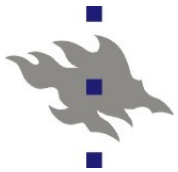
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Linux-ylläpito, kevät 2012
Verkkopalvelut

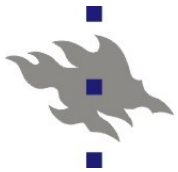
5. luentokalvosetti 12.3 – 21.3

Jani Jaakkola
jjaakkol@cs.helsinki.fi
http://www.cs.helsinki.fi/u/jjaakkol/lyp2014



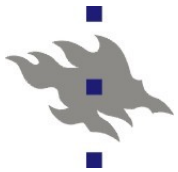
Verkkopalvelut: sisältö

- Yleistä verkkopalveluista
- SSL/TLS ja sertifikaattien hallinta
- WWW-palvelut
  - Apache, proxyt, java-sovellusalustat
  - Julkaisujärjestelmät(?)
- Tietokannat
  - Postgres, Mysql, Oracle(?)
- Mikroverkkopalvelut
  - Käyttäjätunnukset: LDAP, Kerberos, MS:n Active Directory
  - Quota
  - Tulostus
  - SAN-levyjärjestelmät: Storage Area Network
  - Tiedostojen jakaminen asiakkaille: NFS, Samba
  - Varmistuskopiointi(?)
- Virtualisointi
  - VMWare, KVM, pilvipalvelut
- Sähköposti(?)



Linux palvelinalustana

- Miksi valita Linux palvelimeen?
- Unix-tyyppisissä käyttöjärjestelmissä **etäylläpito** on aina toiminut.
  - Ei jälkeenpäin päälle liimattu ominaisuus
  - Myös graafisen työpöydän käyttäminen verkon yli
- Ohjelmistojen ilmaisuus
  - Tärkeää erityisesti grid- ja pilviympäristöissä
- Avoin lähdekoodi voi pelastaa ohjelmointitaitoisen ylläpitäjän päivän
- Palvelinpuolella Linux on hyvin tuettu
  - Kaikki suuret palvelinvalmistajat tarjoavat ajurit ja Linux-tukea
  - Dell, HP, Oracle, IBM (Lenovo)
- Linux on palvelinkäytössä hyvin stabiili
- Linux on riittävän tehokas ja skaalautuva
  - palvelimen teho ei kulu käyttöjärjestelmän pyöritykseen



Tietoturva

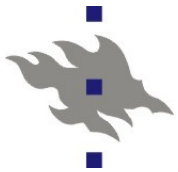
- Jokainen verkkoon näkyvä palvelu on tietoturvariski
- Linuxin palvelinohjelmistoista löytyy aukkoja yhtä säännöllisesti kuin muistakin käyttöjärjestelmistä
- Distribuution paketoimat tietoturvapäivitykset ovat paras turva ohjelmien aukkoja vastaan
- Hyväksytyn asiakasjoukon rajoituksella pienennetään riskiä (palomuurit, tcp_wrapper, lähiverkon partitiointi)
- Palveluita pyöritetään mahdollisimman vähäisin etuoikeuksin (mikä ei aina ole helppoa tai edes mahdollista)
- Palvelut on eristetty toisistaan
  - Jokaisella palvelulla oma käyttäjätunnus, oma juurihakemisto tai oma virtuaalipalvelin
- Selinux – kernelin tason järjestelmä prosessien oikeuksien rajoittamiseen

Verkkopalvelut Linuxissa

- Tyypillisesti jokaisen Linuxissa toimivan verkkopalvelun toteuttaa yksi tai useampi daemon prosessi
  - Verkkopalveluita hoitavat prosessit vastaanottavat palvelupyyntöjä IP-pistokkeiden kautta
- Palvelindaemonin käynnistystapa on jokaiselle ohjelmistolle erilainen
  - Jokaiselle ohjelmistolle on omat sysvinit/upstart/systemd käynnistysskriptit
  - Pakettien asennusskriptit luovat palvelinohjelmistolle oman eristetyn käyttäjätunnuksen ja ympäristön
  - Usein distribuution mukana tulevat palomuurit ja kernelin oikeusrajoitukset usein rajoittavat daemoneiden oikeuksia
    - Mikä aiheuttaa yllätyksiä...
- Yksinkertaisempia verkkopalveluja hoitaa *xinetd*
  - Korvaa perinteisen *inetd*:n
  - Kuuntelee konfiguraatiossa listattuja IP-pistokkeita ja käynnistää varsinaisen palveluprosessin palvelupyynnön saapuessa

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

SSL/TLS ja sertifikaatit



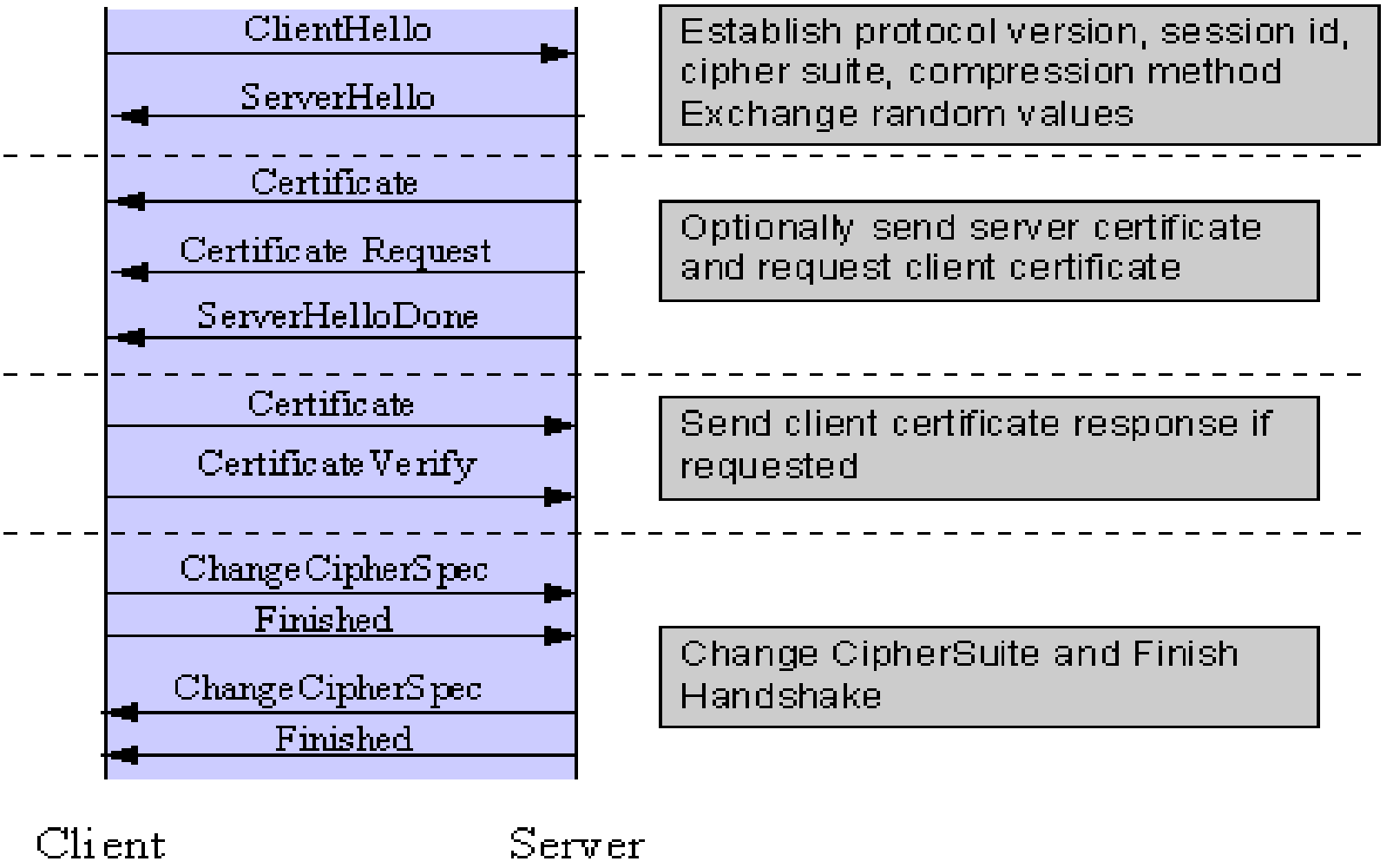
SSL/TLS-protokolla

- SSL on netscape.com:in kehittämä protokolla TCP-yhteyksien kryptaukseen
  - Secure Sockets Layer
  - V2.0 löytyi jo netscape 1.x selaimista
    - Protokolla ei ollut luotettava
  - V3.0 on ekspiroitunut ”internet draft”, netscape-4.x selaimissa toteutettu
- TLS: Transport Layer Security
  - Versio 1.0 RFC 2246 tammikuu 1999
  - Versio 1.1 RFC 4346 huhtuu 2006
  - Versio 1.2 RFC 5246 elokuu 2008
- X.509: sertifikaatit
  - RFC 5280 toukokuulta 2008
  - PKI – Public key Infrastucture
- Jatkossa näissä kalvoissa viitataan SSL/TLS/X.509 kolmikkoon SSL-nimellä

