



Welcome to

4. Encrypting the Network

Communication and Network Security 2019

Henrik Lund Kramshøj hik@zencurity.com

Slides are available as PDF, kramse@Github
4-Encrypting-the-Network-Layer.tex in the repo [security-courses](#)

Plan for today



Subjects

- Basic cryptography
- Encryption Decryption
- Hashing
- Short introduction to algorithms RSA, AES
- Diffie Helman exchange
- Transport Layer Security (TLS)

Exercises

- mitmproxy <https://mitmproxy.org/>
- sslsplit <https://www.roe.ch/SSLsplit>
- sslstrip <https://moxie.org/software/sslstrip/>
- <https://www.ssllabs.com/> and sslscan checking servers

Reading Summary, continued



Skim:

<https://tools.ietf.org/html/rfc5246> The Transport Layer Security (TLS) Protocol Version 1.2

https://en.wikipedia.org/wiki/Transport_Layer_Security



Reading Summary, continued



It is not clear that the link layer is the right one for security. In a coffeeshop, the security association is terminated by the store: is there any reason you should trust the shopkeeper? Perhaps link-layer security makes some sense in a home, where you control both the access point and the wireless machines. However, we prefer end-to-end security at the network layer or in the applications.

Source: Cheswick-chap2.pdf Firewalls and Internet Security: Repelling the Wily Hacker , Second Edition, William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin

kryptering, OpenPGP



kryptering er den eneste måde at sikre:

- fortrolighed
- autenticitet

kryptering består af:

- Algoritmer - eksempelvis RSA
- *protokoller* - måden de bruges på
- programmer - eksempelvis PGP

fejl eller sårbarheder i en af komponenterne kan formindske sikkerheden

PGP = mail sikkerhed, se eksempelvis Enigmail plugin til Mozilla Thunderbird

PGP/GPG verifikation af integriteten



Pretty Good Privacy PGP

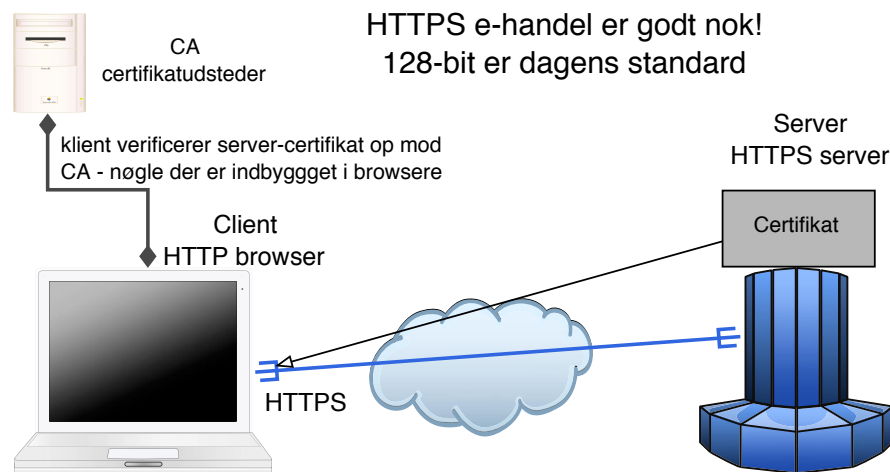
Gnu Privacy Guard GPG

Begge understøtter OpenPGP - fra IETF RFC-2440

Når man har hentet og verificeret en nøgle kan man fremover nemt checke integriteten af software pakker

```
hlk@bigfoot:postfix$ gpg --verify postfix-2.1.5.tar.gz.sig
gpg: Signature made Wed Sep 15 17:36:03 2004 CEST using RSA key ID D5327CB9
gpg: Good signature from "wietse venema <wietse@porcupine.org>"
gpg:                aka "wietse venema <wietse@wzv.win.tue.nl>"
```

SSL og TLS



Oprindeligt udviklet af Netscape Communications Inc.

