Palomuureista

- *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”).

Palomuureista

- Linux-ympäristöissä palomuurien alla on yleensä Netfilter-projekti (www.netfilter.org) ja erityisesti **iptables** ja ip6tables. Lisäksi on erityistarpeisiin ebtables ja arptables. Tulollaan on kaikki mainitut yhdistävä *nftables*.
- Korkeamman tason palomuuriratkaisuja, jotka tarjoavat helpomman käyttöliittymän iptablesin päälle, on paljon, esimerkiksi ufw (Ubuntun softapalomuuri, "uncomplicated firewall"), ja kokonaisia Linux-jakeluitakin rautapalomuurien tekoon, esim. IpFire.

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 erillisiä lisävirityksiä, joita toki on paljon.
- Hyvin monipuolinen ja monimutkainenkin, kaikkien ominaisuuksien tehokas käyttö edellyttää IP-protokollien (TCP/IP ,UDP, ICMP...) 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 aliketjuja.
- 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 (palomuuri)kone itse

  OUTPUT: koneesta ulos lähtevät (siellä syntyneet) paketit

  PREROUTING: pakettien käsittely ennen reititystä (päätöstä määränpäästä)

  POSTROUTING: pakettien käsittely reitityksen (ja yleensä suodatuspäätösten) 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: haluttaessa ohittaa tai muokata yhteysseurantaa joillekin paketeille (kohteet NOTRACK, CT)

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, käsittely lopetetaan

  DROP: paketti hylätään, käsittely lopetetaan

  REJECT: paketti torjutaan, käsittely lopetetaan

  LOG: paketti lokitetaan (käsittely jatkuu)

  RETURN: palataan kutsuneeseen ketjuun

  DNAT, SNAT, MASQUERADE, REDIRECT: pakettien uudelleenohjaus

  CT: yhteyden seuranta helper-modulien ohjaus

- Lisää ks. man 8 iptables-extensions

Kohteet

- Vakiokohteiden kutsu on aina muotoa

  -j kohde [optiot]

- Omia ketjuja voi myös käyttää kohteina
  - Kaksi tapaa kutsua:

    -j *ketju:* palaa nykyiseen ketjuun kutsutun ketjun päätyttyä tai kun aliketjussa hypätään kohteeseen RETURN (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 (reitittimenä toimiva palomuurikone):

    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

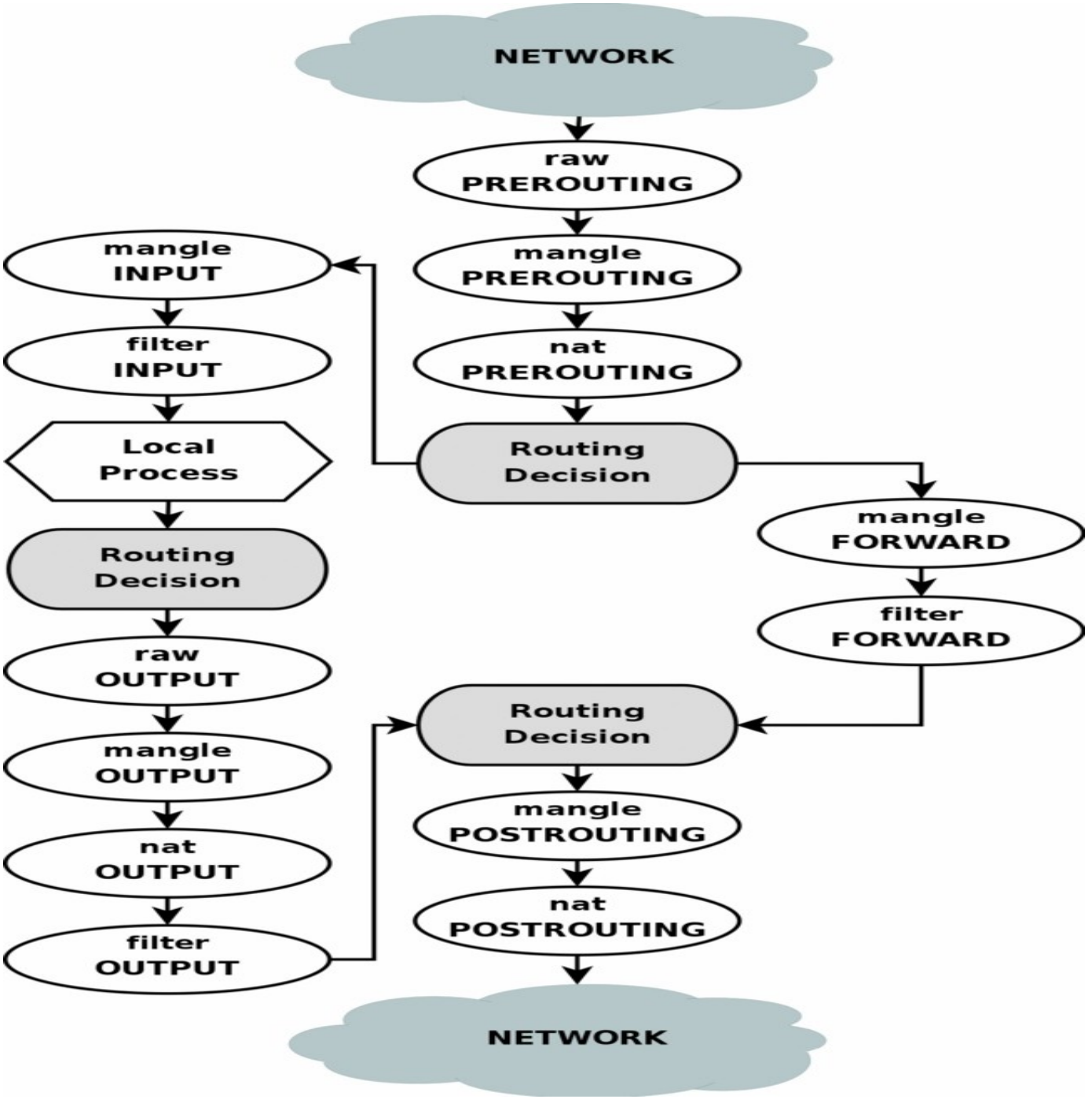
- Palomuurikoneelle osoitetun paketin käsittelyvaiheet:
  - PREROUTING - raw
  - yhteydenseurantakoodin käsittely
  - PREROUTING - mangle
  - PREROUTING - nat (DNAT)
  - reitityspäätös
  - INPUT - mangle
  - INPUT - filter

