytetään, käytetään aina samassa järjestyksessä

Saapuva paketti

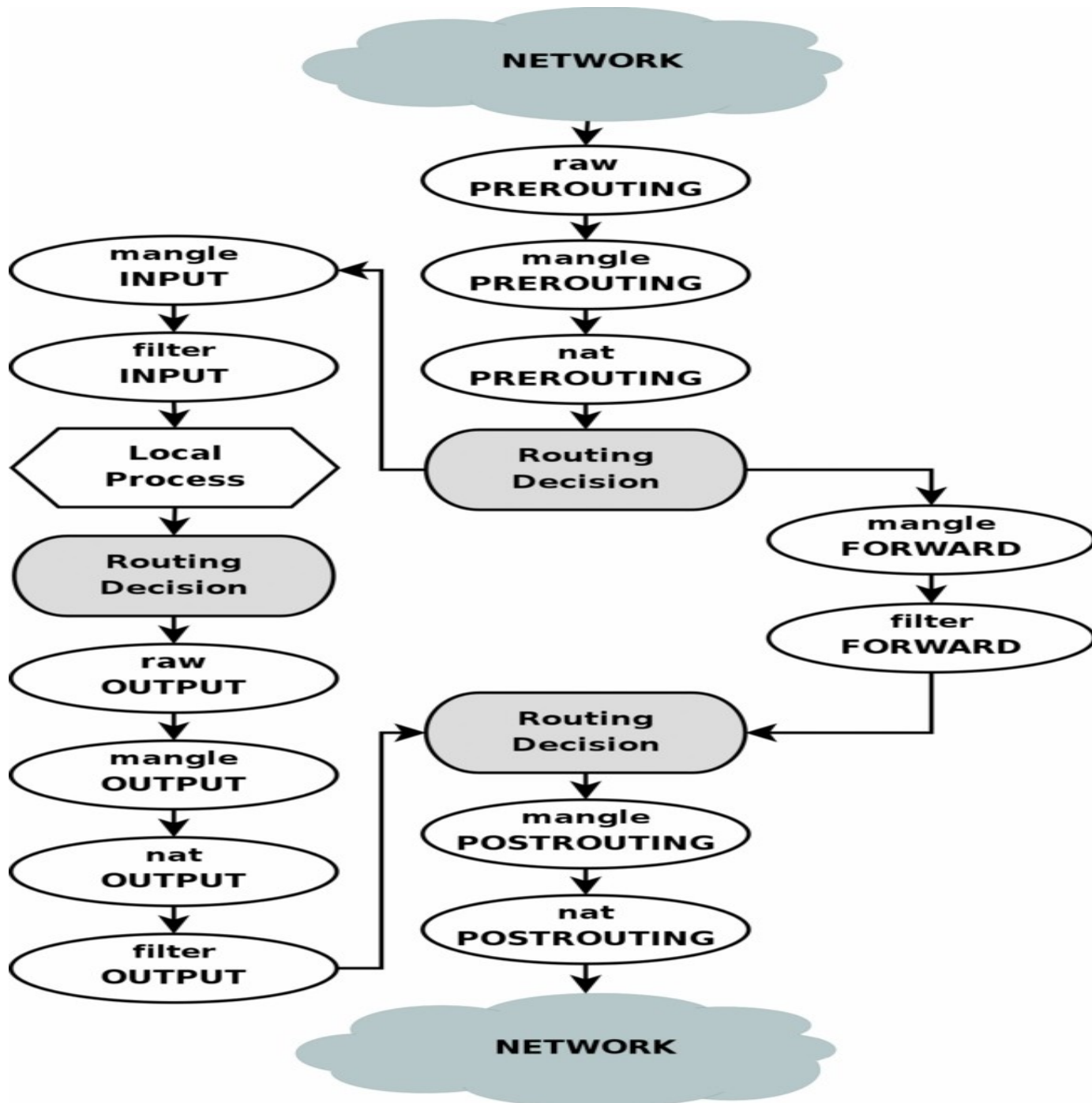
- Palomuurikoneelle osoitetun paketin käsittelyvaiheet:
 - PREROUTING - raw
 - yhteydenseurantakoodin käsittely
 - PREROUTING - mangle
 - PREROUTING - nat (DNAT)
 - reitityspäätös
 - INPUT - mangle
 - INPUT - filter