Secure Sockets Layer SSL er idag blevet adopteret af IETF og kaldes derfor også for Transport Layer Security TLS TLS er baseret på SSL Version 3.0

RFC-2246 The TLS Protocol Version 1.0 fra Januar 1999

SSL/TLS udgaver af protokoller



Check with your system administrator before changing any of the advanced options below:

IMAP Path Prefix:

Port: ☒ Use SSL

Authentication:

Mange protokoller findes i udgaver hvor der benyttes SSL

HTTPS vs HTTP

IMAPS, POP3S, osv.

Bemærk: nogle protokoller benytter to porte IMAP 143/tcp vs IMAPS 993/tcp

Andre benytter den samme port men en kommando som starter:

SMTP STARTTLS RFC-3207

Secure Shell - SSH og SCP



Hvad er Secure Shell SSH?

Oprindeligt udviklet af Tatu Ylönen i Finland,
se <http://www.ssh.com>

SSH afløser en række protokoller som er usikre:

- Telnet til terminal adgang
- r* programmerne, rsh, rcp, rlogin, ...
- FTP med brugrid/password

SSH - de nye kommandoer er



kommandoerne er:

- ssh - Secure Shell
- scp - Secure Copy
- sftp - secure FTP

Husk: SSH er både navnet på protokollerne - version 1 og 2 samt programmet ssh til at logge ind på andre systemer

SSH tillader også port-forward, tunnel til usikre protokoller, eksempelvis X protokollen til UNIX grafiske vinduer

NB: Man bør idag bruge SSH protokol version 2!

SSH nøgler



I praksis benytter man nøgler fremfor kodeord

I kan lave jeres egne SSH nøgler med programmerne i Putty

Hvilken del skal jeg have for at kunne give jer adgang til en server?

Hvordan får jeg smartest denne nøgle?

Installation af SSH nøgle



Vi bruger login med password på kurset, men for fuldstændighedens skyld beskrives her hvordan nøgle installeres:

- først skal der genereres et nøglepar **id_dsa** og **id_dsa.pub**
- Den offentlige del, filen id_dsa.pub, kopieres til serveren
- Der logges ind på serveren
- Der udføres følgende kommandoer:

```
$ cd skift til dit hjemmekatalog
```

```
$ mkdir .ssh lav et katalog til ssh-nøgler
```

```
$ cat id_dsa.pub >> .ssh/authorized_keys kopierer nøglen
```

```
$ chmod -R go-rwx .ssh skift rettigheder på nøglen
```

OpenSSH konfiguration



Sådan anbefaler jeg at konfigurere OpenSSH SSHD

Det gøres i filen `sshd_config` typisk `/etc/ssh/sshd_config`

```
Port 22780    // eller anden tilfældig port
Protocol 2
```

```
PermitRootLogin no
PubkeyAuthentication yes
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
```

```
UseDNS no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
```

Det er en smagssag om man vil tillade *X11 forwarding*



Sikkerhed i netværket

RFC-2401 Security Architecture for the Internet Protocol

RFC-2402 IP Authentication Header (AH)

RFC-2406 IP Encapsulating Security Payload (ESP)

