# Oman koneen ylläpito

- Kurssin jälkeen saa yhden oman virtuaalikoneen (*tunnus1*) pitää omatoimista harjoittelua ja opiskelua varten.
- Lakeja ja JY:n sääntöjä pitää noudattaa: https://www.jyu.fi/itp/ohjeet/manuals/tietoturva-ja-kayttosaannot/kayton-ja-yllapidon-saannot/tietojarjestelmien-yllapitosaannot
- Konetta pitää ylläpitää kunnolla, erityisesti dokumentointi pitää pitää ajan tasalla.

# Oman koneen ylläpito

- Etenkin kaikenlainen henkilötietojen käsittely pitää perustella ja dokumentoida huolella. Omiin virtuaalikoneisiin ei saa lisätä muita käyttäjiä (eikä varsinkaan ylläpitäjiä) sopimatta siitä erikseen.
- Ylläpitäjiä on kuitenkin yleensä syytä olla useita, jos koneessa ylimalkaan pyörii jokin palvelu jatkuvasti. Palvelut kannattaa sammuttaa silloin kun niitä ei tarvita (esim. lomien ajaksi), testiviritykset päälle vain silloin kun niitä testataan.

# Oman koneen ylläpito

- Maailmalle näkyvistä palveluista pitää sopia erikseen (tarvittaessa rajapalomuuriin voidaan avata uusia reikiä).
- Alustakoneeseen ei kurssin päätyttyä pääse suoraan, oman koneen uudelleenasennuksesta jne pitää sopia erikseen.
- Henkilökunnalle järjestettävä admin-oikeudet (ssh, sudo) erikseen sovittavalla tavalla.

# Oman koneen ylläpito

- Asennettaessa ohjelmia niiden lisenssit pitää tarkastaa. Ubuntun ja Debianin oletusrepositoryistä ohjelmia voi asentaa melko huolettomasti, mutta erityisesti kaupallisessa käytössä sielläkin voi olla rajoituksia (sitä tosin ei yliopiston koneilla saa tehdä muutenkaan). Muiden ohjelmien lisenssit pitää lukea tarkasti.
- Asennukset pitää kirjata ylläpitolokiin.

# Oman koneen ylläpito

- Tekijänoikeudet ja tavaramerkit pitää muistaa myös kaikenlaisen maailmalta haettujen kuvien ja muun materiaalin sekä jopa koneiden ja polkunimienkin kanssa.
- Omia koneitaan saa skannailla mutta muita ei (ilman eri lupaa).
- Ylläpitäjä voi joutua vastuuseen myös muiden tekemisistä, jos suhtautuu koneensa tietoturvaan leväperäisesti!

# Oman koneen ylläpito

- Oman virtuaalikoneen saa halutessaan myöhemminkin vain pyytämällä, sitä ei tarvitse jättää päälle odottamaan mahdollista tarvetta (eikä pidäkään ellei ole aikaa sitä ylläpitää).
- Useampiakin omia koneita voi saada jos siihen on järkevä syy.

# dmesg

- Tulostaa *konsoliviestit*
- Optioita mm.

  -T "ihmisystävälliset" aikaleimat  (epäluotettava)

  -c tyhjennä puskuri tulostuksen jälkeen

  -w jatkuva tulostus
- Viestejä säilytetään pienehkössä rengaspuskurissa, vanhemmat hukkuvat nopeasti; jos halutaan säilyttää pitempään, voi tehdä cron-jobin joka kopioi viestit tiedostoon vaikka minuutin välein tyyliin

  dmesg -c >>/var/log/dmesg

# journalctl

- systemd:n (journald:n) lokien käsittely
- lokit oletuksena binäärisiä, eivät suoraan luettavissa (mutta yleensä osin ohjattu tekstitiedostoihin joko suoraan tai ryslogd:n kautta)
- /run/log/journal/*
- paljon optioita haluttujen tietojen valikointiin ja muotoiluun ja käsittelyyn, mm.

| | |
|---|---|
| -r | uusin ensin |
| -u *unit* | tietyn palvelun viestit |
| -S *since* -U *until* | viestit jälkeen/ennen tietyn ajankohdan |
| -f | jatkuva uusien viestien tulostus |

# rsyslogd

- systeemilokien käsittelijä
- työnjako rsyslogd:n ja systemd:n välillä vaihtelee
- /etc/rsyslog.conf
- /etc/rsyslog.d/*
- /var/log/*
- logger
- syslog(3)
- edeltäjä syslogd jossain myös yhä käytössä

# logrotate

- Yleinen työkalu muuten hallitsemattomasti kasvavien lokitiedostojen kierrätykseen
- Toimii lähes kaikkien tiedostoihin lokittavien ohjelmien kanssa
- Palvelukohtaiset säännöt hakemistossa /etc/logrotate.d
- Lokien kierrätystahti (päivittäin, viikottain), säilytysaika ja pakkaus määriteltävissä palvelukohtaisesti (yleensä tulee asennuspaketin mukana)
- Suoritus päivittäin: /etc/cron.daily/logrotate

# purge-old-kernels

- Ubuntun joissakin versioissa kernel-päivitykset jättävät vanhatkin kernelit paikalleen. Ennen pitkää tämä täyttää /boot-osion.
- purge-old-kernels [--keep *n*] poistaa vanhat kernelit (säilyttää *n* uusinta, oletus 2)
- Ei aina valmiiksi asennettuna, tarvittaessa

  apt-get install byobu
- Kutsuu apt-get -komentoa eikä toimi jos se on rikki (erityisesti jos päivitys tai asennus on jo kaatunut levyn täyttymiseen...)

# Levytila loppu? Checklist

- Onko levytila loppu? df; df -i; grep space /var/log/syslog

  apt-get autoremove

  apt-get clean

  - voiko jotain tarpeetonta poistaa? apt-get purge ...
- Onko kryptattuja levyosioita aktivoimatta? /etc/crypttab
  - Jos on: /etc/init.d/cryptdisks force-start, tai cryptsetup luksOpen...
- Onko swappia liikaa? top, vmstat, cat /proc/swaps, swapon -s
  - Jos on: swapoff …
- Onko jotakin jäänyt mountin alle?
- Muista päivittää /etc/fstab sekä grub ja initramfs tarvittaessa

# Levytila loppu? Checklist

- Mikä levyä täyttää?

  du | sort -n

  find /var -type f -size +10000

  lsof file

  fuser file

- Joku käyttäjä epäilyksenalainen?

  ps -fu *uid*

  lsof -u *uid*

- Avoinna oleva (ja iso/kasvava) tiedosto poistettu?

  lsof | grep deleted

# Levytila loppu? Checklist

- Onko vanhoja kerneleitä liikaa?

  ls /boot

  purge-old-kernels

    ei yleensä toimi jos esim. päivitys on kaatunut levyn täyttymiseen

  dpkg -l | grep linux-image

  apt purge linux-image-… # ei aina toimi...

  dpkg -r [--force depends] linux-image # ei ihan aina toimi

  rm /boot/*-4.4.0-121*  # vaarallista!

  rm -rf /lib/modules/4.4.0-121*

  # dpkg -r:n ja rm-tuhojen jälkeen mahdollisimman pian:

  apt -f install; apt dist-upgrade

# Levytila loppu? Checklist

- Jos LVM käytössä:
  - Onko olemassaolevissa VG:ssä tilaa? vgs, vgdisplay
    - Jos on: lvextend... (tai lvresize), resize2fs tai xfs_growfs tai ...
  - Käyttämättömiä levyjä? pvs; ls /dev/?d?; ls /dev/disk/by-uuid
    - jos on: parted tai fdisk, pvcreate, vgextend, lvextend...
  - Onko aktivoimattomia VG:tä? vgscan
    - jos on: vgchange -a y ...
  - Voiko jotain filesysteemiä pienentää?
    - jos voi: lvreduce -r -L … *tai* umount, fsck -f, resize2fs, lvreduce

# Levytila loppu? Checklist

- Ellei LVM:ää:
  - Onko jossakin osiossa tilaa?
    - jos on: mv, ln -s tai partitiointi uusiksi ja resize2fs
  - Onko käyttämättömiä levyjä tai levyllä tyhjiä osioita tai osioimatonta tilaa?
    - jos on: parted/fdisk, mkfs, mount, mv, ln -s tai koko täyden partition siirto
- Onko jonkin osion/LV:n tiedostojärjestelmä pienempi kuin itse osio? resize2fs
- Osio/LV suurennettu suurentamatta kryptattua osaa? cryptsetup resize

# Checklist: palvelu X ei toimi

- Onko tarkoituskaan toimia - löytyykö dokumentaatiota?
- service *palvelu* status
- systemctl status *palvelu*
- Levy täynnä? (df)
- Muisti vähissä? (top)
- CPU ylikuormittunut? (top)
- Lokeissa virheilmoituksia? dmesg?
- Jos lokeja ei löydy: onko (r)syslog kunnossa?

# Checklist: palvelu X ei toimi

- Konfiguraatiotiedostossa vikaa? /etc/*palvelu**, /etc/default/*palvelu*
- Jos prosessi ei pyöri, voiko sen käynnistää käsin? Onko debug-optiota? Tuleeko virheilmoituksia?
- Enabled-linkki puuttuu (etenkin www-palvelimissa)?
- Jonkin tiedoston tai hakemiston oikeudet, omistaja tai ryhmä väärin?
- Jokin kriittinen hakemisto tai tiedosto puuttuu kokonaan?

# Checklist: palvelu X ei toimi

- Verkkokonfiguraatio pielessä, etenkin jos käyttää toista konetta tietokannalle tms: reititys, netmask, dns?
- Palomuuri liian kireällä? Jokin olennainen conntrack-moduli tms puuttuu? Kirjoitusvirheitä?
- TCP wrapper pielessä (/etc/hosts.{allow,deny})?
- Kello pielessä (onko ntp tms asennettuna ja kunnossa)? Aikavyöhyke oikein?
- Tarvittava käyttäjätunnus puuttuu, lukittu, muuten rikki?

# Checklist: palvelu X ei toimi

- Jos et tiedä/muista palvelun nimeäkään:

  service --status-all

  systemctl -a
- Prosessilista auttaa usein:

  ps -ef

  top
- Lokit useimmiten /var/log -hakemiston alla

  /var/log/syslog hyvä paikka aloittaa

  zgrep /var/log/*

  du /var/log # joillakin palveluilla omat alihakemistonsa

# DHCP

- Dynamic Host Configuration Protocol: jakaa IP-osoitteita
- Pyynnöt broadcast-liikennettä, eivät kohdistettuja
- Samassa aliverkossa voi yleensä olla vain yksi dhcp-palvelin (joskus kaksi jotka synkronoivat toimintansa), ettei samaa IP:tä annettaisi kahdelle asiakaskoneelle (”villi” dhcp-palvelin voi aiheuttaa paljon pahaa)
- Sisäinen dhcp helpottaa virtuaalikoneiden kloonaamista

# DHCP

- Osoitteita voidaan jakaa dynaamisesti vapaiden osoitteiden poolista tai staattisesti asiakaskoneen hardware-osoitteen tai nimen perusteella (molempia voidaan tehdä samassa palvelimessa niin, että tunnetuille koneille annetaan kiinteät osoitteet ja muille dynaamiset)
- Palvelinohjelmistoja useita, mm. referenssitoteutus **ISC DHCPD** ja pienissä ympäristöissä (erityisesti myös virtuaalikoneiden kanssa) yleinen **dnsmasq**, joka toimii myös nimipalvelimena sekä tftp-palvelimena.

# TFTP

- Trivial File Transfer Protocol
- Hyvin yksinkertainen, vain autentikoimaton (anonyymi) download
- Käytetään erityisesti verkkoboottijärjestelyissä sekä erilaisten laitteiden firmware-päivityksissä

# TFTP

- Useita palvelintoteutuksia, mm. atftpd ja tftpd-hpa (klassinen tftpd ei toimi pxe-asennusten kanssa); esim. atftpd:n konfiguraatio kokonaisuudessaan /etc/default/atftpd; usein myös yhdistetty dhcp-palvelimeen (esim. dnsmasq)
- Konfigurointi ei yleensä vaadi kuin hakemiston, mutta useimmat toteutukset tarjoavat säätövaraa mm. aikarajoituksissa jne
- Myös komentoriviclientteja, lähinnä testaukseen:

```
$ atftp mytftp.example.org
tftp> get pxelinux.0
Received 27116 bytes in 0.1 seconds
```

# dnsmasq

- Yhdistetty DHCP- ja DNS-palvelin erityisesti pienten sisäverkkojen tarpeisiin
- DNS-toiminnallisuus rajoitettu, ei sovellu julkiseksi (eikä varsinkaan autoritatiiviseksi) palvelimeksi (ensisijaisesti vain välityspalvelin, *caching dns-forwarder*)
- Tarjoaa myös TFTP-toiminnallisuuden verkkoasennuksia varten
- Hyvin yleinen erilaisissa Linux- ja *BSD-pohjaisissa palomuuripaketeissa
- Myös (Ubuntun/Debianin) libvirtd:n sisäisen (default) verkon dns/dhcp/tftp -ratkaisu
- Konfiguraatio /etc/dnsmasq.conf, lukee (yleensä) myös /etc/hosts ja /etc/resolv.conf -tiedostoja

# iSCSI

- ”internet SCSI” (Small Computer System Interface)
- Alemman tason levynjakomekanismi kuin NFS, CIFS ym: iSCSI-levy (*target)* näkyy asiakaskoneelle (*initiator*) laitteena eikä tiedostojärjestelmänä.
- iSCSIlla voidaan jakaa levyä koneesta toiseen niin, että se käyttäytyy kuten lokaali levy: sitä voidaan partitioida, käyttää LVM:n fyysisenä volumena jne.
- Yleinen datacenter-ympäristöissä ja erityisesti juuri virtuaalikoneiden kanssa.

# iSCSI

- Mahdollistaa (virtuaalisten) levyjen migraation helposti, asiakaskoneita juurikaan häiritsemättä.
- Toimii periaatteessa kaikenlaisten SCSI-laitteiden kanssa, mutta eniten käytetään levyjen jakamiseen (joskus nauhureidenkin).
- iSCSI-palvelin on usein dedikoitu verkkolevy (tai levyjärjestelmä, *storage array*), mutta voi olla normaali palvelinkonekin (löytyy useimmille yleisille käyttöjärjestelmille).
- Ei salaa liikennettä, usein toteutettu erillisellä dedikoidulla sisäverkolla

# Migration

- Virtuaalikoneen voi siirtää yhdestä alustakoneesta toiseen monella tavalla:
  - offline: kone sammutetaan siirron ajaksi
  - pseudolive: kone "hibernoidaan" siirron ajaksi
  - live: kone on koko ajan käynnissä, katko vain hetkellinen kun verkkoyhteydet siirtyvät alustakoneesta toiseen

# Offline migration

- Virtuaalikoneen siirto alustakoneesta toiseen sammuttaen se ensin (ikäänkuin kovalevyn siirto fyysisestä koneesta toiseen):
  - Nykyisessä alustakoneessa (esim. lonka7):

    virsh shutdown kone1

    virsh dumpxml kone1 > kone1.xml

    virsh undefine kone1

    scp kone.xml kone1.img [kone1b.img...] lonka6:
  - Uudessa alustakoneessa (lonka6):

    virsh define kone1.xml # editoi ensin jos levyn polku vaihtui
- Jos alustakoneet eivät ole samassa aliverkossa virtuaalikoneen verkkokonfiguraatio pitää muuttaa, useimmiten helpointa tehdä ennen siirtoa; yleensä edellyttää vastaavaa muutosta myös nimipalveluun ja palomuureihin
- Joe levytiedoston polku vaihtuu, riittää kun sen muuttaa xml-tiedostoon

# Pseudolive migration

- Virtuaalikoneen siirto boottaamatta (”levyhibernaation” kautta, kuten läppärille hibernate ja siirto toiseen paikkaan):
  - Nykyisessä alustakoneessa (esim. lonka7):

    touch kone1.save # hakki root-oikeuksien välttämiseksi, ei tarpeen jos root/sudo

    virsh save kone1 kone1.save

    virsh dumpxml kone1 > kone1.xml

    virsh undefine kone1

    scp kone1.xml kone1.save kone1.img [...] lonka6:
  - Uudessa alustakoneessa (lonka6):

    virsh restore [--xml kone1.xml] kone1.save
- Jos alustakoneilla jaettu levy, scp ja touch-temppu ei tarpeen
- Verkkokonfiguraation ja levypolun vaihto tarvittaessa kuten edellä (jos xml-tiedosto muuttunut tarvitaan lisäksi --xml -optio)

# Live migration

- Virtuaalikoneen siirto ”livenä”, kone pysyy käynnissä vanhassa alustakoneessa kunnes siirto on valmis, vain minimaalinen katko kun viimeksi käytössä ollut muistin osa siirretään ja verkkoyhteys vaihtuu
- Edellyttää jaettua levyä (NFS, iSCSI tms), levyimagen pitää näkyä samassa polussa molempiin koneisiin
- Tiedonsiirtovaihtoehtoja useita, esim. ssh:

  virsh migrate --live kone qemu+ssh://alustakone/system
- Yleensä edellyttää root-oikeuksia alustakoneisiin
- Vanhan ja uuden alustakoneen täytyy olla samassa aliverkossa, virtuaalikoneen verkko- tai levykonfiguraatiota ei voi muuttaa

# Preseeding

- Tehtäessä useita samantapaisia asennuksia voi asennusohjelman kysymysten vastaukset laittaa valmiiksi tiedostoon **preseed.cfg** ja lisätä virt-install'in optioihin

  --initrd-inject preseed.cfg

  Huom. tiedostonimeä ei saa vaihtaa!
- preseed.cfg:n voi myös laittaa CD imageen tms (toimii myös muissa kuin virtuaalikoneasennuksissa)

# Preseeding

- Tiedoston preseed.cfg syntaksi on hankala ja tarvittavat asetukset vaihtelevat käyttöjärjestelmäversiosta toiseen, mutta pohjan saa toimivasta asennuksesta komennoilla

  sudo debconf-get-selections --installer

  sudo debconf-get-selections

  Komento debconf-get-selections löytyy paketista debconf-utils.

# Virtuaalikoneen kloonaus

- Identtisiä virtuaalikoneita voi luoda nopeimmin käyttämällä valmista levyimagea ja xml-tiedostomallia
- Vähintään koneen IP-osoite ja nimi pitää määrittää dynaamisesti (yleensä DHCP:llä) ja xml-tiedostoon konekohtaiset UUID:t ja verkkokortin MAC-osoitteet
- Käyttäjätunnuksia ja muita konekohtaisia asetuksia tehdään usein automaattisesti ensimmäisen käynnistyksen yhteydessä

# AppArmor & SELinux

- **AppArmor** on Ubuntun access control system *ohjelmille*: sillä voidaan rajoittaa mitä resursseja tietty ohjelma saa käyttää.
- Konfiguraatiotiedostot hakemistossa /etc/apparmor.d, tiedostonimenä yleensä ohjelman koko polku, jossa kauttaviivat korvattu pisteillä, esim. /usr/sbin/ntpd:lle /etc/apparmor.d/usr.sbin.ntpd
- Kvm-virtuaalikoneiden apparmor-profiilit ovat /etc/apparmor.d/libvirt -hakemistossa ja nimiltään muotoa libvirt-*UUID* ja libvirt-*UUID*.files
- Paikalliset muutokset ensisijaisesti hakemistossa /etc/apparmor.d/local
- **SELinux** (Security Enhanced Linux) on (etenkin RedHatin käyttämä) vastaava mekanismi: monipuolisempi ja järeämpi mutta myös vastaavasti monimutkaisempi ja vaikeampi käyttää ja konfiguroida kun apparmor.

# Intrusion detection

- Tuotantokäytössä olevan palvelimen valvontaa hyökkäysten ja muiden ongelmien varalta voi automatisoida erilaisilla työkaluilla.
- IDS = Intrusion Detection System

  NIDS = Network IDS, HIDS = Host IDS, PIDS = protocol-based IDS ...
- Ohjelmia mm.
  - Tripwire
  - AIDE
  - OSSEC
  - Snort
  - Samhain
- Ei korvaa palomuuria vaan täydentää sitä (usein osana palomuuripakettia, erityisesti rautapalomuuridistroissa).

# iptables: NAT-esimerkki

- Halutaan palomuurisääntö, jolla *tt1*-koneen portti 50022 ohjautuu *tt4*:n ssh-porttiin (22), siis niin, että `ssh -p 50022 tt1` ohjautuukin koneeseen *tt4.*
- Ensin reititys tt1:n kautta:

  # tiedostoon /etc/sysctl.conf'iin rivi

  net.ipv4.ip_forward=1

  # /etc/network/interfaces ens3 määrittelyyn

  up ip route add 192.168.128.0 via 172.20.209.18

# iptables: NAT-esimerkki

- Palomuurisäännöt *tt1*:ssä:

  LAN=ens3; MYIP=172.20.209.19; TT4=192.168.128.19

  iptables -t nat -A PREROUTING -i $LAN -d $MYIP -p tcp --dport 50022 -j DNAT --to-destination $TT4:22

  iptables -t nat -A POSTROUTING -o $LAN -d $TT4 -p tcp --dport 22 -j SNAT --to-source $MYIP

  iptables -A FORWARD -i $LAN -o $LAN -d $TT4 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

  iptables -A FORWARD -i $LAN -o $LAN -s $TT4 -p tcp -m state --state ESTABLISHED -j ACCEPT

# iptables: NAT-esimerkki

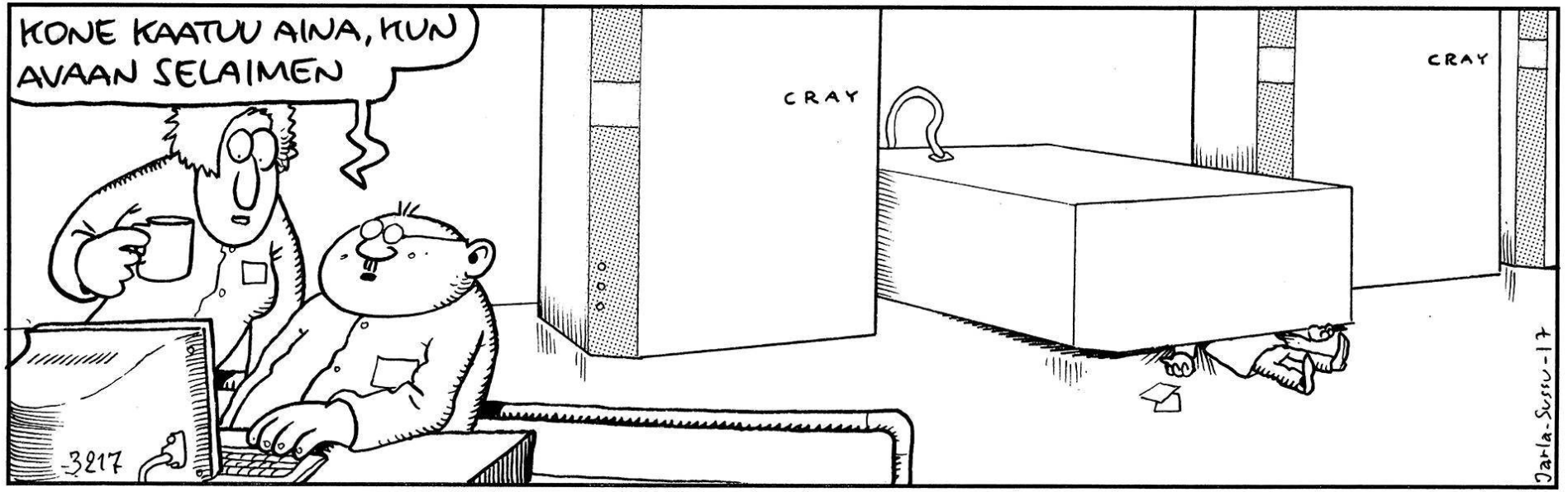
- Edellisen sivun säännöillä yhteys toimii ulkoa, mutta ei *tt1*:stä itsestään. Jos senkin halutaan toimivan, pitää tehdä vastaavat OUTPUT-säännöt:

  ```
  iptables -t nat -A OUTPUT -d $MYIP -p tcp -m tcp --dport 50022
  -j DNAT --to-destination $TT4:22

  iptables -A OUTPUT -d $TT4 -p tcp -m tcp --dport 22 -j ACCEPT
  ```

- Tässä siis DNAT-sääntö OUTPUT-ketjuun PREROUTINGin asemesta eikä SNAT-sääntöä tarvita, kun paluuliikenne jää *tt1*:een.
- Jokerikysymys: vertaa tätä ssh port forwarding'iin!

# The End


KONE KAATUU AINA, KUN AVAAN SELAIMEN
CRAY
CRAY
3217
Jarla-Sussu-17