SSL:n ominaisuudet

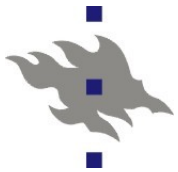
- SSL tarjoaa protokollan TCP-yhteyksien kryptaukseen, eheystarkastuksiin ja palvelimen ja asiakkaan identiteetin varmistukseen
- SSL tarjoaa seuraavat kryptografiset ominaisuudet:
  - Tiedon salaus
  - Tiedon eheys: muutokset havaitaan
  - Asiakas voi varmentaa palvelimen identiteetin palvelimen julkisen avaimen avulla (palvelimen sertifikaatti)
  - Palvelin voi varmistua asiakkaan identiteetistä asiakkaan julkisen avaimen avulla (asiakkaan sertifikaatti)
  - Ennestään tuntemattoman osapuolen identiteetti voidaan varmistaa kolmannen luotetun osapuolen avulla
    - Tällöin kyseinen kolmas osapuoli on aikaisemmin allekirjoittanut asiakkaan ja/tai palvelimen sertifikaatit (CA-sertifikaatilla)

SSL-neuvottelu



X.509 Sertifikaatti

- Sertifikaatti on (jonkin) subjektin identiteetti pakattuna jonoksi bittejä
  - Sertifikaatin formaatti on annettu X.509 standardissa
  - Formaatilla on useita esitystapoja (valitettavasti)
  - Subjekti voi olla palvelin, henkilö, yritys, jne
  - Sertifikaatin ja subjektin ominaisuudet listataan attribuuteilla
- Sertifikaatti sisältää vähintään attribuutteina:
  - Subjektin nimen
    - Serverin tapauksessa www.serveri.com tai *.verkko.com
  - Julkisen avaimen
    - Salainen avain ei tavallisesti ole osa sertifikaattia
  - Sertifikaatin myöntäjän nimen (issuer)
    - Sertifikaatin myöntäjä on tavallisesti sertifikaatti auktoriteetti (CA, certificate authority)
  - Voimassaoloajan
  - Sertifikaatin käyttötarkoituksen
    - Sertifikaatin myöntäjä hyväksyy sertifikaatin käyttötarkoitukset



Erilaisia sertifikaatteja

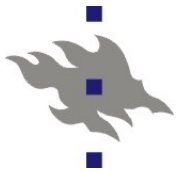
- Palvelinsertifikaatti
  - Palvelimen subjekti samaistetaan sen DNS-nimeen
- Certificate Authority (CA) sertifikaatti
  - Luotettu juuritason sertifikaatti, joka allekirjoittaa muita sertifikaatteja
- Ketjutettu CA-sertifikaatti
  - Juurisertifikaatin allekirjoittama 2. tason sertifikaatti, jolla on edelleen oikeus allekirjoittaa muita sertifikaatteja
  - Tyypillisesti rajoitetusti, esim. vain **.cs.helsinki.fi* -osoitteille
- Asiakassertifikaatti
  - Käyttäjällä oleva salainen avain (esim. Selaimessa), jolla varmistetaan käyttäjän identiteetti
  - Myös sähköpostiviestien allekirjoitukseen
  - Asiakassertifikaattia voidaan käyttää sähköpostien kryptaukseen, siten että vain sertifikaattia vastaavan salaisen avaimen omistaja voi avata viestin
- Certificate Revocation List sertifikaatit

CA: Certificate Authority

- CA on luotettu kolmas taho
  - CA:lla on myös sertifikaatti
  - CA allekirjoittaa myöntämänsä sertifikaatit
  - Allekirjoitukset verifioidaan CA:n sertifikaatin julkisella avaimella
  - CA-sertifikaatti voi edelleen olla allekirjoitettu
  - Luotettuna tahona toimiminen on liiketoimintaa!
  - Selainten ja SSL-kirjastojen mukana asentuu joukko oletusarvoisesti luotettuja CA-sertifikaatteja
  - Selaimissa ja distribuutioissa on satoja oletusarvoisesti luotettuja sertifikaatteja
  - CA:n allekirjoittama sertifikaatti maksaa rahaa
    - CA:ta motivoi ensisijaisesti tarve saada sertifikaatit kaupaksi
  - CA-sertifikaattien salaiset avaimet pidetään syvälle haudatuissa kassakaapeissa.. toivottavasti

Sertifikaatin generointi ja allekirjoitus

- Ylläpitäjä generoi sertifikaatille salaisen avaimen (ja sen julkisen puoliskon)
- Ylläpitäjä generoi Certicate Signing Request-sertifikaatin
  - CSR sertifikaattiin täytetään allekirjoitettavaksi halutut tiedot:
    - Subjekti (DNS-nimi, vaihtoehtoiset DNS-nimet, sähköpostiosoite)
    - Sertifikaatin käyttötarkoitus
    - Sertifikaatin omistaja: organisaatio, aliorganisaatio, maa, kaupunki, osavaltio
    - Sertifikaatin käyttötarkoitus
  - Tiedoista ja sertifikaatin julkisesta avaimesta lasketaan tarkastussumma, jolla sertifikaatti yksikäsitteisesti tunnistetaan
- Generoitu CSR annetaan CA:lle allekirjoitettavaksi
  - CA varmistaa jollain menetelmällä allekirjoitettavan sertifikaatin tiedot ja mahdollisesti editoi niitä
  - CA palauttaa allekirjoitetun sertifikaatin asiakkaalle
- Asiakas ottaa sertifikaatin käyttöön
  - Sertifikaatin salainen avain on koko ajan ollut vain asiakkaalla

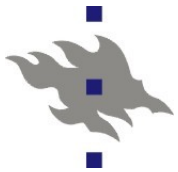


Sertifikaatin luotettavuus?

- CRL - Certificate Revocation Lists
  - Vanhempi mekanismi jolla SSL-asiakkaat voivat pyytää listan sertifikaateista joihin ei enää luoteta
- OCSP – Online Certificate Status Protocol
  - Yksinkertaisempi protokolla, jolla SSL-asiakas voi tarkistaa yksittäisen sertifikaatin luotettavuuden
- OCSP- ja CRL-protokollat olettavat, että CA itse on luotettava
  - Jos itse CA halutaan poistaa luotettujen listalta, täytyy päivittää ohjelmistot ja SSL-kirjastot
- Asiakasohjelmistot (tai ainakin SSL-kirjastot) tarjoavat mahdollisuuden merkata luotetut ja ei-luotetut sertifikaatit
- MD5 tarkastussummaan sertifikaatissa ei voi enää luottaa
  - On demonstroitu että on mahdollista generoida 2 eri sertifikaattia, joilla on sama md5-tarkastussumma
- Viime kädessä käyttäjän pitää osata itse arvioida annetun sertifikaatin luotettavuus

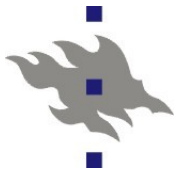
Jo toteutuneet uhkakuvat

- Vuonna 2001 Verisign myönsi kaksi koodin allekirjoitukseen tarkoitettua Microsoft Corporation subjektilla varustettua sertifikaattia tuntemattomalle taholle
- Maaliskussa 2011 Comodo CA:n tytäryhtiön järjestelmiin murtauduttiin
  - Hyökkääjät myönsivät itselleen mail.google.com, login.live.com, www.google.com, login.yahoo.com, login.skype.com, addons.mozilla.org sertifikaatit ja yhden globaalin CA sertifikaatin
  - Comodo myöntää edelleen sertifikaatteja (mm. *.helsinki.fi)
- Syyskuussa 2011 DigiNotar CA:n järjestelmään murtauduttiin
  - Hyökkääjät allekirjoittivat itselleen joukon sertifikaatteja
  - Tuntematon taho Iranissa oli käyttänyt DigiNotarin allekirjoittamaa *.google.com sertifikaattia MITM hyökkäykseen
  - DigiNotar allekirjoitti osan Hollannin valtion virallisista sertifikaateista
  - DigiNotar CA vedettiin kokonaan pois selaimista ja distribuutioista



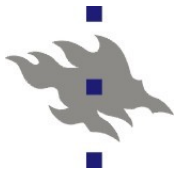
Trustwave CA:n yleissertifikaatti

- Trustwave CA on myynyt tuntemattomaksi jääneelle asiakkalleen ”mustan laatikon”, joka sisälsi TrustWave CA:n allekirjoittaman toisen juuri CA:n
- Tämä musta laatikko toimi MITM reitittimenä, joka lennossa generoi selainten ja asiakkaiden hyväksymän sertifikaatin mille tahansa internet osoitteelle (google.com, facebook.com)
- Tuntematon asiakas käytti tätä omien työntekijöidensä kryptatun verkkoliikenteen vakoiluun
- Trustwave selvisi tästä rangaistuksetta
  - Trustwave myönsi tapahtuneen
  - Tämä on ”yleinen käytäntö” (esim. valtio voi pakottaa CA:n mukaan tällaiseen toimintaan salakuunnellakseen liikennettä)
- Sertifikaatti auktoriteetteihin ei voi luottaa!
  - CA bisnesmallina on rikki: loppukäyttäjä ei ole asiakas
  - Mikko Hyppönen: "This is a nightmare scenario. You have to trust the companies selling these certificates and if we can't, then all bets are off."