RFC-2409 The Internet Key Exchange (IKE) - dynamisk keying

Både til IPv4 og IPv6

MANDATORY i IPv6! - et krav hvis man implementerer fuld IPv6 support

god præsentation på <http://www.hsc.fr/presentations/ike/>

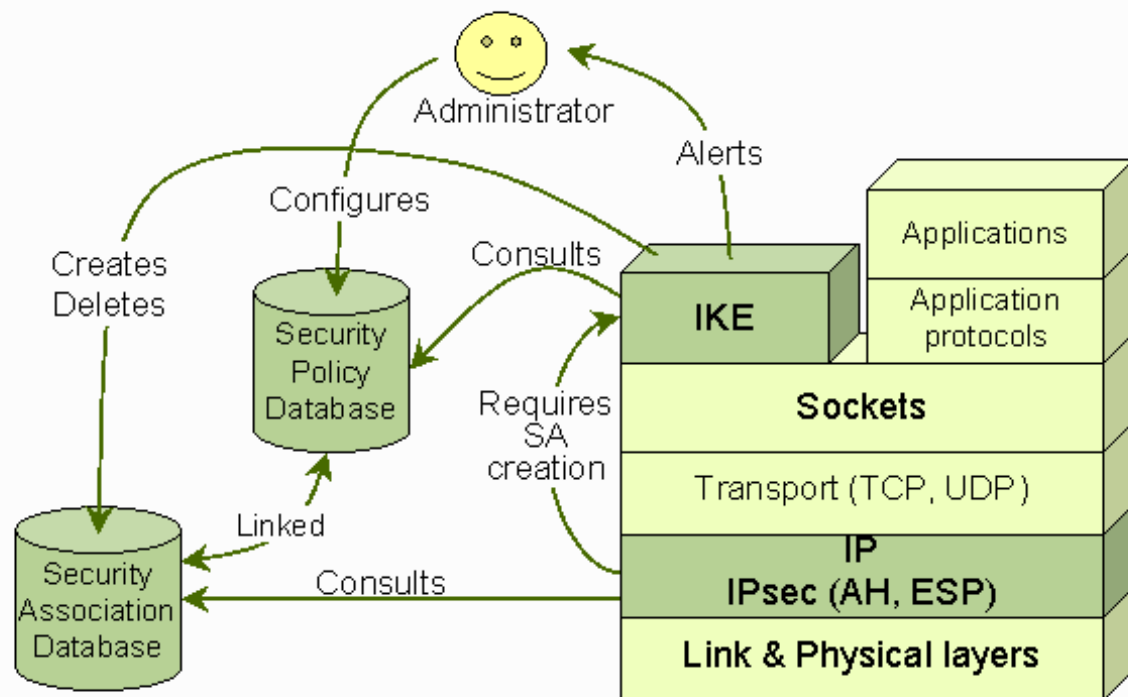
Der findes IKEscan til at scanne efter IKE porte/implementationer

<http://www.nta-monitor.com/ike-scan/index.htm>

IPsec er ikke simpelt!



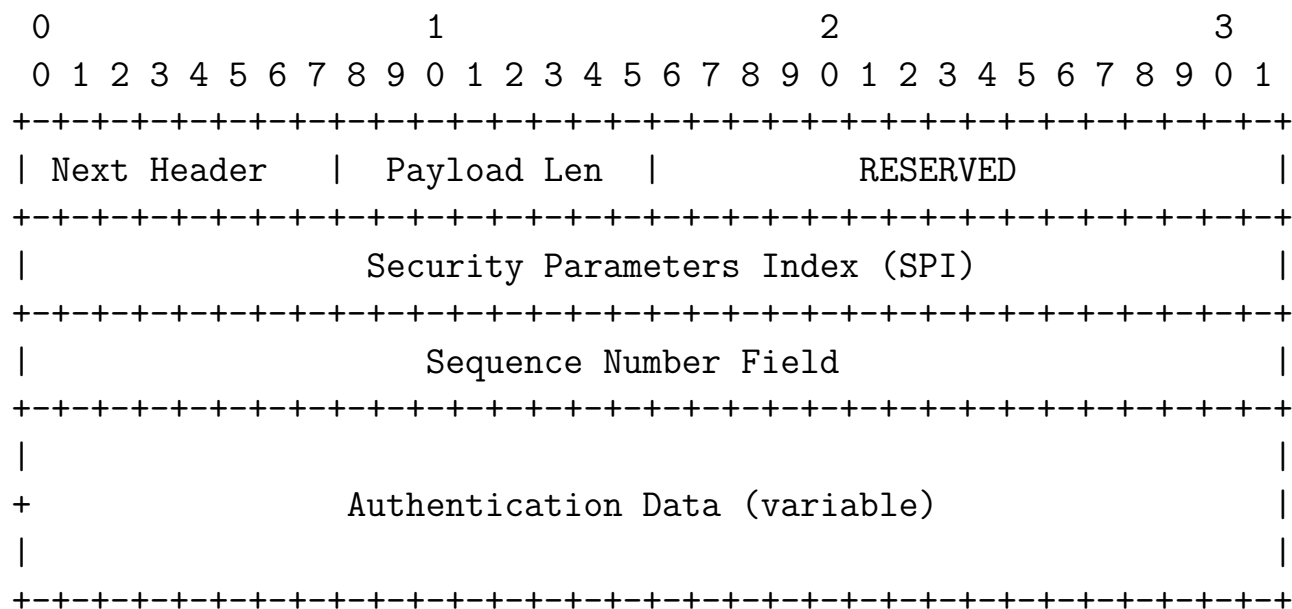
Interactions Between IKE and the IPsec Mechanisms



