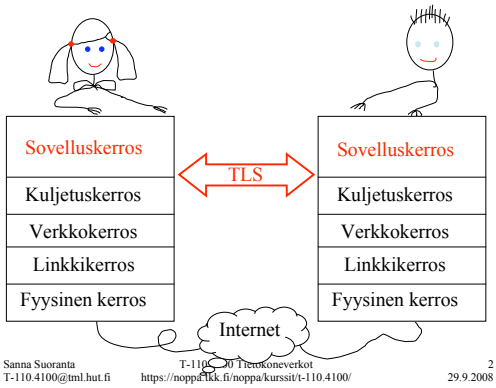
Kuljetus- ja sovelluskerroksen tietoturvaratkaisut

Transport Layer Security (TLS)

ja

Secure Shell (SSH)



Transport Layer Security (TLS)

- Sopii monenlaisille sovellusprotokollille, esim
  - HTTP
  - Sähköposti (IMAP)
- Toimi luotettavan kuljetusprotokolla päällä (esim. TCP)
- Perustuu Netscapen SSL 3.0:aan
  - Eivät toimi yhteen ”kuin periaatteessa”

TLS:n turvaama HTTP

- Kuin käyttäisi tavallista TCP-yhteyttä
  - Kuitenkin ensin sovitaan TLS-yhteyden avaamisesta
- Ennen HTTP-yhteyden sulkemista, suljetaan TLS-yhteys
  - HTTP-palvelimen toivuttava asiakkaan epämääräisestä toiminnasta
  - Jos HTTP-yhteys suljetaan suoraan, ei voida uudelleenkäyttää TLS-yhteyttä

TLS:n suojaama sähköposti

- SMTP-palvelin voi käyttää TLS:ää toisten SMTP-palvelimien ja asiakkaiden tunnistamiseen ja yhteyden suojaamiseen
  - Vain jos palvelimeen ei julkisesti viitata (eli se ei ole MX-rekordissa DNS:ssä!
  - Ei estä sähköpostin kuljetusta runkoverkossa
  - Mutta ei kyllä takaa tietoturvaakaan yleisessä tapauksessa
  - SMTP-yhteys ei ole päästä-päähän yhteys!

TLS:n suojaama sähköposti

- Asiakasohjelman ja sähköpostipalvelimen välinen salasanakysely suojataan TLS:llä IMAP:ssa (Internet Message Access Protocol)
  - Alkuperäinen selväkielistä salasanakyselyä käyttävä menetelmä kielletään
- Kun autentikoitu yhteys on käytössä, uusitaan palvelinta koskevat tiedot sitä käyttäen
- Ei suojaa sähköpostin siirtoa lähettäjältä vastaanottajalle

TLS:n tavoitteet

1. Turvallinen yhteys kahden osapuolen välille
2. Ohjelmoijat pystyvät luomaan TLS:ä käyttäviä sovelluksia välittämättä toistensa koodista
3. Uudet salausmenetelmät on helppo lisätä olemassaolevaan TLS:än sitä muuttamatta
4. Tehokas toiminta yhteyksiä tallettamalla

TLS:n toiminta

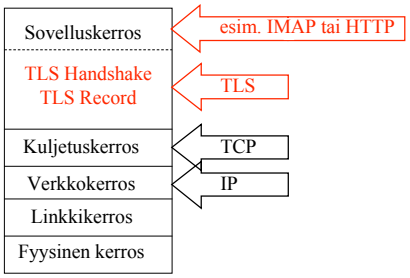
- Viestivien osapuolten
  - Yksityisyys
  - Siirretyn tiedon eheys

  turvataan
- Jokaiselle istunnolle oma symmetrinen salausavain (ei pakko käyttää salausta)
- Viesteihin lasketaan MAC (Message Authentication Code) eheyttä suojaamaan

TLS:n osaprotokollat

- TLS-kättelyprotokolla (Handshake Protocol)
  - Asiakas ja palvelin tunnistavat toisensa
  - Salausalgoritmin ja –avaimen sopiminen
- TLS Record Protocol
  - Istunnon viestien suojaus salaamalla
  - Myös TLS-kättelyprotokolla käyttää tätä