Extended Validation Certificates

- Kuka tahansa domain-nimen omistaja pystyy nykyään hankkimaan itselleen virallisen CA:n allekirjoittaman sertikaatin
  - Myös huijarit ja rikolliset
- CA:t pyrkivät palauttamaan luottamuksen sertifikaatteihin kehittämällä uudet validointikriteerit
  - Sertifikaatin omistajan on oltava tunnettu oikeustoimihenkilö, jolla on myös oikea fyysinen sijainti
  - Omistajan on omistettava domain-nimi ja todistettava että domain-nimi on hänen hallinnassaan
  - Tarkastetaan sertifikaatin ostavien henkilöiden henkilöllisyys ja heidän oikeutensa toimia sertifikaatin omistajan nimissä varmistetaan
- EVC-sertifikaatilla varmistettu sivu näkyy selaimen osoitepalkissa vihreänä
  - Ja toki tuottaa CA:lle enemmän rahaa



Suojatun sivun luotettavuus

- Toimivia ja käytännöllisiä keinoja tavalliseen MITM-hyökkäykseen on useita
- Kuka tahansa voi hankkia sertifikaatin
  - Sertifikaatti varmistaa identiteetin, mutta ei kerro mitään siitä onko sertifikaatin omistaja luotettava
  - Vaikka osoitteelle www.micr0soft.com
- Sertifikaateista ei ole mitään iloa, jos käyttäjät eivät lue huolella selaimen osoiterivin tekstejä
  - Jos palveluun johtava edustasivu ei ole ssl-suojattu, niin hyökkääjän kannattaa hyökätä edustasivua vastaan
  - Merkistöhyökkäyksillä voi saada väärennetyn osoiterivin näyttämään aidolta
  - SSL-suojatulla www-palvelulla ei pitäisi enää näkyä mitään kryptaamatonta, mikä voisi päätyä google-indeksiin tai kirjanmerkiksi
- Valitettavasti mikään ei auta, koska käyttäjät klikkaavat aina OK-nappia

Mitä SSL ei tarjoa

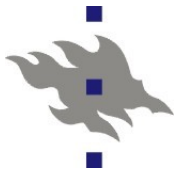
- Ei suojaa tietomurtoja vastaan
- Ei suojaa DOS-hyökkäyksiltä, vaan päinvastoin: SSL-neuvottelu syö merkittävästi CPU-aikaa
- Ei suojaa salaisen avaimen hukkumista vastaan
  - Certificate Revocation Lists (CRL) ovat standardissa (lähes) kuollut kirjain
  - Ellei palvelin ole konfiguroita vaatimaan Diffie-Helman avaintenvaihtoprotokollaa, varastetulla salaisella avaimella pystyy purkamaan talletetun SSL-kryptatun liikenteen **jälkeenpäin**
- Http-protokollan tapauksessa täytyy selaimen osoiterivin kenttä lukea hyvin tarkkaan
  - Sertifikaatti takaa täsmälleen vain ja ainoastaan osoiterivillä olevan serverin identiteetin. Sertifikaatti ei anna mitään takeita serverin luotettavuudesta
  - Osoiterivillä voi nykyään olla muitakin kuin ASCII-merkkejä
  - Osoiterivit pitäisi oikeastaan kirjoittaa aina itse, tai tallettaa kirjanmerkiksi

SSL:n protokollan ongelmia

- Vaikka itse protokolla olisi turvallinen, SSL-kirjastoissa on ollut lukuisia tietoturvaongelmia
- MD5-tarkastussummiin ei voi enää luottaa
  - Sertifikaatista allekirjoitetaan vain tarkastussumma
  - MD5 on murrettu: on mahdollista generoida kaksi eri sertifikaattia, joilla on sama md5-tarkastussumma
  - Stuxnet/Flame troijalainen asensi itsensä käyttämällä CA:n virallisesti allekirjoittamaa sertifikaattia, jossa allekirjoitus oli itse asiassa toiselle sertifikaatille jolla oli sama md5 tarkastussumma
- SSL renegotiation MITM
  - MITM ottaa yhteyden SSL-serveriin, neuvottelee yhteyden valmiiksi ja lähettää serverille joukon bittejä
  - MITM käynnistää ssl:n avainten uudelleenneuvottelun ja antaa yhteyden alkuperäisellä asiakkaalle
  - Asiakas neuvottelee uudet avaimet: palvelimen sertifikaatti validoituu ja kaikki näyttää olevan kunnossa
- Wikipediasta löytyy lisää...

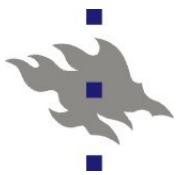
Bugit kirjastoissa ja ohjelmistoissa

- 2014: Applen IOS ja OSX SSL-toteutuksessa oli ylimääräinen *goto fail;* rivi. Sertifikaattien validiutta ei tämän seurauksena oikeasti tarkistettu
- 2005-2014: GnuTLS-kirjasto validoidessaan sertifikaattiketjuja tietyssä tilanteessa palautti OK, kun validointi epäonnistui
- Debian openssl: osasi generoida vain 32768 erilaista salaista avainta
- CA:n ohjelmistojen tuntemattomat bugit
  - Suuri osa CA:ista on täysin automatisoituja
  - Jos CA:n ohjelmistossa on bugeja, CA saattaa myöntää sertifikaatteja joita ei olisi pitänyt hyväksyä
  - Esim: Null-merkit sertifikaateissa - CA:t ja kirjastot tulkitsivat null-merkin domain nimissä eri tavalla
    - www.microsoft.com\0mundomain.com



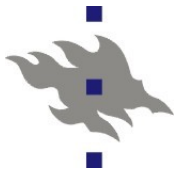
Hinta?

- Thawte 2014:
  - SSL-sertifikaatti yhdeksi vuodeksi $199
  - EV-sertifikaatti vuodeksi $299
  - Wildcard-sertifikaatti vuodeksi $499
- Sonera 2014:
  - SSL-sertifikaatti vuodeksi 190e
  - Wildcard sertifikaatti 390e
- NSA:n CA MITM hyökkäyksiin... todennäköisesti ilmainen



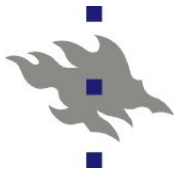
SSL ja Linux-toteutukset

- OpenSSL-kirjasto
  - Oli aikoinaan standardikirjasto, jota kaikki käyttivät
  - Työkalut sertifikaattien generointiin, allekirjoituksiin, esitysmuotokonversioihin ja tarkasteluun
  - Tuki keskitetylle CA-listalle käännetty kirjaston sisään
  - BSD-lisenssi
- Mozilla-projektin nss-kirjasto
  - Käytössä lähinnä mozilla-projektin ohjelmistoissa
  - CA-lista on käännetty kirjastobinäärin sisään: libnssckbi.so pitää kääntää uudelleen, jos haluaa lisätä uuden CA:n
  - Käytössä myös Google chromessa
- GNU TLS: GNU-projektin TLS-toteutus
  - Ei sisällä keskitettyä sertifikaattilistaa!
    - Oletetaan, että työpöydällä on avaintenhallintaohjelma käytössä
- Javan SSL(?)
- Distribuutiot ylläpitävät järjestelmän puolesta CA-listaa
  - Ubuntussa */etc/ssl/certs/ca-certifitacates.pem*



Sertifikaattien luontiprosessi

- CA:n perustaminen
  - CA generoi salaisen ja julkisen avainparin
  - Täytetään CA-sertifikaatin attribuutit, joista yksi on CA:n julkinen avain
  - Allekirjoitetaan CA-sertifikaatti sen omalla salaisella avaimella
- Sertifikaatin luominen
  - Sertifikaatin hakija luo salaisen ja julkisen avainparin
  - Hakija luo sertifikaattipyynnön, jossa on ehdotus sertifikaatissa olevista attribuuteista ja sertifikaatin julkinen avain
  - Sertifikaattipyyntö toimitetaan CA:lle, joka luo pyynnön perusteella varsinaisen sertifikaatin ja allekirjoittaa sen omalla salaisella avaimellaan
  - CA palauttaa allekirjoitetun sertifikaatin hakijalle



OpenSSL

- Varoitus: OpenSSL-kirjaston eri versiot eivät ole keskenään binääriyhteensopivia (edes taaksepäin)
- Sertifikaattiformaatit:
  - *.der* binääriformaatti
  - *.pem* base64-koodattu versio *.der* formaatista johon on lisätty aloitus- ja lopetusrivi
- OpenSSL-kirjaston mukana tulee yleiskäyttöinen SSL-työkalu nimeltään *openssl*
  - Openssl:n konfiguraatiotiedosto kuvattu config(5) manuaalisivussa
  - Ympäristömuuttuja OPENSSL_CONF
  - *openssl req -x509 -newkey rsa:2048 -out cacert.pem -outform PEM*
    - Generoi uuden CA-sertifikaatin pem-formaatissa ja sitä vastaavan salaisen avaimen

Openssl:n käyttö, osa 2

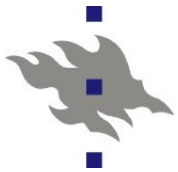
- *openssl x509 -in cacert.pem -noout -text*
  - Ihmisen luettava versio sertifikaatin sisällöstä
- *openssl req -newkey rsa:1024 -keyout private/csl2-crypted.pem -keyform PEM -out tempreq.pem -outform PEM*
  - Serverin salaisen avaimen ja sertifikaatin allekirjoituspyynnön (certificate request) generointi
  - Sertifikaatin yksityiskohdat tiedostossa csl2*.conf*
  - CA:n ei siis tarvitse nähdä itse salaista avainta: allekirjoituspyyntö riittää
- *openssl ca -in tempreq.pem -out hollikari_crt.pem*
  - Sertifikaattiallekirjoituspyynnön allekirjoitus CA:n toimesta