Kilde: <http://www.hsc.fr/presentations/ike/>



RFC-2402 IP AH



RFC-2402 IP AH



Indpakning - pakkerne før og efter Authentication Header:

BEFORE APPLYING AH

```
-----  
IPv4 |orig IP hdr |   |   |  
      |(any options)| TCP | Data |  
-----
```

AFTER APPLYING AH

```
-----  
IPv4 |orig IP hdr |   |   |   |  
      |(any options)| AH | TCP | Data |  
-----  
|<----- authenticated ----->|  
      except for mutable fields
```

RFC-2406 IP ESP



Pakkerne før og efter:

BEFORE APPLYING ESP

```
-----  
IPv6  |          | ext hdrs |          |  
      | orig IP hdr |if present| TCP | Data |  
-----
```

AFTER APPLYING ESP

```
-----  
IPv6  | orig |hop-by-hop,dest*,|   |dest|   |   | ESP   | ESP|  
      |IP hdr|routing,fragment.|ESP|opt*|TCP|Data|Trailer|Auth|  
-----  
                        |<---- encrypted ---->|  
                        |<---- authenticated ---->|
```

ipsec konfigurationsfiler



Der er følgende filer tilgængelige

- konfigurationsfiler i NetBSD/FreeBSD/Mac OS X format - med `setkey` kommandoen
- konfigurationsfil til OpenBSD server - med `ipsecadm` kommandoen