Protokollapino



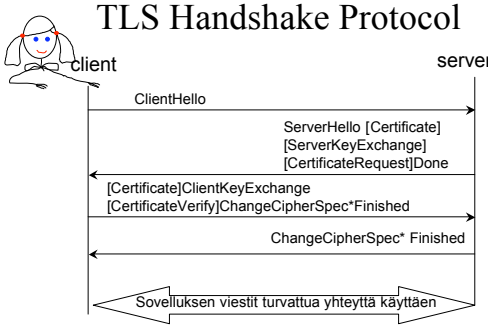
TLS-kättely

- Osapuolen identiteetti voidaan tunnistaa joko julkisen tai salaisen avaimen menetelmällä
  - Yleensä ainakin toinen (eli palvelin) osapuolista tunnistetaan
- Sovitaan symmetrinen istuntoavain
  - Man-in-the-Middle hyökkäyksen esto
  - Hyökkääjä ei voi muuttaa eikä lukea viestejä

TLS-kättely

- Algoritmien sopiminen, satunnaislukujen (nonssien) vaihto
- Yhteisen salaisuuden luominen
- Sertifikaattien vaihto ja osapuol(t)en tunnistus
- Istuntoavainten luonti
- Vastapuolen tilan tarkistaminen (eli yhteinen salaisuus todella on ko. osapuolten välinen)

TLS Handshake Protocol



TLS ClientHello

TYPE=1	LENGTH	
VERSION		RANDOM (client)
RANDOM(CONT.) 256 bits		
	ID LENGTH	SESSION ID..
SESSION ID (variable)		CIPHER LEN
CIPHER SUITES		
COMP. LEN	COMP MET	

TLS ClientHello

- RANDOM: gmt_unix_time+random
- SESSION ID: (pituus = 0, jos ei ole)
  - Asiakasohjelman toivoma tunniste
- CIPHER SUITE LIST
  - Asiakasohjelman tukemat salausmenetelmät, ensimmäisenä ensisijainen
  - Jokaisella kahden oktetin mittainen tunniste
- COMPRESSION METHOD
  - Asiakasohjelman tukemat tiivistysmenetelmät
  - Jokaisella oktetin mittainen tunniste

TLS ServerHello

TYPE=1	LENGTH	
VERSION		RANDOM (server)
RANDOM(CONT.) 256 bits		
	ID LENGTH	SESSION ID..
SESSION ID (variable)		
	CHOSEN CIPHER	CHOS.COMP

TLS ServerHelloDone

TYPE=14	LENGTH=0

SESSION ID Hello-viesteissä

- Jos asikasohjelman lähettämä session id ei löydy palvelimen tietokannasta
  - uusi istunto perustetaan
- Jos löytyy ja palvelin suostuu
  - Uudelleenavataan vanha istunto
- Jos palvelin lähettää tyhjän session id:n
  - Istunnon tietoja ei tallenneta eikä sitä voi siten avata uudelleen.

Avaintenvaihto

- Asiakasohjelma lähettää avainmateriaalia master-avainta varten
  - salattuna palvelimen julkisella avaimella
- Yhdessä satunnaislukujen kanssa ko. avainmateriaalista saadaan master-avain
- Istuntoavaimet luodaan master-avaimesta

ClientKeyExchange

TYPE=16	LENGTH
LENGTH2	MASTER SECRET (var)

ServerKeyExchange

TYPE=12	LENGTH
LENGTH (MODULUS)	MODULUS (var)
LENGTH (EXPONENT)	EXPONENT (var)
LENGTH (SIGNATURE)	SIGNATURE (var)

Sertifikaatit

- Palvelimen pitää (yleensä) lähettää sertifikaattinsa heti
  - Yleensä X.509-sertifikaatti
  - Sisältää palvelimen julkisen avaimen ja tunnisten
- Palvelin saattaa pyytää asiakasohjelmaa kertomaan sertifikaattinsa
  - Jos asiakkaalle ei ole sertifikaattia palvelimen sitä vaatiessa, kättely epäonnistuu

Certificate request

TYPE=13	LENGTH
Len typelist	list of types of keys (var)
Len CA list	Len CA name 1
CA 1 name (var)	
...	