Lähtevä paketti

- Palomuurikoneelta lähtevän paketin käsittelyvaiheet:
 - reitityspäätös
 - OUTPUT - raw
 - yhteydenseurantakoodin käsittely
 - OUTPUT - mangle
 - OUTPUT - nat
 - OUTPUT - filter
 - reitityspäätös
 - POSTROUTING - mangle
 - POSTROUTING - nat (SNAT)

Välitettävä paketti

- Palomuurikoneen läpi kulkevan paketin käsittelyvaiheet:
 - PREROUTING - raw
 - yhteydenseurantakoodin käsittely
 - PREROUTING - mangle
 - PREROUTING - nat (DNAT)
 - reitityspäätös
 - FORWARD - mangle
 - FORWARD - filter
 - POSTROUTING - mangle
 - POSTROUTING - nat (SNAT)



iptables -komento

- Palomuuritauluja käsitellään iptables-komennolla:

`iptables [-t table] -A ketju sääntö # --append, lisäys ketjun loppuun`

`iptables [-t table] -I ketju [numero] sääntö # --insert, lisäys ketjun keskelle`

`iptables [-t table] -R ketju numero sääntö # --replace, säännön muutos`

`iptables [-t table] -D ketju {sääntö|numero} # --delete, säännön poisto`

iptables -komento

iptables [-t table] -L [ketju] # --list, sääntöjen tulostus

iptables [-t table] -S ketju [numero] # --list-rules, sääntöjen tulostus komentomuodossa

iptables [-t table] -F [ketju] # --flush, ketjun (tai kaikkien) tyhjennys

iptables [-t table] -Z [ketju [numero]] # --zero, laskurien nollaus

iptables [-t table] -N ketju # --new-chain, uuden ketjun luonti

iptables -komento

iptables [-t table] -X ketju # --delete-chain, tyhjän ketjun poisto

iptables [-t table] -P ketju kohde # --policy, oletuskohteen määrittäminen

iptables [-t table] -E ketju1 ketju2 # --rename-chain, ketjun nimen vaihto

- Oletustaulu (ilman -t -optiota) aina filter. Lisäksi yleisiä optioita:
-v, --verbose; -n, --numeric; -x, --exact; --line-numbers; --modprobe

iptables: sääntömäärittelyt

- Perussääntömäärittelyt (! = negaatio, "ei"):
 - [!] -p, --protocol {tcp|udp|udp|lite|icmp|esp|ah|sctp|all}
 - [!] -s, --source *address[/mask][...]*
 - [!] -d, --destination *address[/mask][...]*
 - j, --jump *kohde*
 - g, --goto *ketju*

iptables: sääntömäärittelyt

-g, --goto *ketju*

[!] -i, --interface *laite*

[!] -o, --out-interface *laite*

[!] -f, --fragment

-c, --set-counters *paketteja tavuja*

- Lisäksi on laajennuksia (extensions), erityisesti *match*

iptables: match extensions

- Erilaisia laajennuksia on tarjolla moduleina, jotka ladataan optiolla `-m` tai `--match` (jotkut latautuvat myös `-p:n` sivuvaikutuksena). Seuraavassa esitellään vain pieni valikoima.

```
-p tcp {--sport port[:port]|--dport port[:port]|--tcp-flags ...|  
--syn|--tcp-option n}
```

```
-p udp {--sport port[:port]|--dport port[:port]}
```

```
-p icmp --icmp-type type
```

```
-m state --state {NEW|ESTABLISHED|RELATED|INVALID}
```

