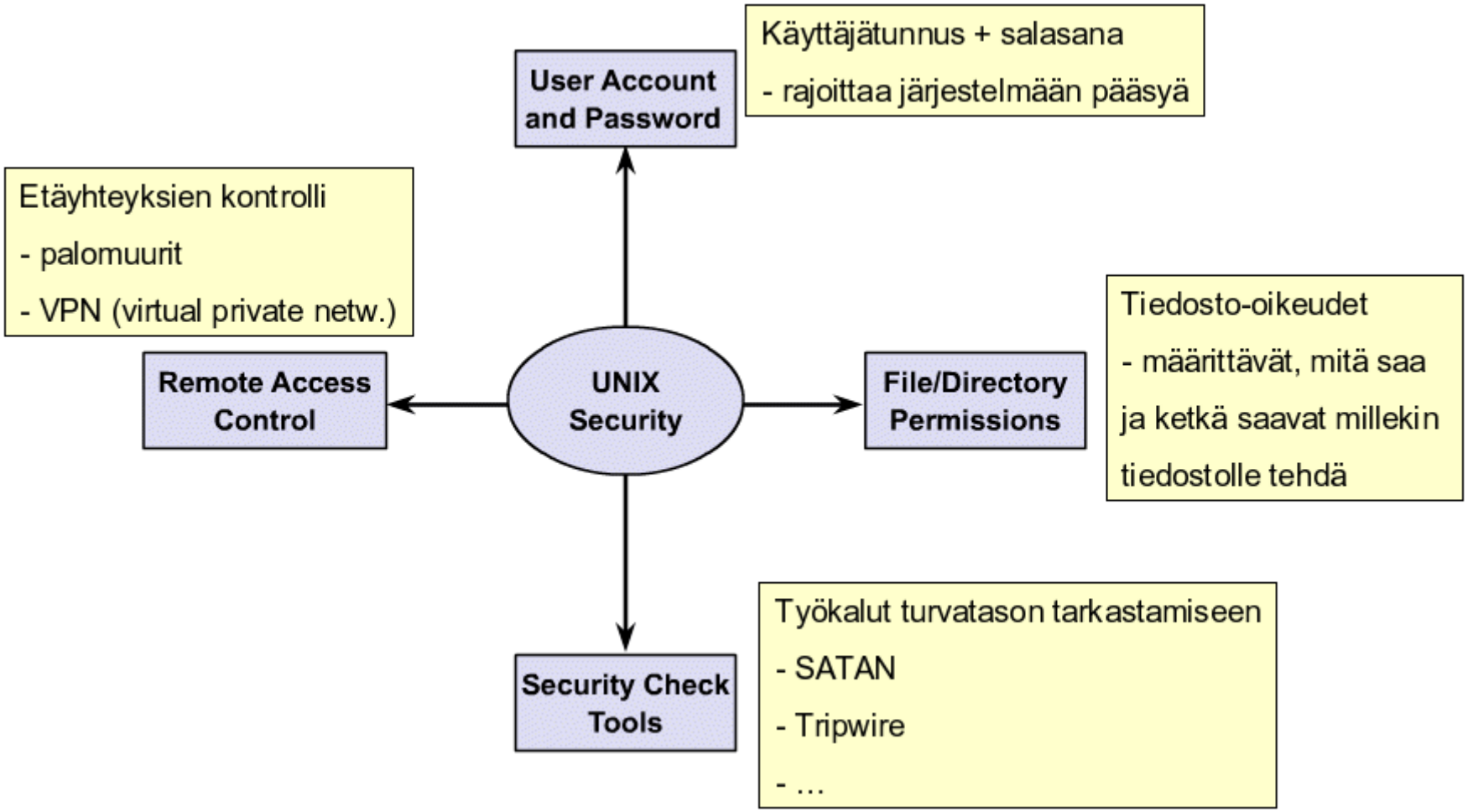
Unix-perusteet

Tiedosto-oikeudet

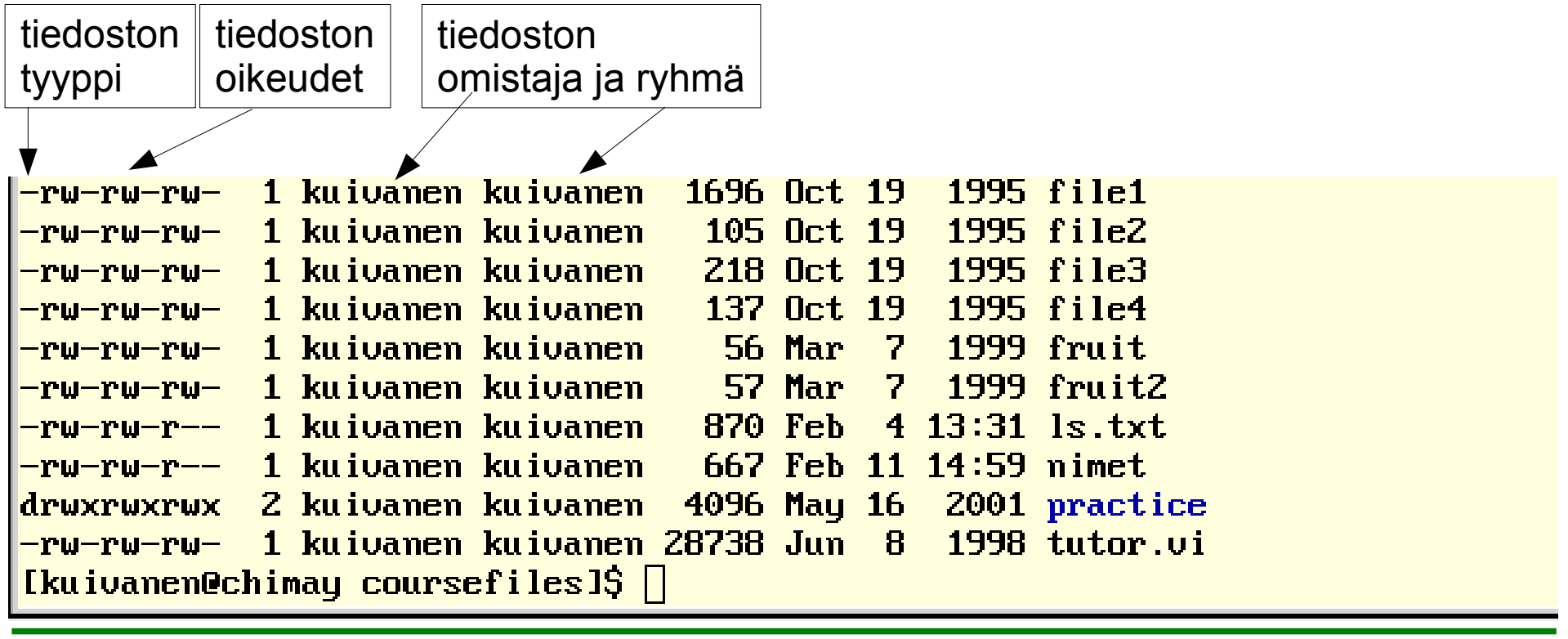
Tietoturvaan liittyviä seikkoja

- kulunvalvonta
  - kellä oikeus päästä laitteiden luokse
- käyttöoikeudet
  - käyttäjätunnus & salasana
- tiedostojärjestelmän oikeudet
  - unixissa omistajan, ryhmän ja muiden oikeudet
- viruksentorjunta
- etäyhteydet
- palomuurit
- varmuuskopiot
- palautussuunnitelma (toipumissuunnitelma)
  - varmuuskopioista ym. ei ole mitään hyötyä, jos ei tiedetä, mitä tehdään jos sattuu jotain
- auditoinnit
  - järjestelmän turvallisuuden testaaminen

Unixin perusturvaominaisuudet



Hakemistoinformaatio



Oikeuksien ryhmittely

rwx	r-x	r--
1.	2.	3.

- 1: Omistajan (user) oikeudet
- 2: Ryhmän (group) oikeudet
- 3: kaikkien muiden (others) oikeudet

- jokaisella tiedostolla on omistaja ja ryhmä

Oikeudet

- r – read
  - oikeuden omistaja saa lukea tiedostoa
  - hakemisto: saa ls-komennolla katsoa sisällön, pitkä listaus vaatii lisäksi x:n
- w – write
  - oikeuden omistaja saa kirjoittaa tiedostoon
  - hakemisto: saa kopioida tiedostoja hakemistoon, vaatii myös x:n
- x – eXecute
  - oikeuden omistaja saa suorittaa ohjelman
  - hakemisto: saa tehdä operaatioita hakemistossa, etsiä tiedostoja, katsoa pitkän listan jne.
- - - oikeus puuttuu

Hakemisto on erikoistapaus: jos haluaa antaa hakemistoon oikeuksia, pitää oikeasti antaa myös x-oikeudet.

chmod

- vaihtaa oikeuksia
  - kaksi muotoa, symbolinen ja oktaalinen
- vain omistaja ja root voi tehdä
  - symbolisesti:
    - `chmod kelle +|-|= mitä`
      - kelle = käyttäjät, eli u = user, g = group, o = others, a = kaikki ryhmät, joka on oletusarvo
      - + antaa lisää oikeuksia, - ottaa pois, = asettaa täsmälleen nämä
      - mitä = oikeudet, eli r – read, w – write ja x – execute
-

chmod, esimerkejä

- esimerkkejä symbolisesta oikeuksien vaihdosta
  - `chmod u+x foo`
    - omistajalle suoritusoikeus tiedostoon foo
  - `chmod =rw bar (taikka: chmod a=rw bar)`
    - kaikki käyttäjäryhmät saavat oikeudet rw tiedostoon bar
  - `chmod u+x,og-rw foobar`
    - omistajalle x lisää, muilta ryhmiltä rw pois
    - HUOM! ei välilyöntejä ensimmäiseen osaan (u+x,og-rw)

chmod numeerisesti (oktaalisesti)

- kuvitellaan oikeudet numeroina: 1 = oikeus annetaan, 0 = ei anneta
- `rwxr-xr--`
- `111101100` = 754
  - `==> chmod 754 foo`
    - antaa yllä olevat oikeudet tiedostolle foo
  - voidaan esittää myös näin:
    - `chmod 0754 foo`
      - näin varataan erikoisoikeuksille yksi numero

Unixin oikeustasojen ongelmia

- Ongelma 1: käyttäjä haluaa antaa vain tietylle käyttäjälle oikeudet muokata tiettyä tiedostoa
  - "perinteinen" ratkaisu: luodaan ryhmä ja siihen käyttäjä, vaatii root:n oikeudet
  - nykyratkaisu: ACL (Access Control List)
    - määritetään käyttäjä, jolle oikeuksia annetaan
    - ei tarvitse vaivata pääkäyttäjää
- Ongelma 2: kuinka estää eräidenkäyttäjien pääsy koko järjestelmään?
  - esim. silloin, kun vaikkapa www-palvelimen reiän takia demonikäyttäjä pääsee ohjelmasta "ulos"
  - ratkaisu: SELinux (Security Enhanced Linux)
    - määrittelee, mitä ko. käyttäjätunnus saa tehdä ja missä. Näin esim. koko maailman lukuoikeus tiedostossa /etc/passwd ei koskekaan tätä käyttäjää

erikoisoikeudet

- SUID
  - katsotaan oikeudet tiedoston omistajan mukaan
    - ls -l /usr/bin/passwd
    - -r-s--x--x 1 root root 13536 Jul 12 2000 /usr/bin/passwd
  - nyt käyttäjän on mahdollista vaihtaa salasanansa, vaikkei hänellä ole kirjoitusoikeuksia /etc/passwd- ja /etc/shadow-tiedostoihin
- SGID
  - oikeudet katsotaan tiedoston ryhmän mukaan
- ”Sticky bit”
  - hakemisto: käyttäjällä on oikeus poistaa omia tiedostojaan
    - drwxrwxrwt 7 root root 1024 Nov 6 04:02 tmp
  - tiedosto: jää muistiin, vaikka suoristus päättyy. Linuxissa mukana vain yhteensopivuussyistä