Certificate

TYPE=11	LENGTH	
LENGTH2		CERT Len1
(jatkuu)	CERT 1 (var)	
CERT Len 2 ...		CERT 2 ...

https://www.example.com/home.html

- Palvelimen nimen (URI:ssa) pitää olla sama kuin sertifikaatissa mainittu
  - Jos asiakkaalla on jotain muuta tietoa palvelimesta, ei nimeä tarvitse tarkistaa
  - Jos nimi ei täsmää, siitä pitää kertoa käyttäjälle ja yhteys pitää katkaista
- Jos URI saatu epäluotettavasta paikasta, voi hyökkääjä vaihtaa URI:n, jolloin nimen tarkistaminen ei riitä

ChangeCipherSpec-viesti

- Kumpikin osapuoli kertoo toisilleen, kun siirtyy käyttämään sovittuja salaus- ja tiivistysmenetelmiä
  - Viesti sisältää yhden tavun salattuna ja tiivistettynä sovitulla menetelmällä
- Kättelyn päättävät Finished-viestit, jotka lähetetään heti tämän viestin jälkeen

TYPE=20	VERSION	LENGTH
=1	=1	

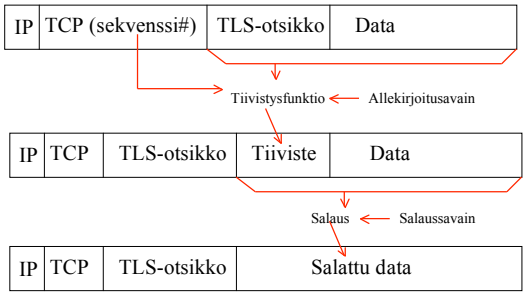
TLS Record Protocol

- Yhteyden tilatieto
  - Osapuolet: onko osapuoli asiakas vai palvelin
  - Käytetyt tiivistys-, salaus- ja eheydensuojausalgoritmit
  - Käytetyt avaimet: juuriavain ja istuntoavain
- Neljäntyyppisiä tietueita
  - Käyttäjän data, kättelyviestit, hälytykset, ja salauksen muutokset

TLS Record Protocol

1. Tiivistetään käyttäen tiivistysfunktiota
2. Lasketaan HMAC
3. Salataan
   – Täytteen avulla saadaan paketin pituus täsmäämään, jos käytetään lohkosalaajaa.

Suojaus

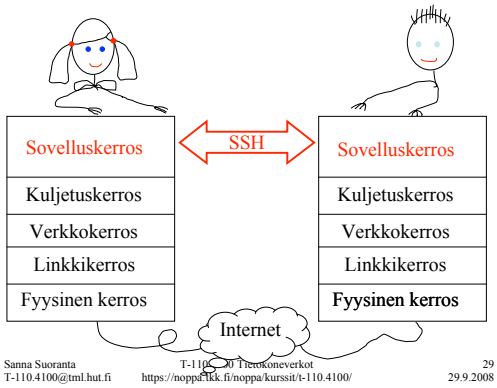


TLS:n laajennukset

- RFC 3546 määrittelee laajennuksia
  – Palvelimen nimi
  – Asiakkaan sertifikaatin sijainti
  – Sertifikaatin tila
  – Luotetut sertifiointiauktoriteetit
  – Jne

TLS

- Sopii erilaisille sovelluksille
- Salaus ja eheys, TCP:n päällä
- Erilliset avaus- ja sulkemismekanismit TCP:n lisäksi
- Tunnistus perustuu sertifikaatteihin



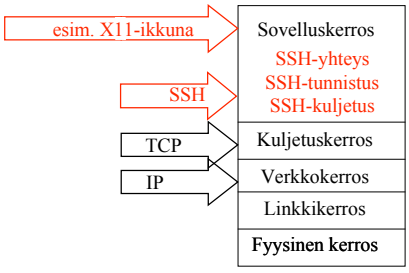
Secure Shell (SSH)

- Turvallinen etäyhteys palvelimeen
- Myös muita turvapalveluita yli turvattoman verkon
  - Minkä tahansa TCP/IP:n portin ja X11-yhteyden tunnelointi
  - Toimii TCP:n päällä (eli tarvitsee luotettavan kuljetuskerroksen)

SSH:n komponentit

- kuljetus: Transport layer protocol
  - Palvelimen tunnistus, luotettavuus ja eheys
- tunnistus: User authentication protocol
  - Asiakasohjelman tunnistautuminen palvelimelle
- yhteys: Connection protocol
  - Limittää salatun tunnelin useampaan loogiseen kanavaan

SSH:n pino



Palvelimen avain

- Palvelimella pitää olla avain
  - Avaimia voi olla useampia eri algoritmeja varten
  - Useammalla palvelimella voi olla sama avain
- Palvelimen avainta käytetään avaintenvaihdon yhteydessä tunnistamaan palvelin asiakkaalle
  - Asiakkaan pitää tietää avain etukäteen