iptables: match extensions

- m iprange {--src-range|--dst-range} *from*[-to]
- m connlimit [--connlimit_mask *n*] --connlimit-above *n*
- m limit --limit *rate*[/second|/minute|/hour|/day]
- m limit --limit-burst *n*
- m mac [!] --mac-source address

iptables: esimerkki 1

- Halutaan sallia kaikki yhteydet ulos ja sisään kaikki *paitsi* ei porttia 25 (smtp) mistään, ssh (22) vain JY:n (130.234.0.0/16) alueelta eikä mitään alueelta 134.170.0.0/16:

```
iptables -F
```

```
iptables -P INPUT ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -p tcp --dport 25 -j DROP
```

```
iptables -A INPUT -p tcp ! -s 130.234.0.0/16 --dport 22 -j DROP
```

```
iptables -A INPUT -s 123.170.0.0/16 -j DROP
```


iptables: esimerkki 1

- Tällainen "blacklisting" tyyli, että sallitaan kaikki mitä ei ole erikseen kielletty, ei yleensä ole hyvä idea palvelimeen sisääntulevan liikenteen rajoittamiseen. Rajapalomuurin FORWARD-säännöissä se voi joskus olla järkevä ratkaisu, jos esimerkiksi asiakkaille halutaan sallia oletuksena melkein kaikki (esim. kaikki paitsi smtp-portti 25, joka yleensä halutaan sulkea spämmerien takia), muulloin "whitelisting" (kiellettyä kaikki mitä ei erikseen sallita) on yleensä parempi.

Yhteyksien seuranta

- Connection tracking, "state machine"; iptables on "stateful firewall", pitää muistissa yhteyksien tiloja eikä vain käsittele jokaista pakettia uutena
- Tilaa voidaan säännöissä testata tyyliin
 - m state --state *state*

Yhteyksien seuranta

- Yhteyksillä voi olla neljä (viisi) tilaa:
 - NEW: yhteyden ensimmäinen paketti (ei kuulu mihinkään muistissa olevaan yhteyteen)
 - ESTABLISHED: paketti kuuluu muodostettuun (ja muistettuun) yhteyteen (liikennettä nähty molempiin suuntiin), vastaus aiemmin lähetettyyn pakettiin
 - RELATED: olemassaolevaan ESTABLISHED-yhteyteen liittyvä mutta ei siihen kuuluva paketti (aikaisemman yhteyden luoma uusi yhteys)
 - INVALID: tilaa ei ole tai sitä ei tunnisteta (koneesta muisti loppunut tai jotain muuta ikävää)
 - UNTRACKED: ei varsinaisesti tila vaan merkki, että tämän paketin tilaa ei seurata (pakettikohtainen, ei yhteyskohtainen)

iptables: esimerkki 2

- Sallitaan sisään http (portti 80) ja https (443) kaikkialta sekä ssh (22) alueelta 130.234.0.0/16, ei rajoiteta ulos lähtevää liikennettä:

```
iptables -F
```

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

```
iptables -A INPUT -p tcp -s 130.234.0.0/16 --dport 22 -j ACCEPT
```

```
iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable
```

iptables: esimerkki 2

- Paluuliikenne (ESTABLISHED) pitää sallia erikseen; senkin voisi rajata haluttuihin portteihin.
- Tässä siis käytetään vain filter-taulua (joka on oletus).
- Viimeinen REJECT ei ole välttämätön, antaa vain siistin virheen.
- Huom. mm. ftp ulospäin ei toimi näillä säännöillä, koska sen paluuliikennettä ei tunnisteta ESTABLISHED-liikenteeksi.

iptables: oikoteitä

- Osoitteet voivat olla yksittäisiä tai maskilla rajattuja aliverkkoja
- Useita osoitteita voi antaa pilkulla erottaen (130.234.0.0/16,172.20.208.17) - iptables jakaa tällaiset sisäisesti kahtia (ja iptables -L jne tulostavat ne kahtena)
- Modulilla iprange voidaan antaa osoitevälejä tyyliin
... -m iprange --src-range 172.20.208.10-172.20.209.120

iptables: oikoteitä

- Yksittäisen portin lisäksi voidaan antaa kaksoispisteellä erotettu väli, esim. 49152:59999; jos välin toisen pään jättää pois, se tarkoittaa kaikkia siitä eteenpäin, esim. 1024: on portit 1024-65535
- Normaalisti samalla kertaa ei voi antaa useita ei-peräkkäisiä portteja, mutta tarkoitusta varten on lisämoduli multiport:
... -m multiport --dports 80,443 -j ACCEPT
- Porttinumeroiden asemesta voi usein käyttää palveluiden lyhenteitä (ne katsotaan /etc/services -tiedostosta), mutta se hidastaa sääntöjen latausta joskus merkittävästikin

iptables: esimerkki 3

- Kuten esimerkki 1, mutta sallitaan ssh myös 172.20.0.0/16-alueelta ja ulospäin vain dns (portti 53) ja apt-proxy (3142):

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```


iptables: esimerkki 3

```
iptables -A INPUT -p tcp -s 130.234.0.0/16,172.20.0.0/16 --dport 22 -j ACCEPT
```

```
iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable
```

```
iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp -d 172.20.208.17 --dport 3142 -j ACCEPT
```

- Ilman nimipalvelua juuri mikään ei toimi; nykyisin syytä sallia sekä udp- että tcp-portit.
- Myös OUTPUT-ketjussa pitää nyt sallia paluuliikenne erikseen.

iptables: icmp

- ICMP-paketeista yleensä halutaan sallia ainakin 3 (destination-unreachable) ja 11 (time-exceeded), usein myös 12 (parameter-problem) ja 13 (timestamp-request, mm. openvpn käyttää sitä):

```
iptables -A INPUT -p icmp -m icmp --icmp-type 3 -j ACCEPT
```

```
iptables -A OUTPUT -p icmp -m icmp --icmp-type destination-unreachable -j ACCEPT
```

...

- Ulospäin ICMP:n voi yleensä sallia vapaamminkin (vaikka kaikki pakettityypit)

iptables: icmp

- Erityisesti icmp echo (8 echo-request, 0 echo-reply) eli ping halutaan usein sallia ainakin ulospäin, sisäänpäin yleensä vain rajoitetusti.
 - Ping ulospäin:
iptables -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
lisäksi tarvitaan joko em. INPUT... ESTABLISHED-sääntö tai
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
 - Ping sisäänpäin vastaavasti
iptables -A INPUT -s 172.20.0.0/16 -p icmp --icmp-type echo-request -j ACCEPT
jälleen lisäksi joko OUTPUT...ESTABLISHED tai
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

iptables: sääntöjen talletus

- Säännöstön voi tallettaa komennolla
iptables-save >tiedosto
ja palauttaa komennolla
iptables-restore <tiedosto
 - iptables-save on joskus kätevä myös ilman uudelleensuuntausta sääntöjen tutkimiseen.
- Voi myös kirjoittaa iptables-komennot skriptiin

iptables: sääntöjen talletus

- Jotta säännöt saa pysyviksi bootin yli, ne pitää ajaa päälle jossain käynnistysskriptissä. Vaihtoehtoja on monta, yksi luonteva paikka on `/etc/network/interfaces` -tiedostossa `pre-up` -säännöllä (suoritetaan ennen verkkoyhteyden käynnistämistä), esim. jompi kumpi näistä:

```
pre-up iptables-restore < /etc/network/iptables.up.rules
```

```
pre-up /etc/network/iptables.up.run
```

iptables: sääntöjen testaus

- Uusia sääntöjä interaktiivisesti testatessa kätevä on

```
iptables-apply [-t seconds] [tiedosto|-c [skripti]]
```

joka kysyy onnistuiko ja palauttaa vanhat säännöt ellei saa vastausta määräajassa.