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)


NETWORK
raw
PREROUTING
mangle
PREROUTING
nat
PREROUTING
Routing
Decision
mangle
INPUT
filter
INPUT
Local
Process
Routing
Decision
raw
OUTPUT
mangle
OUTPUT
nat
OUTPUT
filter
OUTPUT
mangle
FORWARD
filter
FORWARD
Routing
Decision
mangle
POSTROUTING
nat
POSTROUTING
NETWORK

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

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

iptables -komento

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

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

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: policy

- *Policy* on oletustoiminto, joka tehdään ketjun lopussa ellei mikään sääntö muuta määrää
- Vain INPUT, OUTPUT ja FORWARD-ketjuille
- Vaihtoehtoja on vain kaksi: ACCEPT (blacklisting) ja DROP (whitelisting)
- Asetetaan komennolla iptables -P *ketju*
- Esim.

  iptables -P INPUT DROP

  iptables -P OUTPUT ACCEPT

  iptables -P FORWARD DROP

iptables: sääntömäärittelyt

- Perussääntömäärittelyt (! = negaatio, ”ei”):

  [!] -p, --protocol {tcp|udp|udplite|icmp|esp|ah|sctp|all}

  [!] -s, --source *address[/mask][...]*

  [!] -d, --destination *address[/mask][...]*

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

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

  [!] -f, --fragment

  -c, --set-counters *paketteja tavuja*
- Lisäksi on laajennuksia (extensions), erityisesti *match*
- Lopussa aina -j tai -g

iptables: match extensions

- Erilaisia laajennuksia on tarjolla moduleina, jotka ladataan optiolla -m tai --match, jotkut latautuvat myös -p:n sivuvaikutuksena. Esimerkkejä:

  -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}

  -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 134.170.0.0/16 -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. Näin voidaan erityisesti käsitellä paluuliikenne (vastaukset lähteneisiin paketteihin) eri tavalla kuin uudet.
- Tilaa voidaan säännöissä testata tyyliin

  -m state --state *state # tai*

  -m conntrack --ctstate *state*

  (olennaisesti samoja, jälkimmäinen uudempi, enemmän ominaisuuksia erikoistarpeisiin)

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 (koska policy on DROP), 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, vastaavasti :1023 on sama kuin 1-1023
- 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 (esim. /etc/network/iptables.up.run)

iptables: ifupdown

- Jotta säännöt saa pysyviksi bootin yli, ne pitää ajaa päälle jossain käynnistysskriptissä.
- Jos käytössä on ifupdown, luonteva paikka on /etc/network/interfaces -tiedostossa pre-up -säännöllä (suoritetaan ennen verkkoyhteyden käynnistämistä), esim. jompi kumpi näistä (jälkimmäinen siis jos olet kirjoittanut skriptin sääntöjen lataamista varten):

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

  pre-up /etc/network/iptables.up.run
- Joissakin erikoistilanteissa up on parempi kuin pre-up

iptables: netplan

- Netplan ei suoraan tue pre-up -tyyppisiä kutsuja, mutta sen (yleensä) käyttämä *renderer* networkd tarjoaa keinon: tehdään skripti /etc/networkd-dispatcher/routable.d tai configured.d -hakemistoon, esim. 40-iptables-up:

  ```
  #! /bin/sh

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

- networkd -skriptit suoritetaan asynkronisesti eikä niiden ajoitus vastaa yksi yhteen ifupdown'in pre-up -jne tiloja. Karkeasti:
  - pre-up, up: configuring.d, configured.d
  - up, post-up: routable.d
  - down, post-down: off.d, no-carrier.d

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. Mutta rsyslog.conf'issa voi ohjata viestejä eri tiedostoihin tyyliin

  kern.debug /var/log/kernel.log

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

iptables: lokitus

- Sääntöjä testatessa on usein kätevää seurata lokista tiettyä IP:tä:

  tail -f /var/log/kern.log | grep --line-buffered 130.234

  - tail -f lukee lokia jatkuvasti
  - grep --line-buffered tulostaa joka rivin saman tien eikä puskuroi niitä muistiin
  - tulostusta voi myös tiivistää, esim.

    tail -f /var/log/kern.log | sed -n '/130.234/ {s/IN=.*SRC=/SRC=/;s/LEN=.*PROTO=TCP//;s/ WINDOW=.*//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-taulussa.
- Ftp ja muut vastaavat vaatisivat tässäkin omat lisäsäätönsä.