Luottamusmallit

- Asiakasohjelmalla on paikallinen tietokanta, johon on tallennettu avain-nimipareja
  - Ei tarvita luotettavaa kolmatta osapuolta tai keskitettyä hallintaa
- Jokin luotettu kolmas osapuoli (CA) varmentaa palvelimen avaimet
  - Asiakas tietää etukäteen CA:n juuriavaimen
- Ensimmäisen yhteyden aikana ei tarkisteta

SSH Transport Layer Protocol

- Konetason tunnistus (ei käyttäjän)
- Salaus ja eheydensuojaus (+tiivistys)
  - Kumpaankin suuntaan oma
  - Joka yhteydellä omat avaimensa
- Avaintenvaihtoon käytetään Diffie-Hellman-menetelmää

Yhteyden sopiminen

- Kumpikin osapuoli lähettää listan tuetuista algoritmeista
  - salaukselle, allekirjoitukselle ja tiivistämiselle
  - Toisen osapuolen menetelmät voidaan arvata
- Suora ja epäsuora tunnistus
  - Allekirjoitus tms
  - Todistus, että palvelin tuntee yhteisen salaisuuden
- Yhteinen salaisuus ja tiiviste
  - Salaus ja eheysavaimet lasketaan näistä

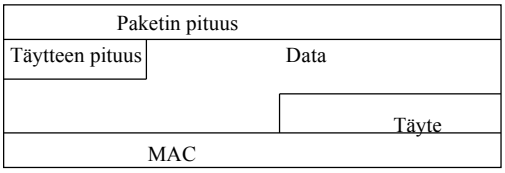
SSH: Käyttäjän tunnistaminen

- Toimii SSH:n kuljetusprotokollan päällä
  - Eheys ja luottamuksellisuus
- Palvelin kertoo käytettävissä olevat menetelmät, joista asiakas valitsee
  - Julkinen avain
  - Salasana
  - Tunnettu tietokone
  - (ei käyttäjän tunnistusta)

SSH-yhteys

- Etäyhteys, komentojen suorittaminen, TCP/IP-porttien ja X11-yhteyksien edelleenlähetys
  - Samaan salattuun tunneliin
  - Kumpi tahansa osapuolista voi avata yhteyskanavan ja kanavia voidaan sulkea
- Käyttää edelläesitettyjä SSH:n kuljetus- ja tunnistusprotokollia

SSH-yhteys



SSH

- Turvallinen etäkäyttö
  - TCP/IP-porttien edelleenlähetys
  - X-ikkunoiden tunnelointi
- Toimii TCP:n päällä
- Palvelimen ja käyttäjän tunnistus
  - useita tapoja
- Yhteyksien limittäminen samaan turvattuun tunneliin

	IPsec	TLS	SSH
Tunnistus			
Salaus			
Eheys			
Man in the Middle			
Toistohyök.			
Yksityisyys			
DoS			

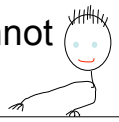
Lähteet

- RFC 2246 The TLS Protocol Version 1.0, 1999
- RFC 3546 Transport Layer Security (TLS) Extensions, 2003
- RFC 2595 Using TLS with IMAP, POP3 and ACAP, 1999
- RFC 2818 HTTP over TLS, 2000
- SSH Protocol Architecture, 2005
  - Draft-ietf-secsh-architecture-22.txt
- SSH Tranposrt Layer Protocol, 2005
  - Draft-ietf-secsh-transport-24.txt
- SSH Connection Protocol, 2005
  - Draft-ietf-secsh-connect-25.txt
- C. Kaufman et al. Network Security, Prentice Hall, 2002

Lisää turvakursseja

- T-110.5110 Computer networks II – advanced features
- T-110.4200 Tietoturvallisuustekniikka
- T-110.5200 Tietoturvallisuuden laboratoriotyöt
- T-110.5211 Cryptosystems
- T-110.5220 Tietoturva ja käytettävyys
- T-110.5230 Special course in practical security of information systems
- T-110.5290 Seminar on network security
- T-79.5501 Cryptology
- T-79.5502 Advanced course in cryptology

Seuraavat luennot



2.10. Bengt Sahlin: DNS & DNSsec

Huom!
Doctoral thesis defense by Teemu Koponen 2.10. at 12 in T1
Title of the thesis: A Data-Oriented Network Architecture
Opponent: Professor Jon Crowcroft, University of Cambridge, UK