Tiedoston oletusarvo on `/etc/network/iptables.up.rules`, syntaksi sama kuin `iptables-restore`'lla, oletusskripti taas on `/etc/network/iptables.up.run` (se voi suorittaa mitä tahansa komentoja, katsoa koneen oman osoitteen `/etc/hosts`'ista, tehdä sääntöjä monelle koneelle silmukassa jne; huom. verkko ei aina ole päällä sitä suoritettaessa).

iptables: lokitus

- Yksi erityinen kohde iptables-säännöille on **LOG**, jolla haluttu (epäilyttävä) paketti saadaan kirjattua lokiin. Lokitusta voidaan tarkemmin säädellä optioilla:

--log-level *level* # syslog-tasot, debug, notice, info, warn, err jne

--log-prefix *string* # merkkijono lokiviestien alkuun

--log-tcp-sequence # kirjataan tcp-pakettien numerot lokiin

--log-tcp-options # tcp-optiot lokiin

--log-ip-options # ip-optiot lokiin

iptables: lokitus

- Esimerkki:

```
iptables -A ... -j LOG --log-level debug --log-prefix "nasty packet"
```

```
iptables -A ... -j DROP
```

- `syslog-facility` on "kern", sitä ei voi vaihtaa. Ts. `rsyslog.conf`'issa tarvitaan esim.

```
kern.debug /var/log/kernel.log
```

- Käytettävissä on myös ULOG-kohde, joka tarjoaa monipuolisempia lokitusmahdollisuuksia. Sen käyttää omaa protokollansa ja sen käyttö edellyttää sitä varta vasten kuuntelevaa ohjelmaa (ulogd tms).

iptables: ftp

- Ftp on palomureille vaikea protokolla, koska siinä on erotettu ohjaus ja datansiirto eri portteihin ja porttineuvottelut käydään palomuurin näkökulmasta siirrettävän datan sisällä. Sillä on lisäksi kaksi toimintatapaa, aktiivinen ja passiivinen:
 - Aktiivinen ftp:
 - Asiakas ottaa (ohjaus)yhteyden ftp-palvelimen porttiin 21.
 - Palvelin kertoo asiakkaalle mistä (satunnaisesta) portista datayhteys tulee.
 - Palvelin ottaa (data)yhteyden asiakkaan porttiin 20.
 - Data siirtyy portin 20 kautta edes ja takaisin, ohjauskomennot portin 21 kautta.

iptables: ftp

- Passiivinen ftp:
 - Asiakas ottaa (ohjaus)yhteyden ftp-palvelimen porttiin 21.
 - Palvelin ilmoittaa asiakkaalle (valitsemansa satunnaisen) portin datayhteyttä varten.
 - Asiakas ottaa yhteyden palvelimen ilmoittamaan porttiin.
 - Data siirtyy edellä valitun portin kautta, ohjauskomennot taas portin 21 kautta.
- Tämä edellyttää palomuurilta yhteyden seuranta ja ftp-protokollan tuntevaa modulia (`nf_conntrack_ftp`), joka lukee ohjausyhteyden pakettien sisältöjä selvittääkseen mikä datayhteys siihen liittyy.

iptables: ftp client

- Ftp-asiakasohjelmaa varten tarvitaan palomuriin seuraavanlaiset säännöt:

```
modprobe nf_conntrack_ftp
```

```
# Yhteyden muodostus: sallitaan yhteydenotto porttiin 21  
maailmalla
```

```
iptables -A OUTPUT -p tcp --dport 21 -m state --state  
NEW,ESTABLISHED -j ACCEPT
```

```
# sallitaan vastaus (ei tarpeen jos vastaukset yleisemminkin  
sallittu)
```

```
iptables -A INPUT -p tcp --sport 21 -m state --state  
ESTABLISHED -j ACCEPT
```

iptables: ftp client

Aktiivinen ftp: annetaan palvelimen ottaa yhteys takaisin porttiimme 20

```
iptables -A INPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

ja sallitaan itsemme vastata sille:

```
iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
```

iptables: ftp client

Passiivinen ftp: sallitaan itsemme ottaa yhteys palvelimen ilmoittamaan porttiin

```
iptables -A OUTPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED,RELATED -j ACCEPT
```

ja sallitaan palvelimen vastata

```
iptables -A INPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED -j ACCEPT
```

iptables: ftp server

- Ftp-palvelin vaatii olennaisesti vastaavat säännöt kuin asiakaskin, mutta peilikuvina (vertaa input/output ja dport/sport edelliseen):

```
modprobe nf_conntrack_ftp
```

```
# Yhteyden muodostus: sallitaan yhteydenotto omaan porttiimme  
21 maailmalta
```

```
iptables -A INPUT -p tcp --dport 21 -m state --state  
NEW,ESTABLISHED -j ACCEPT
```

```
# sallitaan itsemme vastata
```

```
iptables -A OUTPUT -p tcp --sport 21 -m state --state  
ESTABLISHED -j ACCEPT
```

iptables: ftp server

- # Aktiivinen ftp: sallitaan itsemme ottaa paluuyhteys portin 20 kautta

```
iptables -A OUTPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

ja sallitaan vastaukset siihen

```
iptables -A INPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
```

iptables: ftp server

Passiivinen ftp: sallitaan yhteydenotto maailmalta uuteen omaan porttiimme

```
iptables -A INPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED,RELATED -j ACCEPT
```

ja sallitaan itsemme vastata

```
iptables -A OUTPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED -j ACCEPT
```

- Passiivisen yhteyden testaus: ftp -p ...

iptables: omat ketjut

- Jos samat säännöt toistuvat tai samaa ehtoa käytetään isolle joukolle sääntöjä, ne voi koota omaksi ketjukseen ja kutsua sitä tarvittaessa.
- Esimerkiksi lokitus yhdistettynä hylkäykseen:

```
iptables -N LOGDROP
```

```
iptables -A LOGDROP -j LOG --log-prefix "bad packet"
```

```
iptables -A LOGDROP -j DROP
```

jota sitten käytetään tähän tapaan:

```
iptables -A INPUT -s 134.170.0.0/16 -j LOGDROP
```

iptables: omat ketjut

- Ellei oma ketju tee jotain joka päättää paketin käsittelyn (kuten DROP tai ACCEPT) sen loputtua käsittely jatkuu siellä mistä sitä kutsuttiin jos käytettiin -j:tä, tai käsittely päättyy kuten kutsuva ketju olisi loppunut jos sen sijaan käytettiin -g:tä. Ylläolevassa esimerkissä näillä ei olisi eroa, koska ehdoton DROP päättää käsittelyn kuitenkin. Ketju voidaan myös keskeyttää eksplisiittisesti kohteella RETURN.

iptables: FORWARD

- Jos palomuurikone jo reitittää liikennettä normaalisti, sen suodattaminen on helppoa, esim.

```
iptables -P FORWARD DROP
```

```
iptables -A FORWARD -d 192.168.122.0/24 -s 130.234.0.0/16 -p tcp  
--dport 22 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.122.19 -p tcp --dport 80 -j ACCEPT
```

```
iptables -A FORWARD -d 192.168.122.19 -p tcp --dport 443 -j  
ACCEPT
```

```
iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.122.0/24 -p udp --dport 53 ACCEPT
```

```
iptables -A FORWARD -s 192.168.122.0/24 -p tcp --dport 53 ACCEPT
```

iptables: FORWARD

- Edellä ESTABLISHED -sääntö toimii tarkoituksella molempiin suuntiin, vaikka ulospäin ei pääsisikään kuin DNS.
- Tässä siis reititys hoidetaan normaaliin tapaan reititystaululla eikä osoitteita eikä portteja muuteta, päätetään vain mitä välitetään eteenpäin ja mitä ei.
- Kaikki säännöt ovat forward-ketjussa ja filter-tilussa.
- Ftp ja muut vastaavat vaatisivat tässäkin omat lisäsäätönsä.