oletusoikeudet

- jostakin tiedostolle on oikeudet annettava, jos mitään ei erikseen mainita
- näitä varten on olemassa oletusoikeudet
  - `umask`
    - komento, joka kertoo oletusoikeudet
  - `umask 077`
    - vaihtaa oletusoikeudet. Oikeudet annetaan siis numeerisesti

kuinka oletusoikeudet vaihdetaan?

- ajattelumalli 1: suodatin, jossa merkitään 1:llä sellaiset oikeudet, joita ei anneta
- ajattelumalli 2: perusoikeudet hakemistolla 777, tiedostolla 666 (suoritusoikeutta ei oletuksena anneta), vähennetään näistä halutut. Lopputuloksena umask:n tarvitsemat oikeudet.
  - tapaus 1: tiedosto
    - halutaan oikeudet rw-r-----, eli 640
    - umask saadaan siis: 666 – 640 = 026. Komento on siis
      - `umask 026`
    - huomaa: hakemisto saa tällä oikeudet rwxr-x--x. Jos viimeinen x on liikaa, niin ota se huomioon ja komento on tällöin
      - `umask 027`
        - usein on siis turvallisempaa ajatella asiaa hakemistojen kautta

Tiedostojen omistaja ja ryhmä

- Jokaisen tiedoston omistaa joku
  - normaalisti tiedoston/hakemiston luojasta tulee sen omistaja
  - kopioitaessa tiedosto oman hakemiston alle, hakemiston omistajasta tulee kopion omistaja
- tiedoston omistaja kuuluu johonkin ryhmään
  - yo. tapauksissa tiedoston ryhmäksi tulee omistajan ryhmä
- ryhmiä on erilaisia
  - ensisijainen ryhmä
    - määritelty /etc/passwd:ssä
    - tiedoston ryhmä määräytyy ensisijaisesti tämän mukaan
  - toissijainen ryhmä
    - /etc/group
    - käyttäjä voi kuulua max. 8 - 16 toissijaiseen ryhmään (riippuu Unix-versiosta, moneenko, Linuxissa 16)

chown + chgrp

- Ensimmäinen vaihtaa omistajaa, jälkimmäinen ryhmää
  - chown uusi_omistaja tiedosto(t)
  - chgrp uusi_ryhma tiedosto(t)
- Linux:
  - vain root saa vaihtaa omistaja
  - ryhmän voi vaihtaa niiden ryhmien välillä, mihin kuuluu
- Solaris:
  - omistaja ja root voivat vaihtaa omistajaa
    - tosin tämänkin voi määrityksissä estää, jos haluaa
    - tämä on tosi asiassa jonkintasoinen tietoturvareikä…
  - jos omistaja kuuluu useampaan ryhmään
    - voi vaihtaa tiedoston ryhmää näiden välillä

käyttäjien tunnistamiseen komentoja

- who
  - kertoo keitä on järjestelmässä
  - kertoo käyttäjän todellisen identiteetin (RUID, Real User Identity)
- su [-] username
  - vaihda käyttäjätunnusta
  - pitää luonnollisesti tietää salasana
  - - lataa myös uuden käyttäjän ympäristön
  - ei käyttäjänimeä: yrittää vaihtaa super useriksi
- id
  - kertoo efektiivisen käyttäjän id:n (EUID, Effective User Id)
  - siis sen, joksi on itsensä vaihtanut
- finger username
  - antaa joukon tietoja käyttäjästä, ilman usernamea järjestelmässä olijat
  - ennen vanhaan tällä saattoi ”fingeroida” käyttäjiä muistakin järjestelmistä, ei yleensä nykyisin (tietoturvareikä!)