Gnutls: *certtool* -työkalu

- Helpompi käyttää kuin kilpailijoiden vastaavat
- Avaimen generointi:
  - *certtool --bits 2048 --generate-privkey --outfile ca-key.key*
- CA-sertifikaatin generointi
  - *certtool --generate-self-signed --load-privkey ca-key.key --outfile ca-cert.pem*
- Sertifikaatin tiedot:
  - *certtool -i --infile ca-cert.pem*
- Allekirjoituspyynnön generointi:
  - *certtool --generate-request --load-privkey server.key --outfile server.request*
- Sertifikaatin allekirjoitus CA:lla
  - *certtool --generate-certificate --load-request server.request --load-ca-certificate cert.pem --load-ca-privkey secret.pem --outfile server.cert*

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Apache HTTP-palvelin



Apache HTTP-palvelin

- Apache-projekti on sateenvarjo usealle eri OS-ohjelmistoprojektille, joista yksi on Apache HTTP-palvelin
- Apache on edelleen maailman käytetyin HTTP-palvelin
- Apache on Linuxin rinnalla menestynein OS-ohjelma
- Apache on “ilmainen”
  - Maailmalla on paljon harrastelijoiden apache-asennuksia tai vain unohtuneita palvelimia, joita kukaan ei enää ylläpidä
- Apachella on pitkä historia takanaan
  - Apache projekti perustettiin 1995 jatkamaan NCSA httpd:n kehitystä
- Apache on ”valmis”.
  - Jo apache-1.3 toteutti kaikki http/1.1:n ominaisuudet
- Apache ei ole sovellusalusta
  - Apache tarjoaa vain http-palvelun, ei ohjelmointiympäristöä www-sovellusten toteuttamiseen

Netcraft: WWW-palvelimet

- http://news.netcraft.com/archives/category/web-server-survey/

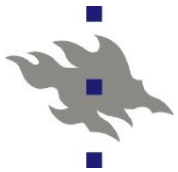
Kehittäjä:	Maaliskuu 2014	Osuus
Apache	355 M	38.60%
Microsoft	286 M	31.10%
nginx	143M	15.56%
Google	21M	2.28%

Miksi Apache?

- Yleisesti tunnettu ja käytetty
- Riittävän tehokas
- Täydellinen http-tuki
  - Tuki myös ominaisuuksille, joita kukaan ei käytä
- Skaalautuva
  - Apache hyötyy moniprosessoriarkkitehtuureista
  - Apachen proxy-tuella mahdollista myös kuormanjako ja generoitujen sivujen talletus välimuistiin
- Modulaarisuus
  - Varsinaiset www-sovellusympäristöt ovat apache-projektin ulkopuolisia ja usein toteutettu apachen moduleilla
    - php, python, perl, ruby
- SSL/TLS-tuki
  - aina tämä ei ole ollut helposti saatavilla
- Apache on hyvin dokumentoitu
  - http://httpd.apache.org/docs

Apachen ominaisuudet

- Apache valitsee selaimelta saadun URL:in perusteella jonkin palvelun, jolla HTTP-pyyntöön vastataan
- URL-avaruus konfiguroidaan sääntöpohjaisesti Apachen direktiiveillä seuraavassa järjestyksessä:
  - Globaali konfiguraatio kaikille palveluille
  - VirtualHost:in konfiguraatio
  - URL:ien konfiguraatio
  - Tiedostojärjestelmän hakemistoihin kohdistuva konfiguraatio
    - Hakemistokohtaiset *.htaccess* -tiedostot
  - Tiedostokohtainen konfiguraatio tiedoston nimen perusteella
- Sääntöjen tuloksena voidaan palauttaa:
  - Staattinen tiedosto, mahdollisesti jonkin suotimen (filter) läpi
  - Apache-modulin generoima sisältö (PHP, Python, jne)
  - CGI-skriptin generoima sisältö (Common Gateway Interface)
  - Proxyna toimiminen jollekin toiselle palvelulle (HTTP proxy)
  - HTTP-status koodi: uudelleenohjaus (HTTP redirect) tai virheilmoitus

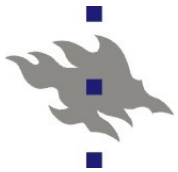


Lisää Apachen ominaisuuksia

- Skaalautuvuus joko rinnakkaisilla prosesseilla tai säikeillä
- SSL/TLS ja sertifikaatit
- Autentikointi
  - IP-osoitteella, salasanalla tai sertifikaatilla
- Proxy-tuki
  - Edustaproxy, joka jakaa palvelut intranetin sisälle
    - JavaEE sovelluksille tai kuormantasaus
  - HTTP-välimuisti Internetin ja lähiverkon välissä
- Hakemistojen ja tiedoston käsittely
  - Automaattiset hakemistolistaukset
  - Sisältöneuvottelu (Content-Negotiation)
    - Merkistöjen, kielen ja tiedostotyyppien valinta selaimen toivomusten perusteella
  - Tiedostotyyppien tunnistus
    - Hakemistolistauksille, metadatan generointiin ja sisältöneuvottelulle
- WebDAV – Sisällön etäpäivitys

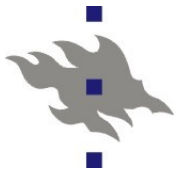
Vielä lisää apachen ominaisuuksia

- Suodattimet (filters)
  - SSI – Server Side Includes
  - Kompressointi
  - Sisällön suodatus ulkoisella ohjelmalla
- Monipuolinen lokien konfigurointi
- Modulaarisuus
  - Suurin osa Apachen ominaisuuksista on toteutettu erillisillä moduleilla
  - Esim. mod_rewrite: yleistyökalu sääntöpohjaiseen konfigurointiin
- Apachen ulkopuolisten projektien laajennusmoduleita
  - PHP-tulkki *mod_php*
  - Perl-tulkki *mod_perl*
  - Python-tulkki *mod_python*



Apachen versiot

- Versio 2.2 julkaistiin 1.12.2005
  - Parempi proxy-tuki, erityisesti www-välimuistina toimimiselle
  - Konfigurointia yksinkertaistettu
  - Mukana konfigurointiesimerkkejä erilaisiin yleisiin tarpeisiin
  - Suodatinten (filters) dynaaminen konfigurointi
  - SQL-tietokantatuki lisätty apachen ytimeen
    - Tietokantoja käyttävien modulien ei tarvitse enää toteuttaa tietokantayhteyksien luontia itse
    - Modulit voivat jakaa luotuja tietokantayhteyksiä
  - Tuki TLS-yhteyden neuvottelulle http-portissa:
  - Nimipohjaiset virtuaalipalvelimet myös SSL/TLS-yhteyksille (RFC 2817)
- Versio 2.4 julkaistiin
  - Asynkroninen IO
  - Uutta konfiguraatiosyntaksia: ehtolausekkeita, muuttujia
  - Fast CGI moduli oletusasennuksessa ja muita uusia moduleita

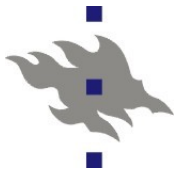


Politiikkapäätöksiä

- Rivikäyttäjien omille sivuille suunnattu palvelin ja keskitetysti ylläpidetty sovelluspalvelin tarvitsevat kovin erilaiset konfiguraatiot
- Yhdellä WWW-hotellipalvelimella voi olla tuhansia käyttäjiä
  - Miten estetään heitä aiheuttamasta vahinkoa itselleen, toisilleen ja palvelimelle?
- Sallitaanko käyttäjien asentaa omia dynaamista sisältöä tuottavia ohjelmistoja?
- Mitä eri www-sovellusalustoja halutaan tukea?
- Käyttäjien omien ohjelmien eristäminen toisistaan eri käyttäjätunnusten taakse on mahdollista, mutta työlästä ja raskaampaa kuin yhden ainoan jaetun tunnuksen käyttö
- Virtuaalikoneet ovat uusi ratkaisu ongelmaan

Apachen asennuksesta: Fedora

- 'yum install httpd php mod_python mod_perl'
  - Apache www-palvelimen kääntäminen ja asentaminen on melko helppoa
  - Ulkoisten apache modulien ja niiden tarvitsemien muiden kirjastojen ja ohjelmistojen asennus taas ei ole!
  - ts. distribuution tarjoaman valmiin apache-asennuksen käyttäminen säästää paljon aikaa
  - Distribuution oma apache-konfiguraatio voi erota paljon pakasta vedetyn apachen konfiguraatiosta
- 'yum install mod_ssl'
  - Asennetaan tai generoidaan ssl-sertifikaatti
- Editoidaan apachen konfigurointitiedosto kuntoon politiikkapäätösten mukaan
  - Helposti sanottu..
- /etc/rc.d/init.d/apache start



Fedoran apache-paketit

- Apache-moduleja:
  - *mod_dav_svn* – subversion versionhallinta http:n yli
  - *mod_ssl* – SSL tuki apacheen (openssl-kirjastolla)
  - *mod_jk* – Apache moduli tomcat:illa toteutettujen web-sovellusten liittämiseen osaksi Apachen URL-avaruutta
  - *mod_perl, mod_python, php, mod_mono*
    - Moduleja www-sovellusten kehitykseen eri kielillä
    - *mod_mono*: asp.net toteutus
- PHP on jaettu myös moduleihin
  - Tämän vuoksi PHP-riippuvuudet FC4:stä eteenpäin hieman järkevämmät
  - *php-cli* – komentorivin PHP-tulkki
  - *php_ldap* – Tuki LDAP-protokollalle
  - *php_mysql, php_pgsql* – tietokantatuki
  - *php_gd* – Grafiikan generointi php:lla

Apachen konfiguraatio: Fedora

- Apache toimii *httpd* käyttäjätunnuksen alla
- */etc/httpd* – Apachen konfiguraatiohakemisto
- */etc/httpd/conf/httpd.conf* – Apachen globaali konfiguraatiotiedosto
- /etc/httpd/conf.d – Hakemisto johon paketoidut apache-modulit pudottavat modulien konfiguraatiotiedostot
- */etc/pki/tls/private/localhost.key*
  /etc/pki/tls/certs/localhost.crt
  - mod_ssl salainen avain ja sertifikaatti
  - Fedorassa sertifikaatit ja avaimet /etc/pki hakemistossa
- */var/www*
  - DocumentRoot ja cgi-bin hakemistot
- */var/log/httpd*
  - Palvelimen lokit

Apachen konfiguraatio: Ubuntu

- Apache toimii käyttäjätunnuksen *www-data* omistamana
- Konfiguraatiohakemisto */etc/apache2*
  - *httpd.conf* – globaali konfiguraatiotiedosto
  - *mods-available* – hakemisto johon pudotetaan modulien konfiguraatiotiedostot
  - *Mods-enabled* - hakemisto josta on symlinkit käytössä oleviin Apachen-moduleihin
  - *Sites-available* – ohjelmistopaketeista asennetut http-palvelin konfiguraatiot
  - *Sites-enabled* – Käytössä olevat palvelin konfiguraatiot
  - *a2enmod* ja *a2dismod* – Skripti modulin käyttöönottoon
  - *a2ensite* ja *a2dissite* – Skripti palvelimen käyttöönottoon

Apache: rinnakkaisuus

- Apachessa rinnakkaisuuden toteutus on konfiguroitavissa
  - MPM: Multi-Processing Modules
  - Apache 2.2: MPM on staattisesti linkitetty Apache-binääriin
  - Apache 2.4: MPM toteutus on aito moduli
  - Oleellisesti valitaan käytetäänkö rinnakkaisuuden toteutuksessa säikeitä vai erillisiä prosesseja (vai jotain muuta)
- Apachen rinnakkaisuus erillisillä prosesseilla: prefork
  - Kaatuva Apache-moduli vie mukanaan vai yhden prosessin
  - Hidas jos alustana on Windows
- Apachen rinnakkaisuus säikeillä: worker
  - Yhdistelmä säikeitä ja erillisiä proseseja
  - Marginaalisesti tehokkaampi (ellei alusta ole Windows)
  - Voi paljastaa erilaisten 3. osapuolen modulien bugeja (tai moduliin itseensä ladattavien pluginien)
  - Kaatuva moduli vie mukanaan kaikki saman prosessin sisällä käynnissä olleet säikeet

Konfiguraatiosyntaksia

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

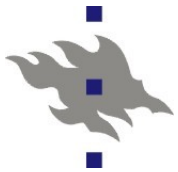
- Konfiguraatiodirektiivit noudattavat syntaksia 'direktiivi *argumentti1 argumentti2 …* '
- '*#*'-merkillä alkavat rivit ovat kommentteja
- Direktiiveillä on konteksti, jossa ne vaikuttavat
  - Konteksti voi olla palvelimen globaali kofiguraatio, VirtualHost, URL, hakemisto, virtualhost, tiedostonimi, jne
  - Jotkin direktiivit ovat käytettävissä vain tietyissä konteksteissa

Apache: globaali konfiguraatio

- *ServerRoot* – Apachen konfiguraation juurihakemisto
- Rinnakkaisuuden konfigurointi: MPM-modulien optiot
  - Myös rinnakkain käsiteltävien http-pyyntöjen maksimilukumäärä
- IP-osoitteet ja portit joita Apache kuuntelee
  - Myös SSL-portit
- Käyttäjätunnus jolla Apache-prosessi toimii
  - *User* ja *Group* -direktiivit
- Käyttöön otettavat modulit: *LoadModule*
- *DocumentRoot* – palvelimen oletusdokumenttihakemisto
  - Tyypillisesti jokaiselle VirtualHostille kuitenkin oma
- Tiedostojen, mime-tyyppien ja merkistöjen tunnistus ja oletuskonfiguraatio
- Lokitiedostojen oletuskonfiguraatio

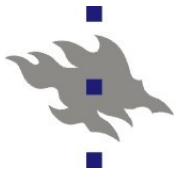
Apachen SSL-tuki

- SSL-protokollan RSA/DSA-neuvottelu on raskas CPU-aikaa kuluttava prosessi
  - Yhteyden kryptaus valmiiksi neuvotellulla avaimella on paljon kevyempi operaatio
  - Apache osaa säilyttää valmiiksi neuvoteltuja salaisia avaimia välimuistissa
  - *mod_ssl:n* SSL/TLS-toteutus ei siis ole täysin tilaton
- *mod_ssl* -paketin mukana tulee valmis SSL-VirtualHost konfiguraatio
- *mod_ssl* -paketin asennusskriptit generoivat valmiin itse-allekirjoitetun sertifikaatin
- SSL-protokolla sallisi kompressoinnin, mutta käytännössä sitä ei missään käytetä
  - Sen sijaan *mod_deflate*:n toteuttama kompressointi toimii http-tasolla yhteistyössä selainten kanssa



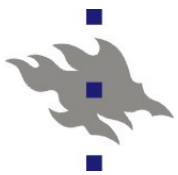
mod_ssl -konfigurointi

- *SSLOptions* – lähinnä CGI-skripteille hyödyllisten SSL-ominaisuuksien konfigurointi
  - *StdEnvVars* – SSL-neuvottelun konteksti skripteille
  - FakeBasicAuth – Asiakassertifikaatti näyttää skriptille onnistuneelta http basic autentikaatiolta
- *SSLPassPhraseDialog* – serverin sertifikaatin dekryptauksen konfigurointi
- *SSLRequire* – yleiskäyttöinen työkalu vaadittujen SSL-ominaisuuksien valintaan
- *SSLRequireSSL* – sallii pääsyn ainoastaan jos SSL-yhteys on onnistuneesti neuvoteltu
- *SSLVerifyClient* – vaatii asiakkaalta sertifikaatin



Apachen lokit

- Fedorassa */var/log/httpd* -hakemistossa
- Ubuntussa */var/log/apache2* -hakemistossa
- *access_log*
  - hakujen loki
- *error_log*
  - Virheloki
  - Tänne lokitetaan sivun käsittelyssä, apachen konfiguraatiossa tai skriptien suorituksessa tapahtuneet virheet
  - **Täältä etsitään vihjeitä, jos asiat eivät toimi!**
  - Hakuloki (access_log) ja virhelogi (error_log) ovat erikseen
- *ssl_access_log*, *ssl_error_log*
  - Vastaavat lokit SSL-kryptatuille sivuhauille
- *suexec_log*
  - Käyttäjän oikeuksin suoritettujen cgi-skriptien loki



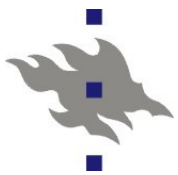
URL:ien kuvaus tiedostoiksi

- *DocumentRoot* – hakemisto josta Apache palvelee tiedostoja
  - *DocumentRoot "/home/www/htdocs"*
- *mod_userdir* - käyttäjien www-sivujen polku
  - Perinteinen *public_html* -hakemisto käyttäjän kotihakemistossa:
    UserDir public_html
  - WWW-sivut erillisessä hakemistohierarkiassa:

    UserDir /var/html

    UserDir /var/www//docs*
  - On mahdollista antaa useita vaihtoehtoisia sijainteja käyttäjän omille sivuille:

    UserDir public_html /hakemisto/jossain http://redirect.here/
  - '~'-merkkiä ei voi vaihtaa toiseksi muuttamatta apachen lähdekoodia(?)



URL:ien kuvaus tiedostoiksi

- mod_alias – URL:ien kuvaus tiedostojärjestelmään ja uulleenohjaukset
- Palveltavan hakemiston tai tiedoston lisäys palvelimelle: *Alias <URL-polku> <Tiedostopolku>*

  Alias /icons/ ”/usr/local/apache/icons”

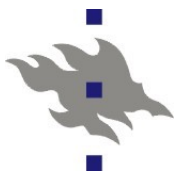
  Alias /hallinto /group/hallinto

  - Tiedostojen ei tarvitse olla *DocumentRoot* hakemiston alla
- Myös säännölliset lausekkeet käytettävissä

  AliasMatch ”^/u/tkt_legg/(.)” \ ”/home/fs/leggio/public_html/$1”*
- URL:ien kuvaus CGI-skripteiksi

  ScriptAlias /cgi-bin/ /www/cgi-bin

  ScriptAliasMatch ^/cgi-bin(.) /www/cgi-bin$1*



Yksinkertaiset uudelleenohjaukset

- mod_alias tarjoaa yksinkertaiset uudelleenohjaukset:
  - *Redirect <http-status> <polku palvelimella> <uusi URL>*
  - Esim:

*Redirect permanent /Jani.Jaakkola *
http://www.cs.helsinki.fi/u/jjaakkol
RedirectMatch (.)\.gif$ http://www.anotherserver.com$1.jpg*

- Uudelleenohjausdirektiivit sovelletaan ensin, Alias-direktiivit vasta uudelleenohjausten jälkeen
- *mod_rewrite* tarjoaa ilmaisuvoimaisempia vaihtoehtoja uudelleenohjauksiin

Ehdollinen konfigurointi

- <*IfModule*>-direktiivillä voidaan ottaa käyttöön ryhmä direktiivejä, jos jokin annettu apache.moduli on käytössä
  - Esim. SSL/TLS-direktiivejä käytetään vain, jos *mod_ssl* on käytössä
- <IfDefine>-direktiivillä voidaan ryhmä direktiivejä ottaa käyttöön, jos jokin annettu ehto on voimassa
  - Apachea käynnistettäessä voidaan komentorivin *-D* optiolla valita vaihtoehtoisia konfiguraatioita

Direktiivien ryhmittely

- Direktiivit hakemistoja, tiedostoja ja URL:eja varten
- *<Virtualhost osoite:portti>*
  - Virtuaalipalvelimeen sovellettavat direktiivit
  - *ServerName host* – direktiivi valitsee VirtualHostin nimen
- *<Location /url/>*
- *<LocationMatch regex>*
  - Annettuun URL:iin sovellettavat direktiivit
- *<Directory /folder>*
- *<DirectoryMatch regex>*
  - direktiivit vaikuttavat vain fyysisessä hakemistossa */folder* oleviin tiedostoihin
  - *.htaccess*-tiedostot ovat hakemistoon itseensä säilöttyjä *<Directory>*-ryhmiä (tietyin rajoituksin)
- *<Files *.png>*
- *<FilesMatch regex>*
  - Direktiivit vaikuttavat vain annetun nimisiin tiedostoihin

Direktiivien soveltaminen

- Yksi http-haku saattaa sopia moneen direktiiviryhmään
- Ryhmille on määritelty sovellusjärjestys:
  - Ensin <*Directory*> -ryhmät ja hakemiston *.htaccess*-tiedostosta löytyvät komennot hakemistonimen pituuden mukaan lyhyimmästä nimestä pisimpään
  - <*DirectoryMatch regex*> -säännölliseen lausekkeen sopivat
  - <*Files*> ja <*FilesMatch*> samaan aikaan
  - <*Location*> ja <*LocationMatch*> samaan aikaan
  - <*Virtualhost*> -ryhmän sisällä olevat direktiivit suoritetaan <*Virtualhost*> -ryhmän ulkopuolella olevien direktiivien jälkeen, jotta <Virtualhost> -ryhmällä voi ohittaa oletusasetukset
  - Muuten siinä siinä järjestyksessä missä ryhmät löytyvät konfiguraatiotiedostosta

Virtualhost

- DNS-nimen pohjautuva konfiguraation valinta
  - Yhdellä IP-osoitteella ajetaan eri konfiguraatioita eri DNS-nimille tuleviin pyyntöihin
  - *NameVirtualHost* ja *ServerName*
- IP-osoitteeseen pohjautuva konfiguraation valinta
  - <*VirtualHost ip-osoite:portti*>
  - Edellyttää että palvelinkoneella on useampi IP-osoitteita
  - Mahdollistaa myös useamman eri sertifikaatin käytön samalla palvelimella (myös ilman TLS-protokollan apua)

```
<VirtualHost *:80>
   ServerName swat2012.helsinki.fi
   Documentroot /home/group/swat2012/public_html
</Virtualhost>
```

Options -direktiivi

- Valitaan mitkä apachen ominaisuudet ovat käytettävissä direktiivin kontekstissa:
  - *ExecCGI* – sallitaanko CGI-skriptien suorittaminen
  - *All* – kaikki paitsi *MultiViews*. Tämä on oletusarvo
  - *FollowSymLinks* – seuraanko symbolisia linkkejä
  - *Includes* – onko apachen www-sivujen esikäsittelijä käytössä (SSI, Server Side Includes)
  - *IncludesNOEXEC* – onko CGI-skriptien suorittaminen SSI-esikäsittelijän kautta sallittu
  - *Indexes* – generoiko apache hakemistolistauksia hakemistoista, joista puutuu *index.html* -tiedosto
  - *SymLinksIfOwnerMatch* – apache seuraa vain symlinkkejä, joiden omistaja on sama kuin kohdetiedoston omistaja
  - Myös syntaksit: *Options +Foo* ja *Options -foo*

.htaccess -tiedostot

- *.htaccess* – tiedostot ovat Apachen konfigurointitiedostoja joilla hakemistotason konfiguroinnin tehdään hakemistossa itsessään
  - Mahdollistaa käyttäjien itsensä tekemän konfiguroinnin
- *Options* ja *AllowOverride* -direktiiveillä ylläpitäjä mitkä Apachen ominaisuudet ovat käytettävissä
  - Esim. CGI-skriptien sallimisella on tietoturvaseurauksia
- Pääsyoikeuksien konfigurointi
- Metatiedot (mime-tyyppi, merkistöt, kielet)
- HTTP-uudelleenohjaukset, myös Rewrite-modulilla
- SSI:n konfigurointi
- CGI-skriptien konfigurointi

AllowOverride -direktiivi

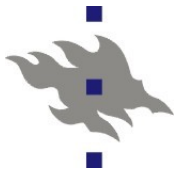
- Valitaan *.htaccess* -tiedostoilla konfiguroitavat ominaisuudet
  - *AllowOveride none* – ei käytetä .htaccess-tiedostoja lainkaan
  - *AuthConfig* – käyttäjätunnuksilla autentikoinnin konfigurointi sallittu
  - *FileInfo* – metainformaation konfigurointi sallittu (mm. Mime-tyypit)
    - Myö rewrite-sääntöjen kirjoittaminen
  - *Limit* – IP-pohjaisten pääsyoikeuksien konfigurointi sallittu
  - *Options* – sallii *Options* -komennon käyttämisen
    - Tämä voi olla vaarallinen: tällä sallitaan CGI-skriptien käytössä olon konfigurointi *.htacces* -tiedostoilla
- AllowOverrideList
  - Apache 2.4:n ominaisuus
  - Listataan sallitut konfiguraatiodirektiivit
  - Esimerkki: sallitaan vain *Redirect* ja *RedirectMatch* direktiivit

```
AllowOverride None
AllowOverrideList Redirect RedirectMatch
```

Pääsyrajoitukset

- Apache 2.2:n *mod_access* ja 2.4:n *mod_access_compat* tarjoaa yksinkertaiset IP-osoitepohjaiset pääsyrajoitukset:
  Allow from all|host|env=env-variable
  Deny from all|host|env=env-variable
- Käytettävissä DNS-nimet, ipv4-osoitteet ja ipv6-osoitteet
- *Order* -direktiivillä valitaan onko pääsy oletusarvoisesti sallittu ja missä järjestyksessä *Allow* ja *Deny säännöt* suoritetaan
- *Order Deny,Allow*
  - Pääsy oletusarvoisesti sallittu, *Allow* -säännöllä presedenssi
- *Order Allow,Deny*
  - Pääsy oletusarvoisesti kielletty, Deny -säännöllä presedenssi
- Apache 2.4
  - *Require* direktiivi korvaa vanhat pääsyrajoitusdirektiivit

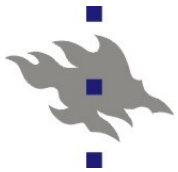
```
Require host example.org
Require host .net example.edu
Require ip 10.1.0.0/255.255.0.0
```



<Directory> esimerkki

```
<Directory /home/*/public_html>
   AllowOverride FileInfo AuthConfig Limit Indexes
   Options MultiViews Indexes SymLinksIfOwnerMatch -ExecCGI Includes
   <Limit GET POST OPTIONS PROPFIND>
      Order allow,deny
      Allow from all
   </Limit>
   <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
      Order deny,allow
      Deny from all
   </Limit>
</Directory>
```

- *Options*
  - Mitkä apachen ominaisuudet hakemistossa ovat käytössä
- *AllowOverride*
  - Mitkä konfiguraatiotiedoston säädöt voidaan vaihtaa '*.htaccess*'-tiedostolla
- Pääsyrajoitukset



Käyttäjätunnukset

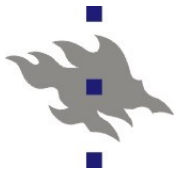
- mod_auth - tuki yksinkertaiselle salasana-autentikoinnille
- Salasanat kulkevat verkossa selväkielisinä ellei ssl-kryptaus ole käytössä
  - Vaihtoehtoisesti *mod_auth_digest* modulilla voi toteuttaa challenge-response autentikoinnin
    - Vaatii salasanat omassa salasanaformaatissa
- *AllowOverride AuthConfig* sallii autentikoinnin konfiguroinnin *.htaccess* -tiedostojen avulla
- Tarvitaan salasanatiedosto, joka sattuu olemaan syntaksiltaan samanlainen kuin */etc/shadow*
- Apachen mukana tulee pieni komentoriviohjelma *htpasswd* jolla voi editoida salasanatiedostoja
- Salasanatiedoston voi laittaa dbm-tietokannan taakse
- *AuthGroupFile <tiedosto>* direktiivillä voi salasanatiedostoista löytyviä käyttäjiä jakaa ryhmiin
- *mod_auth_pam:* pam-autentikointi
- *mod_auth_ldap:* LDAP-serveriä vastaan tehty autentikointi

Lisää käyttäjätunnuksista

- Politiikkapäätös: Jos apachen pääsee käsiksi linux-asennuksen salasanatiedostoon, se voi myös vahingossa vuotaa sen ulos verkkoon
  - Vaihtoehtoisesti voi konfiguroida apachen käyttämään esim. PAM-autentikointia tai LDAP-autentikointia, jolloin apachelle ei tarvitse antaa pääsyoikeuksia salasanatiedostoihin
- *AuthUserFile <salasanatiedosto>*
  - Salasanatiedoston sijainti. Salasanatiedosto on syytä pitää paikassa, josta Apache ei suostu jakamaan sitä verkkoon
  - Edes symlinkeillä!
- *Require user userid userid2*
  - Vaaditaan jokin listatuista käyttäjätunnuksista
- *Require group group group2*
  - Vaaditaan käyttäjän kuuluvan johonkin listatuista ryhmistä
- *Require valid-user*
  - Vaaditaan vain salasanatiedostosta löytyvä validi salasana

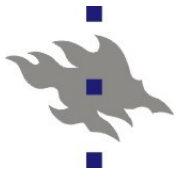
CGI-skriptit

- CGI-skriptit ovat Apachesta erillisiä ohjelmia, jotka suoritetaan vastauksen generoimiseksi http-pyyntöön
  - Yksinkertaisin tapa generoida dynaamista sisältöä
  - CGI-skriptit ovat aina tietoturvariskejä
- Politiikkapäätöksiä CGI-skriptien tapauksessa:
  - Sallitaanko omien CGI-skriptien kirjoitus rivikäyttäjälle
  - Käytetäänkö suexec-mekanismia
  - Vaihtoehtona FastCGI-palvelinprosessit
- *ScriptAlias*
  - Apachen ylläpitäjän hakemisto CGI-skripteille
- *AddHandler cgi-script .php .pl*
  - Hakemistokohtaisesti valittaan mitkä tiedostot suoritetaan cgi-skripteinä
  - *AddHandler* toimii vain jos *Option ExecCGI* on käytössä
  - *AddHandler* toimii myös *.htaccess* -tiedostoissa



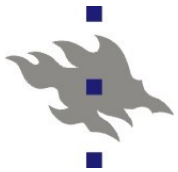
Sovellukset käyttäjän oikeuksin

- Apachen *suexec* ominaisuus mahdollistaa CGI-skriptin suorituksen skriptin omistajan oikeuksin
  - *Apache-asennuksessa on tällöin suexec* -niminen setuid-root binääri, jonka avulla apache voi vaihtaa suorittavan prosessin käyttäjätunnuksen, mikäli joukko tietoturvavaatimuksia toteutuu
  - suexec:in asennus on tehty tahallaan hankalaksi
- Suexec on käytössä jos ja vain jos
  - CGI-skripti tuli mod_userdir apache-modulilta (ts. tavallisesti public_html-hakemisto)
  - *SuExecUserGroup user group – CGI-skriptejä suorittavan käyttäjän valinta virtuaalipalvelimella*
- Suphp -paketti tarjoaa vastaavan virityksen jolla php-skriptit voi suorittaa käyttäjän oikeuksin
  - PHP safe mode on vaihtoehto



FastCGI

- CGI-prosessin käynnistäminen voi olla raskasta
  - Apachen *suexec* – modulin läpi käynnistettynä tuplasti raskasta
  - Prosessin käynnistykseen kuluu aikaa millisekunteja
- FastCGI on protokolla, jossa CGI-skriptit toimivat toisella taustapalvelimella, jolta Apache pyytää palveluita
- Taustapalvelimella palvelinprosessi on valmiiksi käynnistetty
- Käytetty skriptikieli voi itse tukea skriptien käynnistystä FastCGI-rajapinnan kautta: PHP, Python, Ruby, Perl
- Toteuttaa luonnollisella ja turvallisella tavalla käyttäjätason prosessien erottamisen Apache käyttäjästä



SSI - Server Side Includes

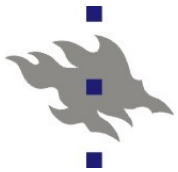
- SSI toteutettu apache-modulissa *mod_include*
- SSI on joukko html-tiedostoihin tageina upotettavia komentoja, jotka apache suorittaa
- SSI on kevyehkö menetelmä yhdistellä useasta lähteestä kokonaisia www-sivuja
- *Options +Includes* – otetaan SSI-käyttöön
- *AddOutputFilter INCLUDES .shtml* – tehdään SSI-prosessointi kaikissa .shtml-päätteisissä tiedostoissa
- *XBitHack on* – jos html-tiedostossa on x-bitti päällä tehdään SSI-prosessointi
- *XbitHack full* – SSI-käsitellyn tiedoston aikaleimaan vaikuttaa ainoastaan lähdetiedoston aikaleima

SSI-komennot

- Syntaksi:
  <!--#element attribute=value attribute=value ... -->
- Tiedostojen inkluusio:
  <!--#include virtual="/footer.html" -->
  - Myös cgi-skriptien inkluusio, mikäli polun takana itse asiassa oli cgi-skripti
- Tavallisen komennon suorittaminen:
  <!--#exec cmd="date" -->
  <!--#exec cgi=/cgi-bin/foo.pl -->
  - *#exec cgi* tapauksessa oletetaan että cgi-skripti generoi http-otsakkeet
- Myös muuttujien ja ympäristömuuttujien asetukset ja ehdollinen suoritus
- Kontekstitiedon liittäminen: milloin muutettu, tiedoston koko

Proxy-tuki

- Proxy apache vastaa saamaansa www-pyyntöön tekemällä itse haun jollekin toiselle palvelimelle
  - Proxy apache voi toimia FTP, TCP, tai Ajp-asiakkaana
- Apachea voi käyttää perinteisenä www-proxyna
  - WWW-proxya konfiguroitaessa täytyy **aina** ottaa käyttöön jonkinlainen autentikointi: spämmerit, kräkkerit ja muut ikävät ihmiset löytävät avoimet proxyt väistämättä
  - *mod_cache, mod_mem_cache, mod_disk_cache* toteuttavat välimuistin proxyn kautta haetuille sivuille
- Reverse-proxy
  - Ohjaa annettuun URL:iin tulleet http-pyynnöt toiselle palvelimelle, ts. Edustapalvelimelta taustapalvelimelle
  - Kuorman tasaus
  - Apachen ulkopuolisten http-palvelinten liittäminen osaksi virtuaalista apache URL-avaruutta
  - esim. tomcat-konfiguraatio db.cs.helsinki.fi -palvelimella



Proxyn konfigurointi

Perinteinen proxy

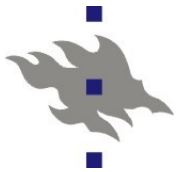
```
ProxyRequests On
ProxyVia On

<Proxy *>
Order deny,allow
Deny from all
Allow from internal.example.com
</Proxy>
```

Reverse proxy

```
 ProxyRequests Off

<Proxy *>
Order deny,allow
Allow from all
</Proxy>

ProxyPass /foo http://foo.example.com/bar
ProxyPassReverse /foo http://foo.example.com/bar
```

Välimuistin konfigurointi

```
#
# Sample Cache Configuration
#
LoadModule cache_module modules/mod_cache.so

<IfModule mod_cache.c>
#LoadModule disk_cache_module modules/mod_disk_cache.so
# If you want to use mod_disk_cache instead of mod_mem_cache,
# uncomment the line above and comment out the LoadModule line below.
<IfModule mod_disk_cache.c>
CacheRoot c:/cacheroot
CacheEnable disk /
CacheDirLevels 5
CacheDirLength 3
</IfModule>

LoadModule mem_cache_module modules/mod_mem_cache.so
<IfModule mod_mem_cache.c>
CacheEnable mem /
MCacheSize 4096
MCacheMaxObjectCount 100
MCacheMinObjectSize 1
MCacheMaxObjectSize 2048
</IfModule>
# When acting as a proxy, don't cache the list of security updates
CacheDisable http://security.update.server/update-list/
</IfModule>
```



WebDAV

- Web-based Distributed Authoring and Versioning
  - Tiedostojen etäpäivitys käyttäjille
  - RFC 2518 (http PUT, DELETE, jne)
    - Tiedostojärjestelmä http:n yli
  - Myös samanaikaisuuden hallinta
  - Dreamweaver ja MS Office tukevat
- Apache modulit:
  - *mod_dav* – toteuttaa WebDAV-protokollan
  - *mod_dav_fs* – WebDav tiedostojärjestelmässä
- Konfiguraatiossa direktiivi *Dav On*
  - Toimii hakemistoissa, joihin apache-palvelimella on kirjoitusoikeus
  - Apache ei tue talletettavan tiedoston omistajan asettamista
    - Tiedoston omistajaksi tulee Apache-prosessin käyttäjätunnus

Satunnaisia kikkoja

- Kompressointi lennossa:

  <Directory "/your-server-root/manual">

  AddOutputFilterByType DEFLATE text/html

  </Directory>

- Merkistön valinta:

  AddCharset EUC-JP .euc

  AddCharset UTF-8 README

- Tiedostojen mime-tyyppi:

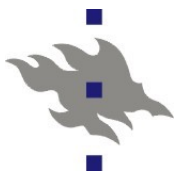
  AddType text/vnd.wap.wml .wml

- Serverin tilatieto:

  <Location /server-info>

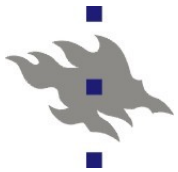
  SetHandler server-info

  </Location>



PHP: Hypertext PreProcessor

- Ohjelmointikieli nopeaan ja likaiseen www-sovellusten toteuttamiseen
  - Modulaarinen: suuri osa toiminnoista on jaettu erillisiin PHP-moduleihin
  - Monipuolinen
  - Konfigurointitiedosto */etc/php.ini*
  - Suoritetaan komentoriviltä, CGI-skriptinä tai Apachen modulina
- Apache moduli *mod_php*
  - Mahdollistaa PHP-skriptien suorituksen apache-prosessin sisällä
  - Tehokkaampaa kuin CGI-skriptinä suoritus
  - Tavallisin tapa PHP:n käyttöön
  - *mod_php* on osa virallista PHP-jakelupakettia, ei osa Apachea
  - Php-koodi suoritetaan www-palvelinkäyttäjän oikeuksilla
- Apache moduli *mod_suphp*
  - Suorittaa PHP-skriptit skriptin omistavan käyttäjän oikeuksin
  - Erillisen suid-root wrapperin avulla



Squid www-proxy

- Www-proxylla voidaan pitää organisaatiossa usein käytettyjä sivuja välimuistissa keskitetysti
  - Cache-osumat voivat vähentää latenssia
  - Säästetään www-liikenteessä, tyypillisesti n. 20-30%
- Proxy voi toimia palomuurina pahan ulkomaailman ja intranetin välissä – intranetistä ei välttämättä ole suoraa reittiä ulkomaailmaan lainkaan
- http-porttikäytäviä muihin protokolliin (esim. Ftp)
- Squid tukee pakkoproxy-konfiguraatiota
  - Pakkoproxy sieppaa kaiken http-liikenteen proxylle käsiteltäväksi suoraan verkosta
- Organisaatiolle keskitetty paikka monitoroida www-liikennettä
- Konfiguroitavat pääsyrajoitukset
- Squid tuntee myös proxy-hierarkiat ja kollegat
  - ICP – Internet Cache Protocol

Linux ja Java

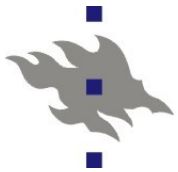
- Javan omistaa nykyään Oracle
- Oracle on kieltänyt virallisen Oracle-java paketin edelleenjakamisen, joten Oracle-java pakettia ei voi asentaa ohjelmistopakettijärjestelmän kautta
  - Poikkeuksena (ehkä?) Oraclen omat Linux-distribuutiot
- Jos Java-paketin asentaa käsin, ei saa tietoturvapäivityksiä
  - On olemassa ppa-repository, josta voi asentaa paketin joka automaattisesti lataa ja asentaa Oracle-javan.
    - Toimivuudesta ei ole mitään takeita
  - Oracle Javan tietoturvahistoria on kovin murheellinen
- Ratkaisuna OpenJDK ja IcedTea
  - OpenJDK on OS versio Oracle JDK:sta
  - IcedTea on OpenJDK:n versio joka on käännetty käyttäen vain OS-versioita Java-työkaluista

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Tomcat: Java EE alusta

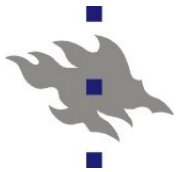
Java palvelut: Java EE

- Java Platform, Enterprise Edition, (oli: J2EE)
  - Speksi ja API web-palvelujen toteuttamiseen javalla
    - WWW-palvelut (servletit), tietokantarajapinta (JDBC), xml, etäproseduurikutsut (RMI), jne
  - Java EE sovellukset ovat (teoriassa) laitteisto-, KJ- ja toteutusriippumattomia
  - Sunin JDK on nykyään avoin ja löytyy distribuutioista
- Linux-ylläpitäjälle näkyvät osat:
  - Java EE sovelluspalvelin, jossa sovellukset toimivat
    - Sovelluspalvelinvaihtoehtoja on paljon: Tomcat, Jboss, IBM Websphere, Sun Java System Application Server...
  - Sovelluksella on standardoitu hakemistorakenne
    - Hakemiston voi zip-pakata Web Archive (WAR)-tiedostoksi
    - Hakemisto tai .war-tiedosto annetaan sovelluspalvelimen suoritettavaksi
  - J2EE-sovelluksilla on oma java-hiekkalaatikko



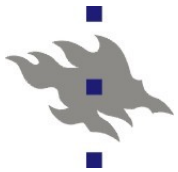
Java EE -sovellus

- Ubuntun ja Fedoran tomcat-paketissa asentuvat */var/lib/tomcat7/webapps* -hakemistoon
  - Oletuskonfiguraatiolla *webapps/<sovelluksen_nimi>* hakemistosta tiedostot näkyvät aivan kuin näkyisivät tavallisesta http-palvelimesta
    - *http://tomcat.serveri:port/<sovelluksen_nimi>*
  - *ROOT* -niminen sovellus näkyy tomcat-palvelimen juuressa
    - *http://tomcat.server:port/*
  - *<sovelluksen_nimi>/WEB_INF/web.xml*
    - *Sovelluskohtainen konfiguraatiotiedosto*
  - *<sovelluksen_nimi>/WEB_INF/classes*
    - Sovelluksen java-luokat
  - *<sovelluksen_nimi>/WEB_INF/lib*
    - Sovelluksen jar-kirjastopaketit



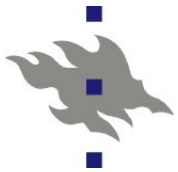
Tomcat

- Tomcat on Java EE -sovelluspalvelimen referenssitoteutus
  - *http://tomcat.apache.org/*
  - Toteuttaa sovelluspalvelimen, servlet- ja jsp-speksin
  - Tomcat löytyy distribuutioiden vakiokokoonpanosta
  - Versio 7 toteuttaa servlet speksin version 3.0 ja JSP-speksin version 2.2
  - Pohjalla Sunin omassa J2EE-referenssitoteutuksessa
  - Sovellusten dynaaminen asentaminen ja päivittäminen
  - Sovellukset näkyvät oletusarvoisesti tomcatin sisäänrakennusta http-palvelimesta
    - AJP-protokollalla tai reverse-proxylla sovellukset saa liitettyä osaksi apachen URL-avaruutta
  - *server.xml* – tomcat-palvelimen konfiguraatiotiedosto
    - Palvelimen lokit, sovellusten sijainti
    - Kuunteltavat portit: http-palvelin, mod_jk-palvelin ja portti tomcat:in alasajoa varten (salasanalla)



Tomcat - komponentit

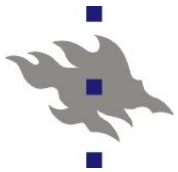
- Catalina – Servlet Container
  - Servlet Container
  - Suoritusympäristö jonka sisällä J2EE ohjelmat (servlet ja jsp) toimivat
- Coyote – Servlet Connector
  - HTTP-1.1 protokollan toteutus joka välittää http-pyynnöt Catalinalle
- Jasper – JSP Engine
  - Jasper kääntää jsp-sivut ensin java-koodiksi ja javakoodin java-luokiksi, jotka Catalina lopulta suorittaa
- Myös saatavuus (High Availability) ja klusterointituki



Fedora tomcat

- Distribuution tomcat-asennus toimii OpenJDK:lla
- Oma tomcat-käyttäjätunnus
- Konfiguraatio */etc/tomcat7* hakemistossa
- Sovellukset */var/lib/tomcat7/webapps* hakemistossa

- Oletuskonfiguraatiossa kuuntelee http-connector portissa 8080 ja ajp-connector portissa 8009
  - *server.xml* - tomcat-daemonin konfiguraatiotiedosto
    - Portit, joita tomcat kuuntelee
    - *web.xml* – globaali konfiguraatiotiedosto kaikille J2EE-sovelluksille, jonka formaatti on speksattu J2EE standardissa



Tomcat 7 Ubuntussa

- Asennetaan ubuntu-paketit:
  - *tomcat7* – tomcat engine
  - *tomcat7-examples* – esimerkkisovellukset
  - *tomcat7-user* – skripti käyttäjäkohtaisen tomcat-enginehakemiston alustamiseen
    - Kuten TKTL:n *wanna-tomcat*
- */var/lib/tomcat7/webapps* – tomcat7 sovellukset
- */etc/tomcat7* - konfiguraatiotiedostot
- */var/log/tomcat7* – lokitiedostot

mod_jk

- Apache moduli, jolla tomcat-sovelluspalvelimen voi liittää osaksi apachen URL-avaruutta
  - Välittää ajp13-protokollalla Apachelle tulleet pyynnöt java-sovelluspalvelimelle
  - Ajp13 on TCP-protokolla: sovelluspalvelimen ei välttämättä tarvitse sijaita samalla fyysisellä palvelimella
  - Voi käyttää kuorman jakamiseen usealle fyysisellä tomcat-palvelimelle
  - Tukee autentikointia jaetulla salaisuudella
  - Vastaavan toiminnallisuuden voi toteuttaa *mod_proxy*:llä
  - Mahdollistaa staattisten tiedostojen palvelun suoraan sovelluspalvelimen hakemistoista
- Konfigurointi
  - *JkMount <URL> <työläinen>*
  - *workers.properties* -tiedosto, jolla sidotaan Apache:n URL-avaruuteen liitetyt työläiset tomcat-sovelluspalvelimen instansseihin