NAT: Network Address Translation

- Muuttaa paketin ip:n ja/tai portin mennessä tullen.
- Alunperin viritys IP-osoitteiden säästämiseen, rikkoo joitakin protokollia; käynee harvinaisemmaksi IPv6:n yleistymisen myötä.
- Yleinen virtuaalikoneiden alustakoneen ja sen VM:ien välissä.
- SNAT, DNAT, PNAT

SNAT, DNAT, PNAT

- SNAT (Source NAT), masquerade: vaihdetaan lähdeosoite ts. IP, josta paketti on tulossa (source address), yleensä privaatti-IP:stä julkiseksi, mahdollisesti useita samaksi (silloin vaihdetaan myös porttinumero); palomuurin sisäpuolelta ulos lähtevälle liikenteelle.
- DNAT (Destination NAT): vaihdetaan paketin kohdeosoite, esim. ohjataan palomuurin ulkopuolelta samaan osoitteeseen tulevia paketteja eri koneisiin palomuurin sisäpuolella (yleensä) porttinumeron perusteella.
- PNAT (Port NAT): yleinen termi edellisille silloin kun muutetaan osoite-portti -yhdistelmää.
- many-to-one NAT (erityisesti PNAT) vs. one-to-one NAT

iptables: SNAT

- SNAT (Source NAT) tarkoittaa lähtevän (välitettävän) paketin lähtöosoitteen (source address) vaihtamista; usein myös portti vaihdetaan. Yleisin käyttö tälle on palomuurin sisältä tulevan liikenteen yksityisten osoitteiden muuttaminen palomuurin ulkopuolen julkiseksi osoitteeksi vaihtaen samalla lähtöporttia vastausten ohjaamiseksi takaisin. Mahdollista on myös vain vaihtaa yksityinen osoite julkiseksi porttia muuttamatta (one-to-one NAT). Jos palomuurilla on useita julkisia osoitteita, niitä voidaan myös käyttää satunnaisesti tms.

iptables: SNAT many-to-one

- Lähtevälle paketille vaihdetaan IP:ksi palomuurin julkinen IP ja portiksi jokin (satunnainen vapaa) portti ja muistetaan se, paluupaketti ohjataan takaisin tallennetun porttinumero-IP -parin perusteella (connection tracking)
- Tyypitilanne: työasemia palomuurin takana yksityisillä osoitteilla, palomuurilla vain yksi julkinen osoite.
- SNAT tehdään POSTROUTING-ketjussa ja NAT-taulussa. Kohde voi olla SNAT tai MASQUERADE (tai joskus SAME tai NETMAP).

SNAT variantteja

- SNATin kanssa voi käyttää useita erilaisia kohteita:

... -j SNAT --to-source *addr[-addr2][:port1-port2]*

- *addr* määrää mitä lähtöosoitetta (yleensä jokin palomuurin julkisista osoitteista) käytetään (valitaan satunnaisesti jos useita)

... -j MASQUERADE [--to-ports *port1-port2*]

- lähtöosoite määräytyy automaattisesti, unohtaa vanhat yhteydet verkkoyhteyden katketessa (hyvä jos IP on dynaaminen), hieman tehottomampi kuin SNAT

SNAT variantteja

... -j SAME --to *addr1-addr2* [--nodst]

- kuten SNAT monella osoitteella, mutta muistaa mitä lähtöosoitetta milläkin koneella on käytetty ja yrittää käyttää samaa kun kohdekin on sama (tai aina --nodst -optiolla)

... -j NETMAP --to *addr/mask*

- kohdeosoitteena aina verkkoalue, joka on samankokoinen kuin lähtöalue (-s...), tekee 1-1 NATin pitäen osoitteen lopun samana. Esim.

... -s 172.20.209.0/24 ... -j NETMAP 130.234.209.0/24

SNAT: esimerkki

- Liikenne lonka7:n sisäverkosta 192.168.127.0/24 olevista koneista (tunnus4 -koneet) ohjautuu ulos seuraavilla iptables-säännöillä (iptables -S -formaatti):
 - A POSTROUTING -s 192.168.127.0/24 ! -d 192.168.127.0/24 -p tcp -j MASQUERADE --to-ports 1024-65535
 - A POSTROUTING -s 192.168.127.0/24 ! -d 192.168.127.0/24 -p udp -j MASQUERADE --to-ports 1024-65535
 - A POSTROUTING -s 192.168.127.0/24 ! -d 192.168.127.0/24 -j MASQUERADE
 - A FORWARD -d 192.168.127.0/24 -o virbr0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
 - A FORWARD -s 192.168.122.0/24 -i virbr0 -j ACCEPT
- Uudelleenohjausta ei tehdä sisäverkon sisällä (! -d ...)

SNAT: esimerkki

- Ohjaus tehdään erikseen tcp:lle ja udp:lle, joille halutaan määrätä portit (icmp ja muut eivät porttinumeroita käytä, joten niille ei voi käyttää --to-ports -optiota).
- --ctstate on conntrack -modulin versio --state -testistä (tässä toiminta sama, mutta conntrack sisältää paljon enemmän toiminnallisuutta ja on joskus tehokkaampi).
- Tämä on olennaisesti libvirtd:n oletusverkon NAT-säännöstö (siellä on lisäksi joitakin suodatussääntöjä).

iptables: DNAT

- DNAT (Destination NAT) tarkoittaa, että saapuvan (välitettävän) paketin kohdeosoite (destination address) vaihdetaan; myös portti voidaan vaihtaa. Yleisin käyttö tälle on ohjata samaan julkiseen osoitteeseen tulevat portit eri koneisiin palomuurin sisällä. Joskus myös vain vaihdetaan julkinen IP yksityiseksi (one-to-one NAT) porttiin puuttumatta, ja kohdekone palomuurin sisällä voidaan valita muullakin kuin portin perusteella (vaikkapa lähdeosoitteella).
- Yleinen erikoistapaus: virtuaalikoneiden alustakone ohjaa liikennettä valikoiden sisällään eri virtuaalikoneille.

iptables: DNAT

- Voidaan käyttää myös transparent proxy-viritykseen toiseen suuntaan: esim. http- tai smtp-yhteys palomuurin sisältä ulos ohjataankin omaan välityspalvelimeen (tai sensuurin kynsiin...)
- DNAT tehdään PREROUTING-ketjussa ja NAT-taulussa, kohde DNAT (tai NETMAP).
- Jos DNATin tekee koneessa, joka ei muuten toimi reitittimenä ao. koneiden välissä, sen kanssa yleensä tarvitaan myös SNAT paluuliikennettä varten.

DNAT: esimerkki

- Ohjataan tt2-koneen (172.20.209.119) portti 80 koneeseen tt1 (172.20.209.19) palomuurisäännöillä tt2:ssa:

```
# ohjataan porttiin 80 saapuva liikenne tt1:een:
```

```
iptables -t nat -A PREROUTING -d 172.20.209.119 -p tcp -m tcp --dport 80 -j  
DNAT --to-dest 172.20.209.19
```

```
# ohjataan paluuliikenne takaisin SNATilla:
```

```
iptables -t nat -A POSTROUTING -s 172.20.209.19 -p tcp -m tcp --dport 80 -j  
MASQUERADE
```

```
# sallitaan edelleenohjaus (tarpeen kun forward policy on drop):
```

```
iptables -A FORWARD -d 172.20.209.19 -p tcp -m tcp --dport 80 -j ACCEPT
```

DNAT: esimerkki

sallitaan paluuliikenteen uudelleenohjaus:

```
iptables -A FORWARD -s 172.20.209.19 -p tcp -m state --state ESTABLISHED -j ACCEPT
```

ohjataan ja sallitaan tt2:sta itsestään lähtevä liikenne samaan tapaan:

```
iptables -t nat -A OUTPUT -d 172.20.209.119 -p tcp -m tcp --dport 80 -j DNAT --to-dest 172.20.209.19
```

```
iptables -A OUTPUT -s 172.20.209.119 -d 172.20.209.19 -p tcp --dport 80 -j ACCEPT
```