Kernel-moduleista

- Joskus pitää eksplisiittisesti ladata tai poistaa kernelin moduleita (dynaamisia kirjastoja):
  - modprobe lataa tai poistaa (-r) modulin "älykkäästi" (riippuvuuksineen jne)
  - insmod lataa modulin minimitarkistuksin (modprobe yleensä parempi)
  - rmmod poistaa modulin (modprobe -r yleensä parempi)
  - lsmod tulostaa käytössä olevat modulit
  - modinfo tulostaa tietoja yksittäisestä modulista
- Tiedostossa /etc/modules on luettelo moduleista, jotka halutaan ladattavaksi bootissa (sen lisäksi mitä initramfs tuo).
- Hakemistoon /etc/modprobe.d/ voi laittaa modulikohtaisia sääntöjä modprobe-komentoa varten (yleensä tulevat pakettien mukana).

Palomuurin tyhjennys

- iptables ei tarjoa mitään helppoa keinoa kaikkien sääntöjen poistamiseen kerralla vaan kaikkien taulujen policyt, säännöt ja omat ketjut pitää poistaa erikseen. Yksinkertaisessa tilanteessa (vain filter-taulua käytetty, ei omia ketjuja) riittää:

  iptables -F

  iptables -P INPUT ACCEPT

  iptables -P OUTPUT ACCEPT

  iptables -P FORWARD ACCEPT

Palomuurin tyhjennys

- Perusteellisesti iptables'in saa tyhjäksi näin:

  iptables -P INPUT ACCEPT

  iptables -P OUTPUT ACCEPT

  iptables -P FORWARD ACCEPT

  iptables -F

  iptables -X

  iptables -t nat -F

  iptables -t nat -X

  iptables -t mangle -F

  iptables -t mangle -X

  iptables -t raw -F

  iptables -t raw -X

Palomuurin tyhjennys

- Tyhjennyksen voi tehdä näinkin (yrittää turhaan -P:tä silloinkin kun se ei ole mahdollinen mutta ohittaa virheet):

```
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
```

- Jos ip6tables on käytössä se pitää tyhjentää erikseen (samoin ebtables ja arptables)

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 hävittää pysyvästi, senkin voi tehdä:

  apt-get purge iptables

  Tätä ei yleensä tehdä kuin embedded-ympäristöissä tai jos tilalle asennetaan nftables tms.

iptables: localhost, kerberos, ldap

- 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, tai SSL:n kanssa 636:

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

  iptables -A OUTPUT -p tcp --dport 636 -j ACCEPT

- Paluuliikenne pitää aina sallia erikseen ellei sitä ole yleisemmin sallittu.

- Jos jokin palvelu ei toimi, sen porttinumeroita voi etsiä tiedostosta /etc/services (siellä voi olla lokaaleja lisäyksiä, joista Googlekaan ei tiedä), tai etsimällä blokattuja portteja palomuurin lokista.

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 172.21.208.16  # etsi avoimet portit lonka6:sta

nmap -p80,443 172.21.209.0/24

# 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: 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 …); tosin -e saattaa aiheuttaa ongelmiakin.
- Uudet säännöstöt kannattaa testata ensin **iptables-apply** -komennolla, etenkin jos konetta säätää ssh:n tms verkkoyhteyden yli. (Jos konsoliyhteydenkin saa, se kannattaa silti varmuuden vuoksi avata myös.)

iptables: debugging

- Epäilyttävien (testausvaiheessa kaikkienkin) DROP- ja REJECT-sääntöjen eteen voi laittaa LOG-säännön samalla ehdolla ja katsoa mitä lokissa näkyy. Myös läpi päästettyjä paketteja voi lokittaa (kunhan varoo lokin täyttymistä).
- Lokia voi seurata "reaaliajassa" tail -f -komennolla toisessa ikkunassa.
- Palomuurin toimintaa voi testata mm. komennoilla telnet (tcp-portit), ping (icmp), nmap (kaikki...) ja traceroute (etenkin useamman palomuurin läpi mentäessä).

NAT: Network Address Translation

- Muuttaa paketin ip:n ja usein myös portin mennen tullen.
- Alunperin viritys IP-osoitteiden säästämiseen, rikkoo joitakin protokollia; käymässä harvinaisemmaksi IPv6:n yleistymisen myötä.
- Yleinen virtuaalikoneiden alustakoneen ja sen VM:ien välissä.
- SNAT, DNAT, PNAT, 1-to-1 NAT

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 (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 satunnaistamalla, round-robin -tekniikalla 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).
- Tyyppitilanne: 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 käytetään (yleensä jokin palomuurin julkisista osoitteista), 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 172.21.209.0/24 … -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 192.168.127.0/24 -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 (ne ohjataan perille paketin sisällön perusteella, yleensä onnistuu).
- --ctstate on conntrack -modulin versio --state -testistä (nykyisissä kerneleissä vaikutus on aivan sama, mutta --ctstate tuntee enemmän optioita).
- Edellinen 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 (172.21.209.119) portti 80 koneeseen tt1 (172.21.209.19) palomuurisäännöillä tt2:ssa:

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

iptables -t nat -A PREROUTING -d 172.21.209.119 -p tcp -m tcp --dport 80 -j DNAT --to-dest 172.21.209.19

# ohjataan paluuliikenne takaisin SNATilla:

iptables -t nat -A POSTROUTING -s 172.21.209.19 -p tcp -m tcp --sport 80 -j MASQUERADE

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

iptables -A FORWARD -d 172.21.209.19 -p tcp -m tcp --dport 80 -j ACCEPT
```

DNAT: esimerkki

sallitaan paluuliikenteen uudelleenohjaus:

iptables -A FORWARD -s 172.21.209.19 -p tcp -m state --state ESTABLISHED -j ACCEPT

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

iptables -t nat -A OUTPUT -d 172.21.209.119 -p tcp -m tcp --dport 80 -j DNAT --to-dest 172.21.209.19

iptables -A OUTPUT -s 172.21.209.119 -d 172.21.209.19 -p tcp --dport 80 -j ACCEPT

cat -vet...

- Joskus konfiguraatiotiedostoihin eksyy näkymättömiä kontrollimerkkejä, ja niistä voi aiheutua mitä merkillisimpiä virheitä. Useimmat editoritkaan eivät näytä niitä, mutta ne saa näkyviin cat -komennon optioilla:

  -v näytä erikoismerkit ^- ja M- muodossa, paitsi tab ja rivinvaihto

  -T näytä tab-merkit muodossa ^I

  -t sama kuin -vT

  -E näytä rivinvaihdot $:llä

  -e sama kuin vE

  -A sama kuin -vET

  -n numeroi rivit

Esimerkki: Kerberos ei toimi

- Jos Kerberos-autentikointi (pam) ei toimi, yleisimmät syyt ovat
  - /etc/krb5.conf virheellinen (väärä realm tms)
  - palomuuri blokkaa portin 88
  - kello on pielessä (Onko ntpd/chronyd asennettuna? Blokkaako palomuuri sen?)
  - käyttäjätunnus puuttuu /etc/passwd:stä (tai Kerberoksesta tai LDAPista tai NISistä tms)

Esimerkki: Kerberos ei toimi

- Omia (tai muiden ylläpitäjien) tumpelointeja epäillessä voi tutkia mitä on tehty:

  zgrep sudo /var/log/authlog*

  history # komentohistoriaa voi selata myös nuolinäppäimillä

  cat /root/.bash_history /home/tunnus/.bash_history

Esimerkki: Kerberos ei toimi

- Autentikointiongelmista tulee yleensä virheilmoitus /var/log/auth.log'iin. Kerran sieltä löytyi tällainen viesti:

  sshd[1472]: pam_krb5(sshd:auth): (user tt)
  mkstemp("/tmp/krb5cc_pam_o2YOkN") failed: No such file or directory

  Jos mkstemp valittaa ”No such file or directory”, se tarkoittaa että hakemistoa, jonka alle se yrittää tilapäistä tiedostoa luoda, ei ole - ja tässä tapauksessa todellakin /tmp oli kadonnut. Syyksi osoittautui se yleisin eli ”operator error”, samasta lokitiedostosta löytyi myös tällainen:

  xx3 sudo: xx0 : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/bin/rm -r -f /tmp/

  (name changed to protect the guilty)

  ja korjaukseksi riitti:

  sudo mkdir /tmp; sudo chmod 1777 /tmp # tai chmod o=rwxt,ug=rwx /tmp

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-bitti päällä

  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