World Wide Web fødes



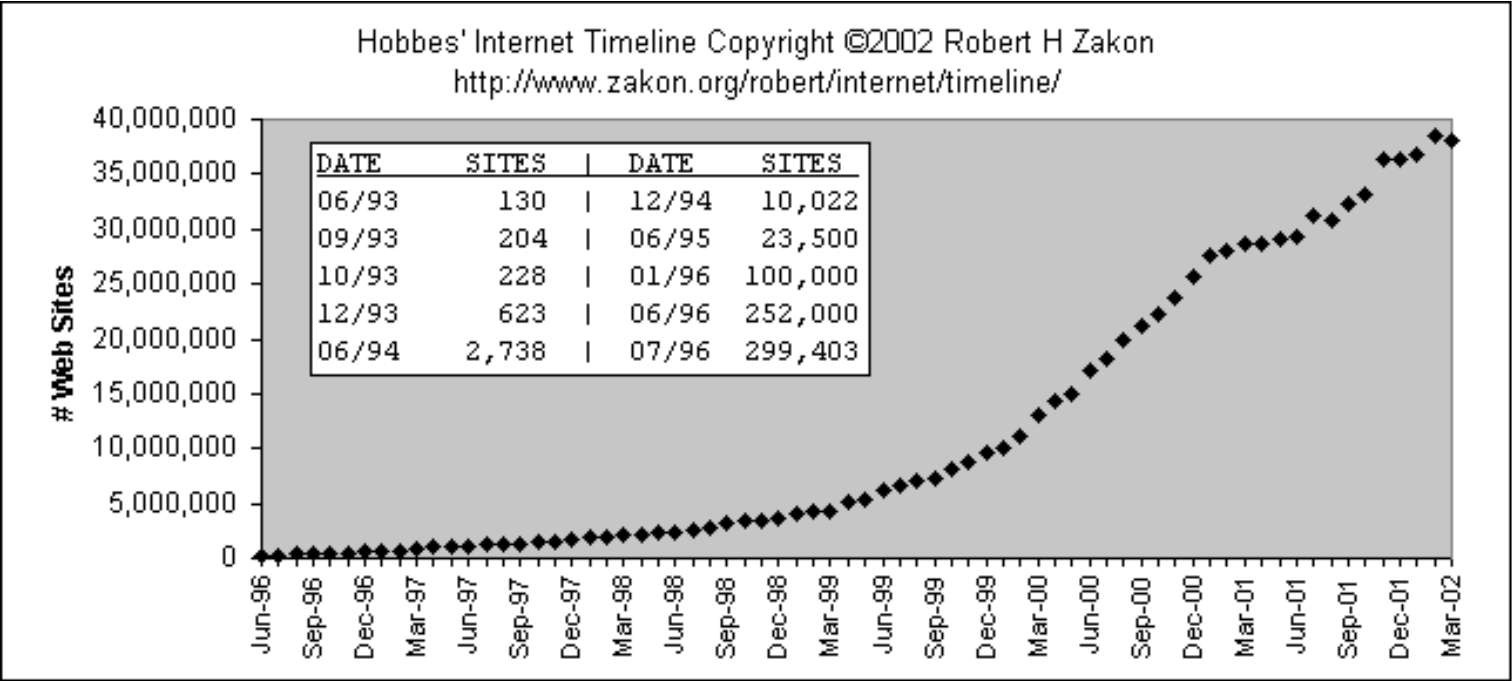
Tim Berners-Lee opfinder WWW 1989 og den første webbrowser og server i 1990 mens han

arbejder for CERN

Kilde: <http://www.w3.org/People/Berners-Lee/>



World Wide Web udviklingen



Udviklingen på world wide web bliver en stor kommerciel success

Kilde: Hobbes Internet time-line

<http://www.zakon.org/robert/internet/timeline/>



Nogle HTTP og webrelaterede RFC'er



- 1945 Hypertext Transfer Protocol – HTTP/1.0. T. Berners-Lee, R. Fielding, H. Frystyk. May 1996.
- 2068 Hypertext Transfer Protocol – HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. January 1997. (Obsoleted by RFC2616)
- 2069 An Extension to HTTP : Digest Access Authentication. J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart. January 1997. (Obsoleted by RFC2617)
- 2145 Use and Interpretation of HTTP Version Numbers. J. C. Mogul, R. Fielding, J. Gettys, H. Frystyk. May 1997.
- 2518 HTTP Extensions for Distributed Authoring – WEBDAV. Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen. February 1999.
- 2616 Hypertext Transfer Protocol – HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999. (Obsoletes RFC2068) (Updated by RFC2817)
- 2818 HTTP Over TLS. E. Rescorla. May 2000.

HTTP er basalt set en sessionsløs protokol bestående at individuelle HTTP forespørgsler via TCP forbindelser

Infokager og state management



- 2109 HTTP State Management Mechanism. D. Kristol, L. Montulli. February 1997. (Format: TXT=43469 bytes) (Obsoleted by RFC2965) (Status: PROPOSED STANDARD)
- 2965 HTTP State Management Mechanism. D. Kristol, L. Montulli. October 2000. (Format: TXT=56176 bytes) (Obsoletes RFC2109) (Status: PROPOSED STANDARD)

1. ABSTRACT This document specifies a way to create a stateful session with HTTP requests and responses. It describes two new headers, Cookie and Set-Cookie, which carry state information between participating origin servers and user agents. The method described here differs from Netscape's Cookie proposal, but it can interoperate with HTTP/1.0 user agents that use Netscape's method. (See the HISTORICAL section.)

(Citatet er fra RFC-2109)

Transport Layer Security



For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Most days have about 100 pages or less, but one day has 4 chapters to read!

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools