

---

Velkommen til

# TCP/IP grundkursus

Januar 2008

Henrik Lund Kramshøj  
hlk@security6.net

<http://www.security6.net>

# Kursusmateriale

Dette materiale består af flere dele:

- Kursusmaterialet - præsentationen til undervisning - dette sæt
- Øvelseshæfte med øvelser

Hertil kommer diverse ressourcer fra internet, specielt RFC-dokumenter

Boot CD'er baseret på Linux

Bemærk: kursusmaterialet er ikke en substitut for andet materiale, der er udeladt mange detaljer som forklares undervejs, eller kan slås op op internet

# Kursusforløb

Vi skal have glæde af hinanden i følgende kursusforløb

- 5 dage TCP/IP grundkursus

I skal uddover at lære en masse om protokoller og netværk lære nogle gode vaner!

Jeres arbejde med netværk kanlettes betydeligt - hvis I starter rigtigt!

# Kontaktinformation og profil



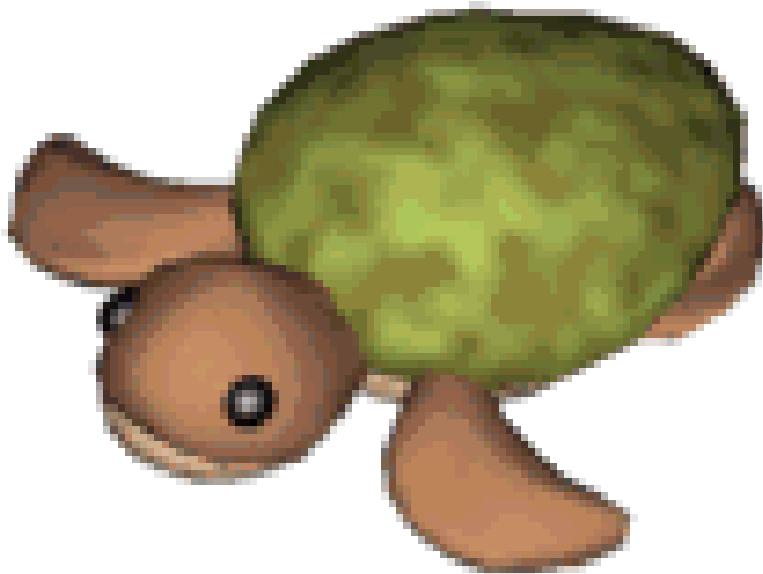
- Henrik Lund Kramshøj
- Mobil: 2026 6000
- Email: [hlk@security6.net](mailto:hlk@security6.net)
- Uddannet cand.scient fra DIKU
- Selvstændig sikkerhedskonsulent siden januar 2003

I er velkomne til at stille spørgsmål efter kurset er slut

# Plan for dagene

- 09:00 - Workshop 1
- 10:00 - Workshop 2
- 11:00 - Workshop 3
- 12:00 - Frokost
- 12:45 - Workshop 4
- 14:00 - Workshop 5
- 15:00 - Workshop 6
- 16:00 - 1630 dagen er slut\*
- \*) Sluttiden kan variere.
- Hver time afsluttes med en kort pause med mulighed for kaffe, te, vand, rygning, ...
- Der vil være praktiske opgaver fordelt henover dagene til at få praktisk erfaring.

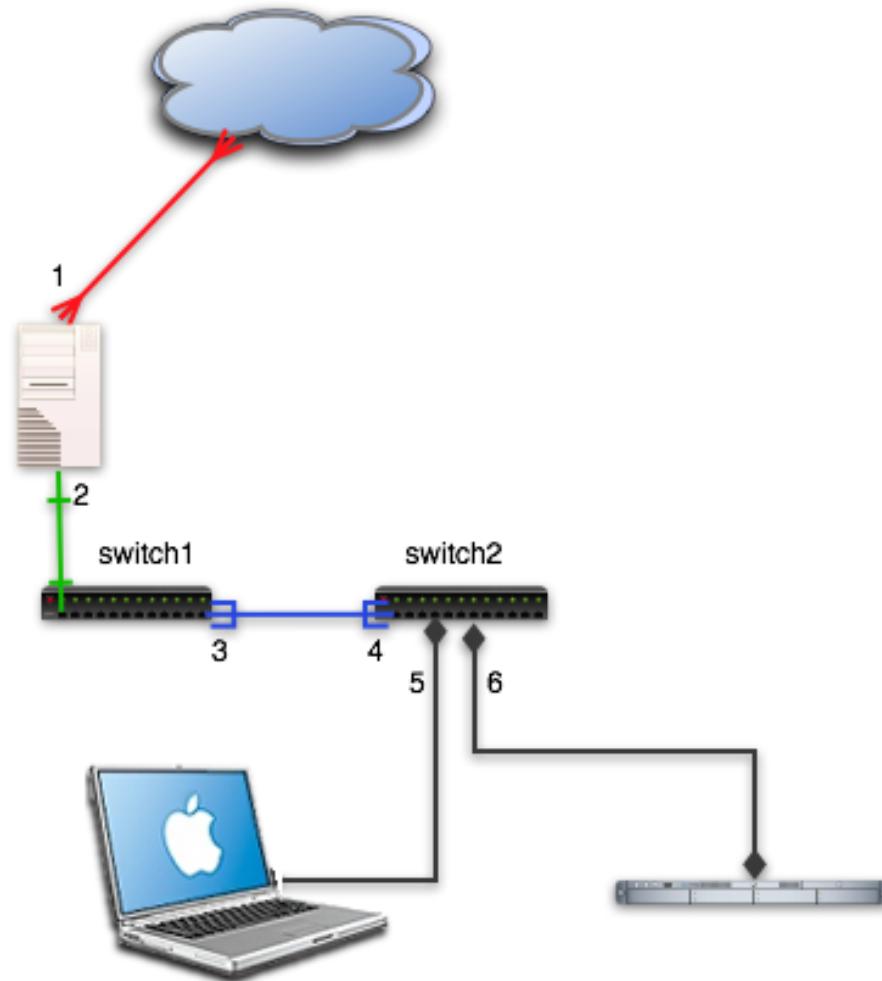
# Er TCP/IP interessant?



IP er med i alle de gængse operativsystemer UNIX og Windows

Internet er overalt

# Formål: TCP/IP grundkursus



IP-baserede netværk

# Formål: mere specifikt

At introducere IP familien af protokoller

Kendskab til almindeligt brugte programmer i disse miljøer

- ping, traceroute, samt serverfunktioner Apache HTTP, BIND DNS m.v.

Gennemgang af netværksdesign ved hjælp af almindeligt brugte setups

- en skalamodel af internet

# Forudsætninger

Dette er en workshop og fuldt udbytte kræver at deltagerne udfører praktiske øvelser

Kurset anvender OpenBSD til øvelser, men UNIX kendskab er ikke nødvendigt

De fleste øvelser kan udføres fra en Windows PC

Øvelserne foregår via login til UNIX maskinen

- Til penetrationstest og det meste Internet-sikkerhedsarbejde er der følgende forudsætninger
- Netværkserfaring
- TCP/IP principper - ofte i detaljer
- Programmeringserfaring er en fordel
- UNIX kendskab er ofte en **nødvendighed**
  - fordi de nyeste værktøjer er skrevet til UNIX i form af Linux og BSD

# Kursusfaciliteter

Der er opbygget et kursusnetværk med følgende primære systemer:

- UNIX server Fiona med HTTP server og værktøjer
- UNIX boot CD'er eller VMware images - jeres systemer

På UNIX serveren tillades login diverse kursusbrugere - kursus1, kursus2, kursus3, ...  
kodeordet er **kursus**

Det er en fordel at benytte hver sin bruger, så man kan gemme scripts

På de resterende systemer kan benyttes brugeren **kursus**

Login: **kursus**

Password: **kursus42**

# Knoppix og BackTrack boot CD'er

Vi bruger UNIX og SSH på kurset

I kan bruge en udleveret CD til at boote Linux på jeres arbejdsstation og derfra arbejde, eller I kan benytte Fiona

Brug CD'en eller VMware player til de grafiske værktøjer som Wireshark

CD'en er under en åben licens - må kopieres frit :-)

ISO image kan hentes fra mirrors

BackTrack <http://www.remote-exploit.org/backtrack.html>

Til begyndere indenfor Linux anbefales Ubuntu eller Kubuntu til arbejdsstationer

# Stop - tid til check



Er alle kommet

Har alle en PC med

Har alle et kabel eller trådløst netkort som virker

Der findes et trådløst netværk ved navn **kamenet**

Mangler der strømkabler

Mangler noget af ovenstående, sæt nogen igang med at finde det

Da UNIX indgår er her et lille *cheat sheet* til UNIX

- DOS/Windows kommando - tilsvarende UNIX, og forklaring
- dir - ls - står for list files, viser filnavne
- del - rm - står for remove, sletter filer
- cd - cd - change directory, skifter katalog
- type - cat - concatenate, viser indholdet af tekstfiler
- more - less - viser tekstfiler en side af gangen
- attrib - chmod - change mode, ændrer rettighederne på filer

Prøv bare:

- **ls** list, eller long listing med **ls -l**
- **cat /etc/hosts** viser hosts filen
- **chmod +x head.sh** - sæt execute bit på en fil så den kan udføres som et program med kommandoen **./head.sh**

# Aftale om test af netværk

**Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 6 måneder straffes den, som uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et anlæg til elektronisk databehandling.**

Hacking kan betyde:

- At man skal betale erstatning til personer eller virksomheder
- At man får konfiskeret sit udstyr af politiet
- At man, hvis man er over 15 år og bliver dømt for hacking, kan få en bøde - eller fængselsstraf i alvorlige tilfælde
- At man, hvis man er over 15 år og bliver dømt for hacking, får en plettet straffeattest. Det kan give problemer, hvis man skal finde et job eller hvis man skal rejse til visse lande, fx USA og Australien
- Frit efter: <http://www.stophacking.dk> lavet af Det Kriminalpræventive Råd
- Frygten for terror har forstærket ovenstående - så lad være!

# Agenda - dag 1 Basale begreber og mindre netværk



Opstart - hvad er IP og TCP/IP

Adresser

Subnets og CIDR

TCP og UDP

Basal DNS

Lidt om hardware half/full-duplex

# Agenda - dag 2 IPv6, Management, diagnosticering



IP version 6

ARP og NDP

Ping

Traceroute

Snifferprogrammer Tcpdump og Wireshark

Management

Tuning og perfomancemålninger

RRDTool og Smokeping

Overvågning og Nagios

Wireless 802.11

# Agenda - dag 3 Dynamiske protokoller og services



Netværksservices og serverfunktioner

DNS protokoller og servere

HTTP protokoller og servere

Dynamisk routing: BGP og OSPF

Produktionsmodning af netværk

Netværksprogrammering: små utilityprogrammer og scripts

# Agenda - dag 4 Netværkssikkerhed og firewalls



SSL Secure Sockets Layer

VLAN 802.1q

802.1x portbaseret autentifikation

WPA Wi-Fi Protected Access

VPN protokoller og IPSec

VoIP introduktion

Mobile IP introduktion

# Agenda - dag 5 Netværksdesign og templates

Netværksdesign

Infrastrukturer i praksis

Templates til almindeligt forekommende setups

Afslutning og opsummering på kursus

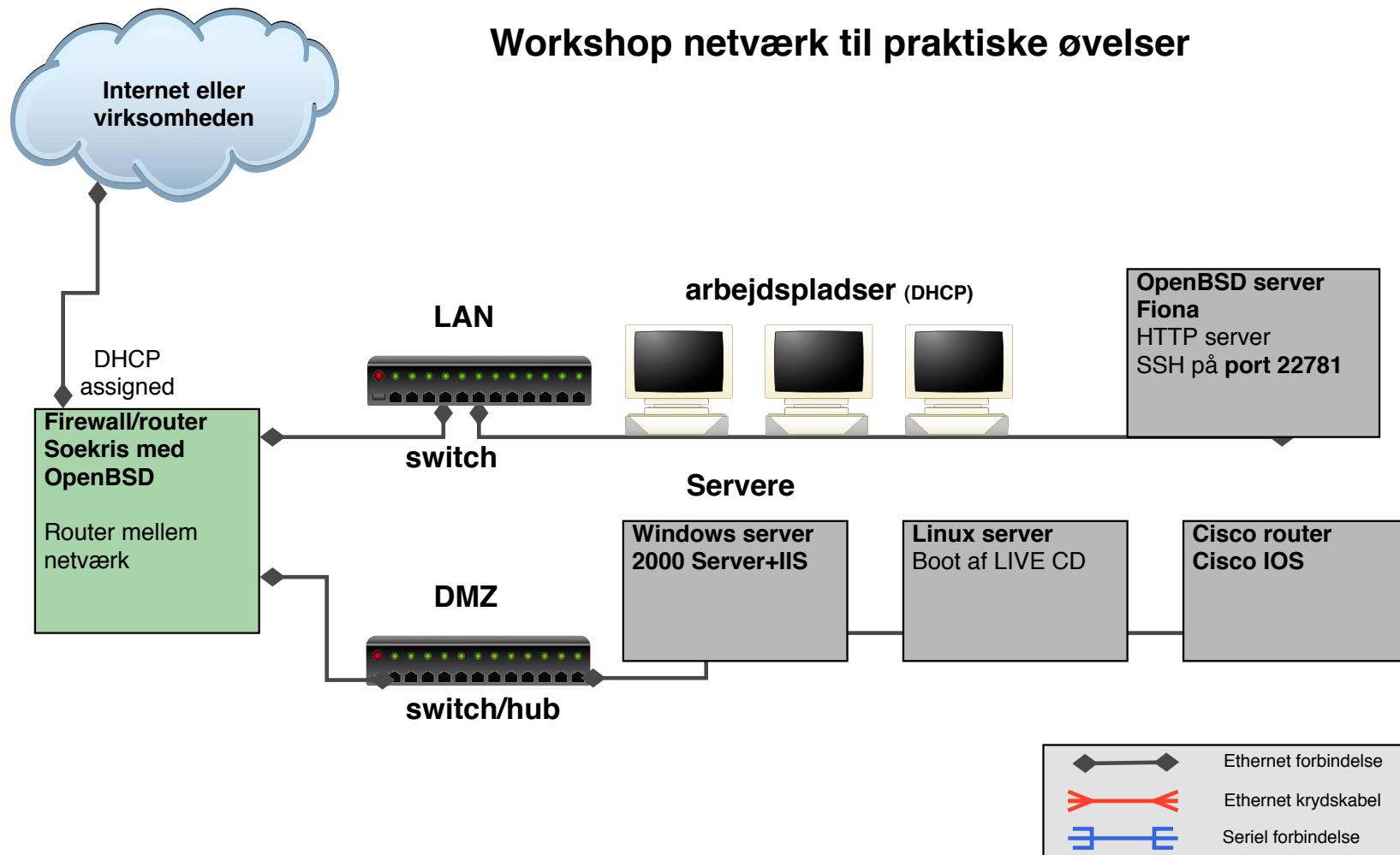
Udfyld meget gerne evalueringsskemaerne, tak

# Dag 1 Basale begreber og mindre netværk

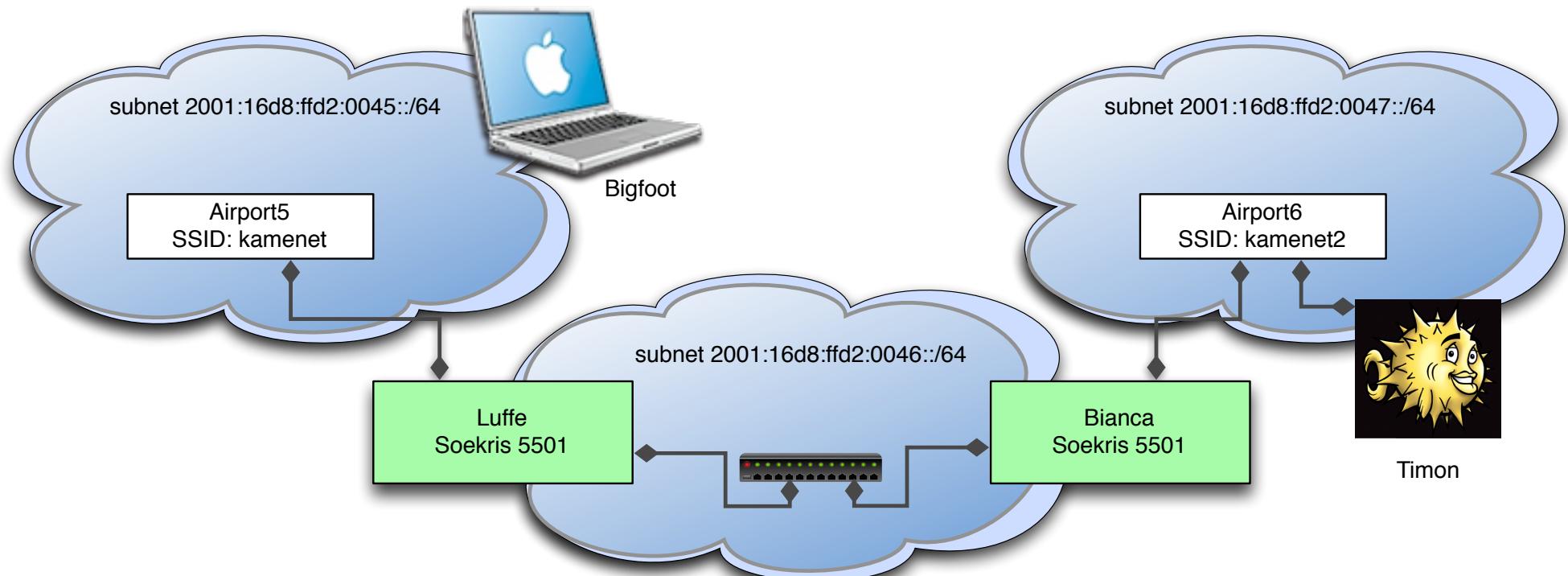


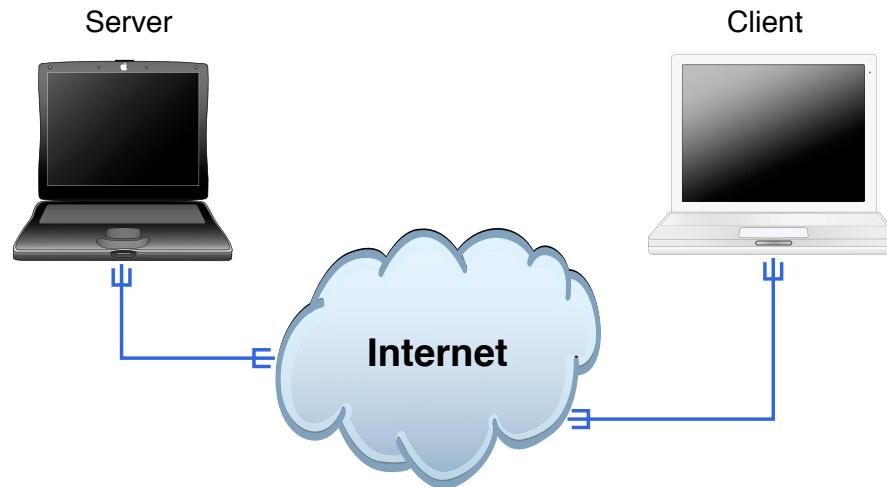
Security  
.net

## Workshop netværk til praktiske øvelser



# Netværk til routning





Klienter og servere

Rødder i akademiske miljøer

Protokoller der er op til 20 år gamle

Meget lidt kryptering, mest på http til brug ved e-handel

Kurset omhandler udelukkende netværk baseret på IP protokollerne

# Internet er åbne standarder!

We reject kings, presidents, and voting.  
We believe in rough consensus and running code.  
– The IETF credo Dave Clark, 1992.

Request for comments - RFC - er en serie af dokumenter

RFC, BCP, FYI, informational  
de første stammer tilbage fra 1969

Ændres ikke, men får status Obsoleted når der udkommer en nyere version af en standard

Standards track:

Proposed Standard → Draft Standard → Standard

Åbne standarder = åbenhed, ikke garanti for sikkerhed

Kommunikation mellem mennesker!

Baseret på TCP/IP

- best effort
- packet switching (IPv6 kalder det packets, ikke datagram)
- forbindelsesorienteret, *connection-oriented*
- forbindelsesløs, *connection-less*

RFC-1958:

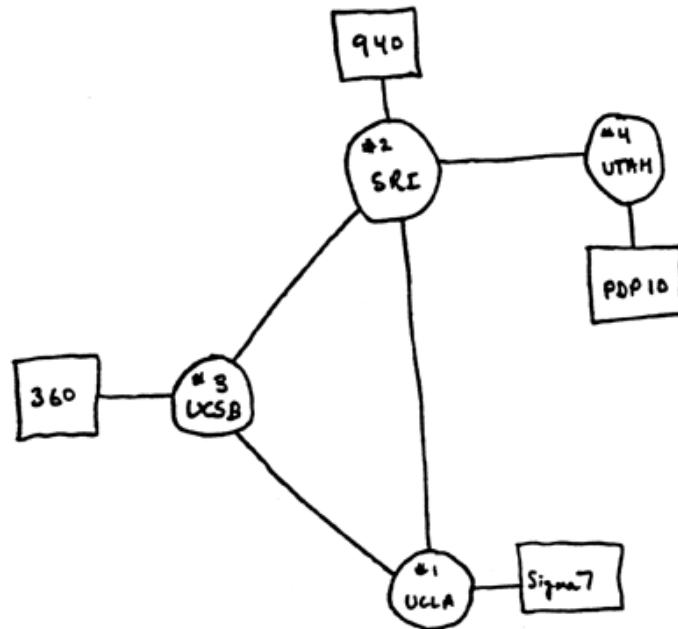
A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology.

# IP netværk: Internettet historisk set



- 1961 L. Kleinrock, MIT packet-switching teori
- 1962 J. C. R. Licklider, MIT - notes
- 1964 Paul Baran: On Distributed Communications
- 1969 ARPANET startes 4 noder
- 1971 14 noder
- 1973 Arbejde med IP startes
- 1973 Email er ca. 75% af ARPANET traffik
- 1974 TCP/IP: Cerf/Kahn: A protocol for Packet Network Interconnection
- 1983 EUUG → DKUUG/DIKU forbindelse
- 1988 ca. 60.000 systemer på Internettet The Morris Worm rammer ca. 10%
- 2000 Maj I LOVE YOU ormen rammer
- 2002 Ialt ca. 130 millioner på Internet

# Internet historisk set - anno 1969



- Node 1: University of California Los Angeles
- Node 2: Stanford Research Institute
- Node 3: University of California Santa Barbara
- Node 4: University of Utah

L. Kleinrock *Information Flow in Large Communication nets*, 1961

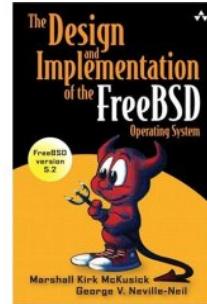
J.C.R. Licklider, MIT noter fra 1962 *On-Line Man Computer Communication*

Paul Baran, 1964 *On distributed Communications* 12-bind serie af rapporter  
<http://www.rand.org/publications/RM/baran.list.html>

V. Cerf og R. Kahn, 1974 *A protocol for Packet Network Interconnection* IEEE Transactions on Communication, vol. COM-22, pp. 637-648, May 1974

De tidlige notater kan findes på nettet!

Læs evt. mere i mit speciale <http://www.inet6.dk/thesis.pdf>



UNIX kildeteksten var nem at få fat i for universiteter og mange andre

Bell Labs/AT&T var et telefonselskab - ikke et software hus

På Berkeley Universitetet blev der udviklet en del på UNIX og det har givet anledning til en hel gren kaldet BSD UNIX

BSD står for Berkeley Software Distribution

BSD UNIX har blandt andet resulteret i virtual memory management og en masse TCP/IP relaterede applikationer

# Open Source definitioner - uddrag



Free Redistribution - der må ikke lægges begrænsninger på om softwaren gives væk eller sælges

Source Code - kildeteksten skal være tilgængelig

Derived Works - det skal være muligt at arbejde videre på

Integrity of The Author's Source Code - det skal være muligt at beskytte sit navn og rygte, ved at kræve ændret navn for afledte projekter

Softwareen kaldes ofte også Free Software, nogle bruger endda Libre

Eksempler er BSD licensen, Apache, GNU GPL og mange andre

Kilder: <http://www.opensource.org/>

<http://en.wikipedia.org/wiki/FLOSS> Free/Libre/Open-Source Software

## BSD licensen er pragmatisk

Security

.net

BSD licensen kræver ikke at man offentliggør sine ændringer, man kan altså bruge BSD kildetekst og stadig lave et kommersIELT produkt!

GNU GPL bliver af nogle omtalt som en virus - der *inficerer* softwaren, og afledte projekter

# Hvad er Internet

80'erne IP/TCP starten af 80'erne

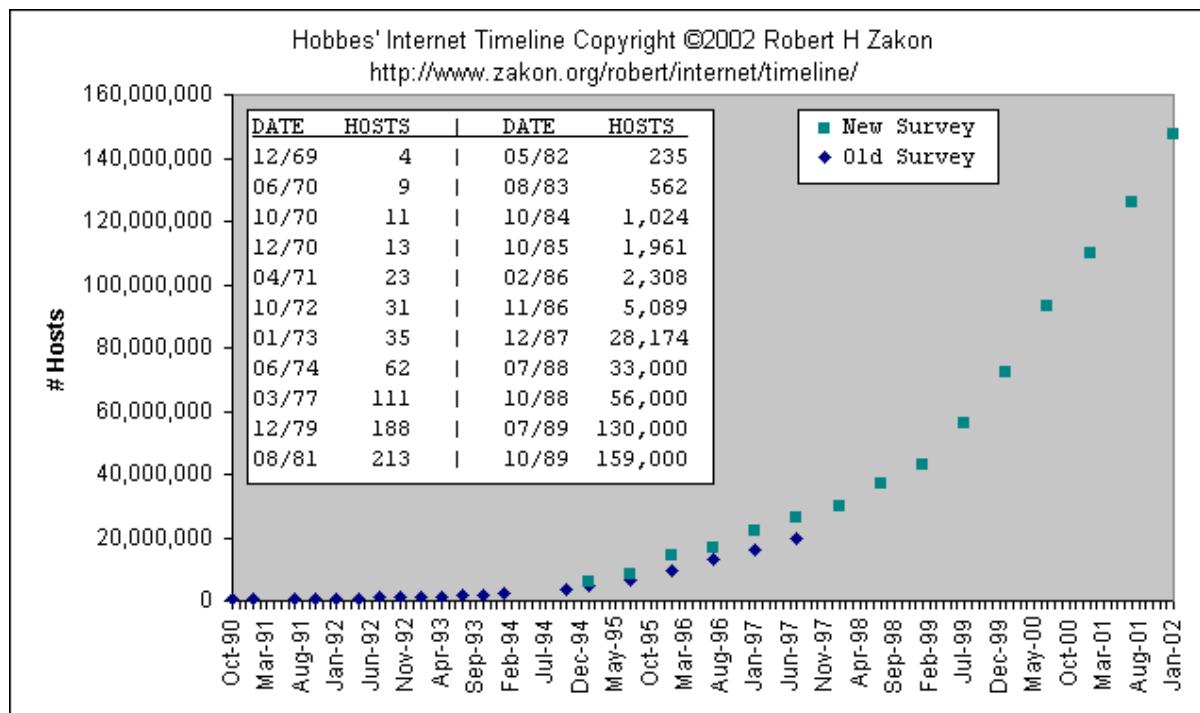
90'erne IP version 6 udarbejdes

- IPv6 ikke brugt i Europa og US
- IPv6 er ekstremt vigtigt i Asien
- historisk få adresser tildelt til 3.verdenslande
- Større Universiteter i USA har ofte større allokering end Kina!

1991 WWW "opfindes" af Tim Berners-Lee hos CERN

E-mail var hovedparten af traffik - siden overtog web/http førstepladsen

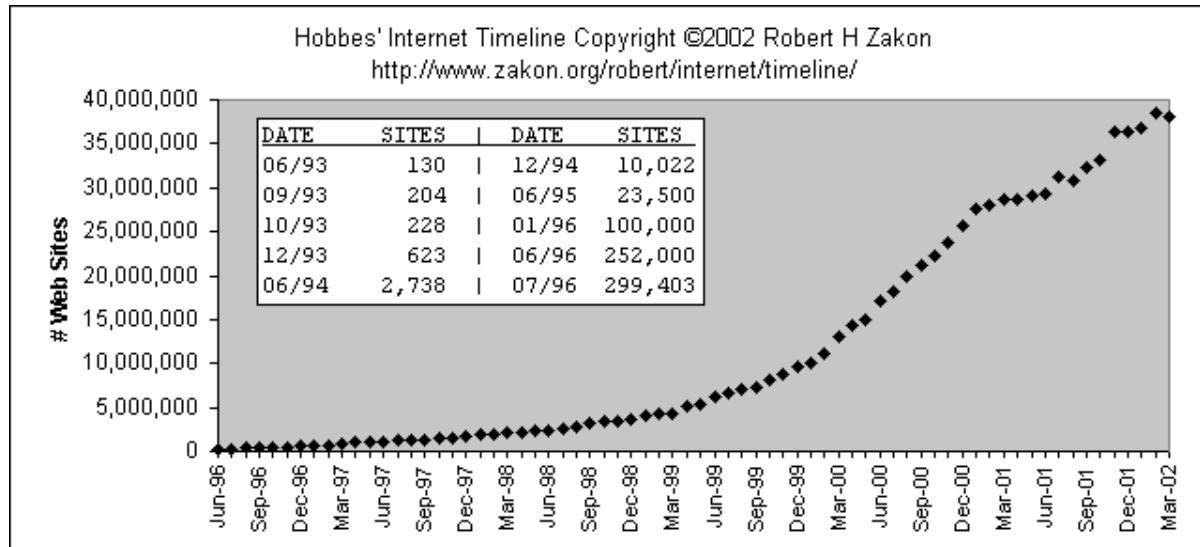
## Antallet af hosts på Internet



Kilde: Hobbes' Internet Timeline v5.6

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

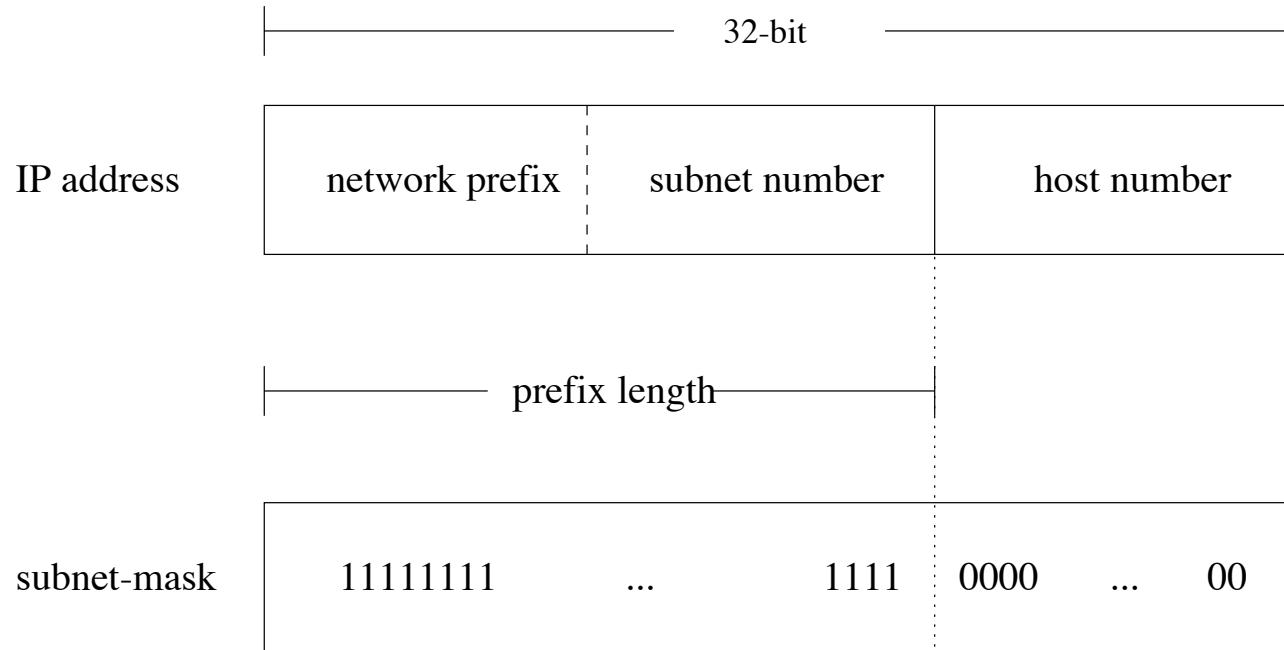
## Antallet af World Wide Web servere



Kilde: Hobbes' Internet Timeline v5.6

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

# Fælles adresserum



Hvad kendetegner internet idag

Der er et fælles adresserum baseret på 32-bit adresser

En IP-adresse kunne være 10.0.0.1

# IPv4 addresser og skrivemåde

```
hlk@bigfoot:hlk$ ipconvert.pl 127.0.0.1
Adresse er: 127.0.0.1
Adresse er: 2130706433
hlk@bigfoot:hlk$ ping 2130706433
PING 2130706433 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.135 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.144 ms
```

IP-adresser skrives typisk som decimaltal adskilt af punktum

Kaldes **dot notation**: 10.1.2.3

Kan også skrive som oktal eller heksadecimale tal

# IP-adresser som bits

IP-adresse: 127.0.0.1

Heltal: 2130706433

Binary: 11111110000000000000000000000001

IP-adresser kan også konverteres til bits

Computeren regner binært, vi bruger dot-notationen

Tidligere benyttede man klasseinddelingen af IP-adresser: A, B, C, D og E

Desværre var denne opdeling ufleksibel:

- A-klasse kunne potentielt indeholde 16 millioner hosts
- B-klasse kunne potentielt indeholder omkring 65.000 hosts
- C-klasse kunne indeholde omkring 250 hosts

Derfor bad de fleste om adresser i B-klasser - så de var ved at løbe tør!

D-klasse benyttes til multicast

E-klasse er blot reserveret

Se evt. [http://en.wikipedia.org/wiki/Classful\\_network](http://en.wikipedia.org/wiki/Classful_network)

# CIDR Classless Inter-Domain Routing

Classfull routing		Classless routing (CIDR)	
4 Class C networks	Inherent subnet mask	Supernet	Subnet mask
192.0.08.0	255.255.255.0	192.0.08.0	255.255.252.0 (252d=11111100b)
192.0.09.0	255.255.255.0		
192.0.10.0	255.255.255.0		
192.0.11.0	255.255.255.0		
		Base network/prefix 192.0.8.0/	

Subnetmasker var oprindeligt indforstået

Dernæst var det noget man brugte til at opdele sit A, B eller C net med

Ved at tildele flere C-klasser kunne man spare de resterende B-klasser - men det betød en routing table explosion

Idag er subnetmaske en sammenhængende række 1-bit der angiver størrelse på nettet

10.0.0.0/24 betyder netværket 10.0.0.0 med subnetmaske 255.255.255.0

Nogle få steder kaldes det tillige supernet, supernetting

# Subnet calculator, CIDR calculator

**Subnet Calculator**

Network Class A <input type="radio"/> B <input type="radio"/> C <input checked="" type="radio"/>	First Octet Range 192 – 223
IP Address 192 . 168 . 0 . 1	Hex IP Address C0.A8.00.01
Subnet Mask 255.255.255.0	Wildcard Mask 0.0.0.255
Subnet Bits 0	Mask Bits 24
Maximum Subnets 1	Hosts per Subnet 254
Host Address Range 192.168.0.1 – 192.168.0.254	
Subnet ID 192.168.0.0	Broadcast Address 192.168.0.255
Subnet Bitmap 110nnnn.nnnnnnnn.nnnnnnnn.hhhhhh	

Der findes et væld af programmer som kan hjælpe med at udregne subnetmasker til IPv4

Screenshot fra <http://www.subnet-calculator.com/>

# RFC-1918 private netværk

Der findes et antal adresserum som alle må benytte frit:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Address Allocation for Private Internets RFC-1918 adresserne!

NB: man må ikke sende pakker ud på internet med disse som afsender, giver ikke mening

# IPv4 addresser opsummering

- Altid 32-bit adresser
- Skrives typisk med 4 decimaltal dot notation 10.1.2.3
- Netværk angives med CIDR Classless Inter-Domain Routing RFC-1519
- CIDR notation 10.0.0.0/8 - fremfor 10.0.0.0 med subnet maske 255.0.0.0
- Specielle adresser
  - 127.0.0.1 localhost/loopback
  - 0.0.0.0 default route
- RFC-1918 angiver private adresser som alle kan bruge

## Stop - netværket idag

Bemærk hvilket netværk vi bruger idag

Primære server fiona har IP-adressen 10.0.45.36

Primære router luffe har IP-adressen 10.0.45.2 (og flere andre)

Sekundære router idag er Bianca som har IP-adressen 10.0.46.2 (og flere andre)

Hvis du kender til IP i forvejen så udforsk gerne på egen hånd netværket

Det er tilladt at logge ind på alle systemer, undtagen Henrik's laptop bigfoot :-)

### **Det er forbudt at ændre IP-konfiguration og passwords**

Nu burde I kunne forbinde jer til netværket fysisk, check med ping 10.0.45.2

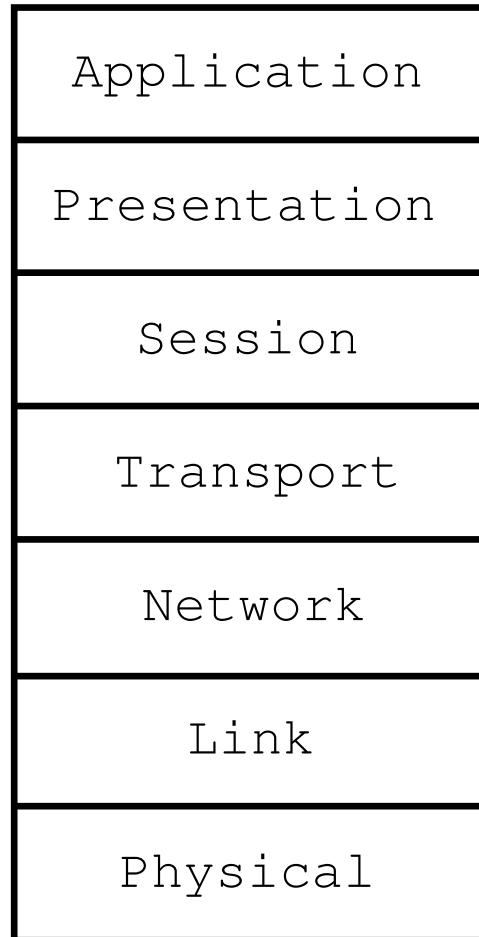
Det er nok at en PC i hver gruppe er på kursusnetværket

Pause for dem hvor det virker, mens vi ordner resten

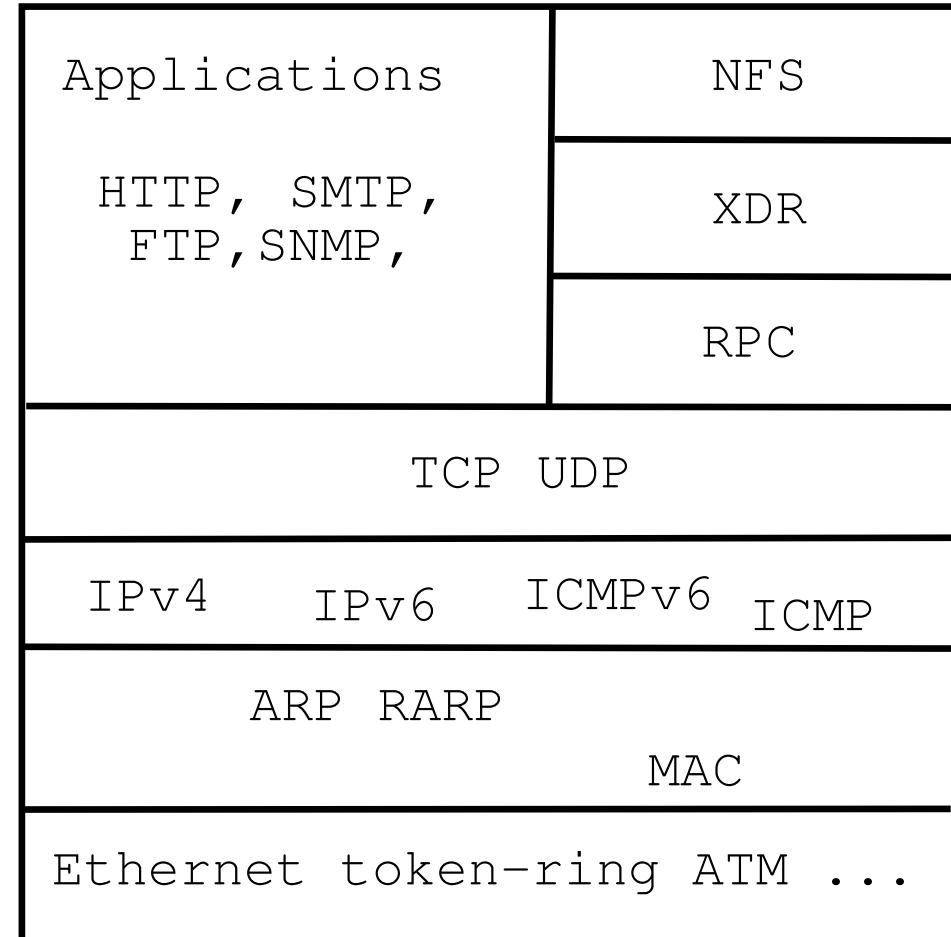
# OSI og Internet modellerne



OSI Reference Model



Internet protocol suite



Der er mange muligheder med IP netværk, IP kræver meget lidt

Ofte benyttede idag er:

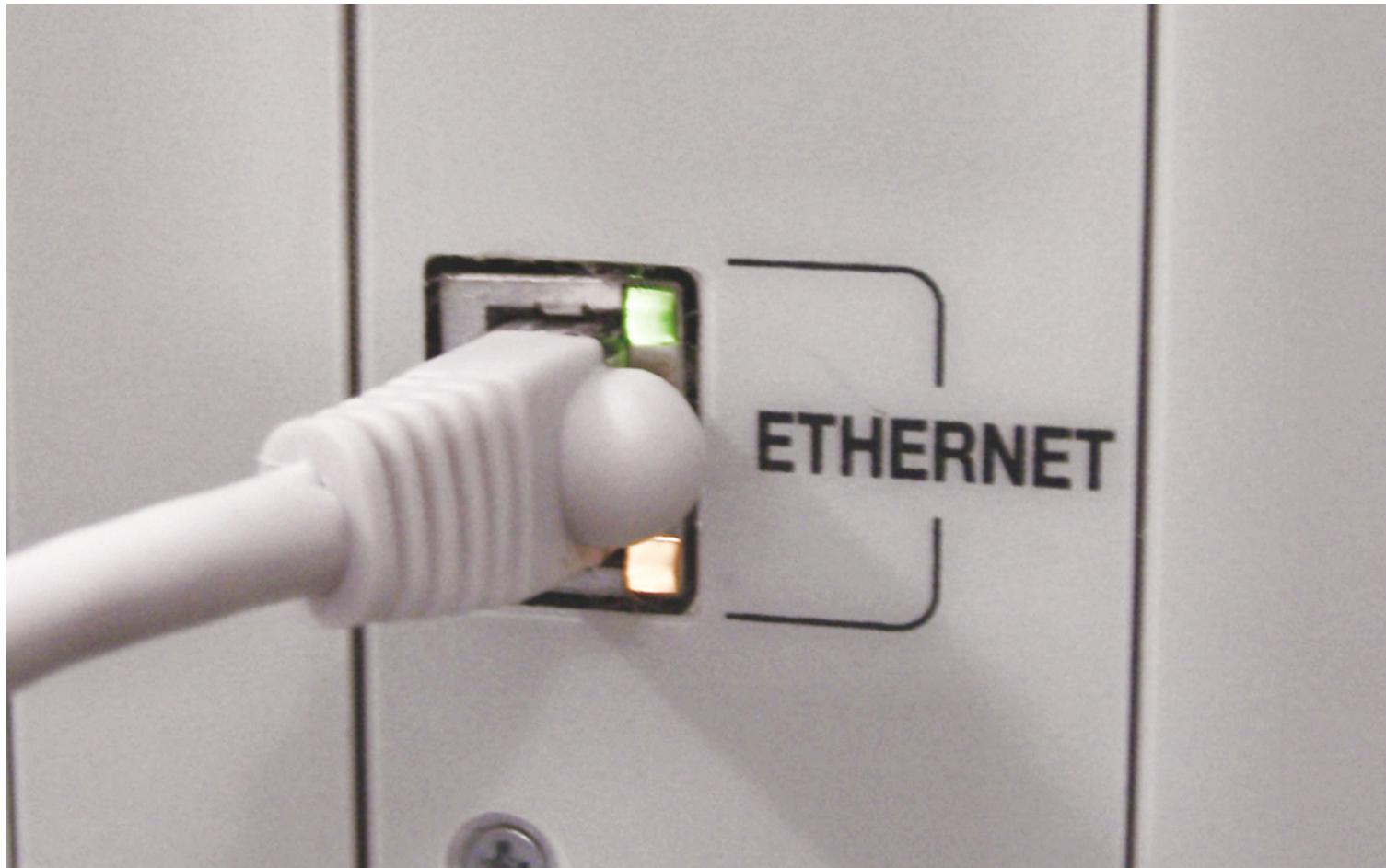
- Ethernet - varianter 10mbit, 100mbit, gigabit, 10 Gigabit findes, men er dyrt
- Wireless 802.11 teknologier
- ADSL/ATM teknologier til WAN forbindelser
- MPLS ligeledes til WAN forbindelser

Ethernet kan bruge kobberledninger eller fiber

WAN forbindelser er typisk fiber på grund af afstanden mellem routere

Tidligere benyttede inkluderer: X.25, modem, FDDI, ATM, Token-Ring

# Ethernet stik, kabler og dioder



Dioder viser typisk om der er link, hastighed samt aktivitet



Et typisk 802.11 Access-Point (AP) der har Wireless og Ethernet stik/switch

# MAC adresser

00-03-93	(hex)	Apple Computer, Inc.
000393	(base 16)	Apple Computer, Inc.
		20650 Valley Green Dr.
		Cupertino CA 95014
		UNITED STATES

Netværksteknologierne benytter adresser på lag 2

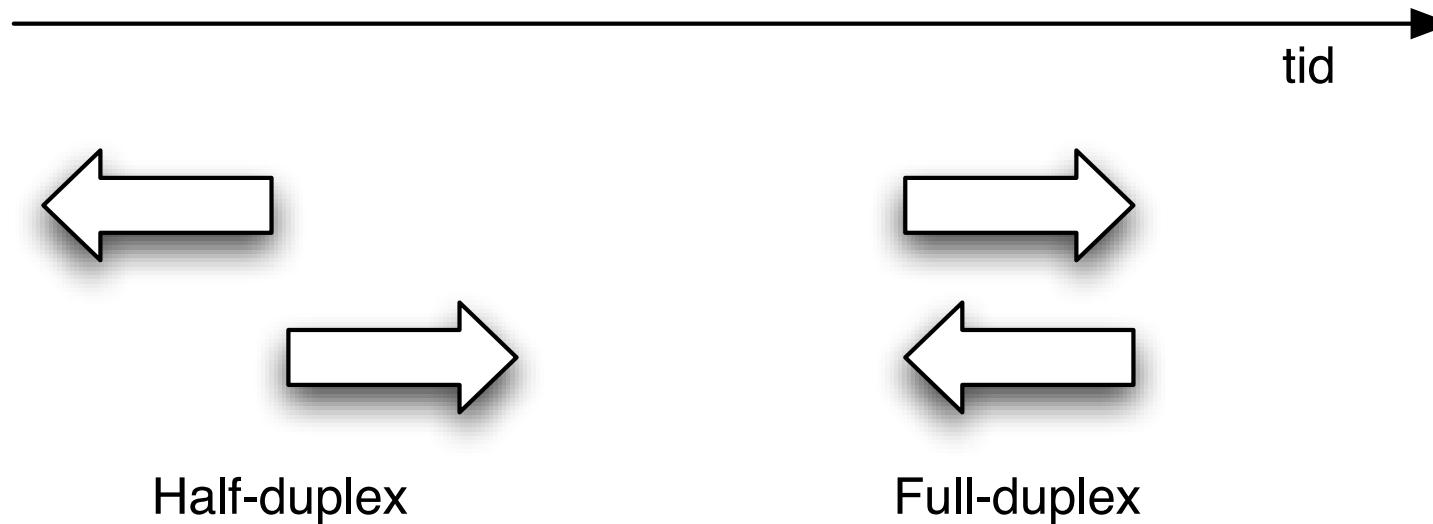
Typisk svarende til 48-bit MAC adresser som kendes fra Ethernet MAC-48/EUI-48

Første halvdel af adresserne er Organizationally Unique Identifier (OUI)

Ved hjælp af OUI kan man udlede hvilken producent der har produceret netkortet

<http://standards.ieee.org/regauth/oui/index.shtml>

# Half/full-duplex og speed



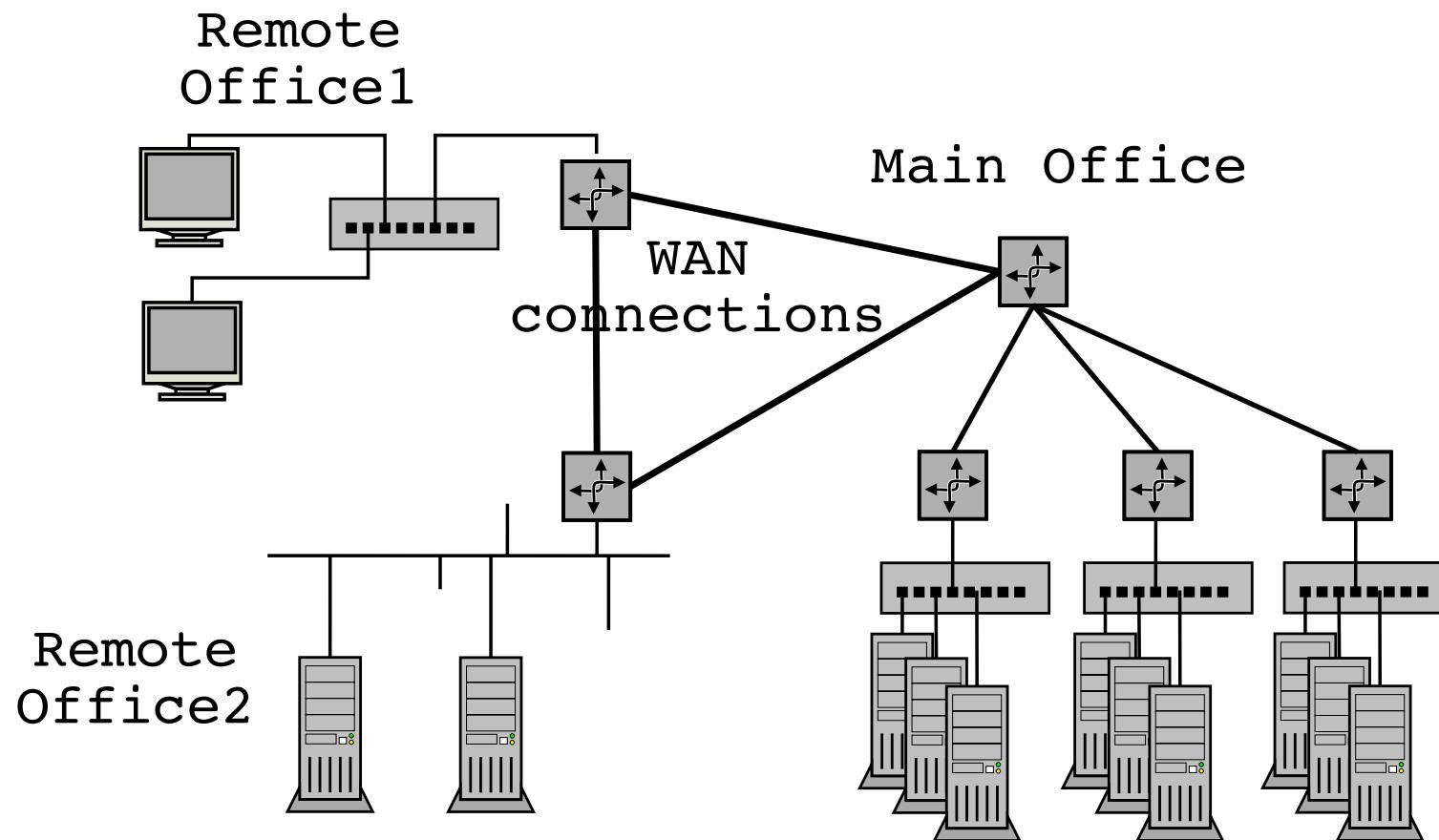
Hvad hastighed overføres data med?

De fleste nyere Ethernet netkort kan køre i fuld-duplex

med full-duplex kan der både sendes og modtages data samtidigt

Ethernet kan benytte auto-negotiation - der ofte virker

Klart bedre i gigabitnetkort men pas på



Fysisk er der en begrænsing for hvor lange ledningerne må være

Ethernet er broadcast teknologi, hvor data sendes ud på et delt medie - Æteren

Broadcast giver en grænse for udbredningen vs hastighed

Ved hjælp af en bro kan man forbinde to netværkssegmenter på layer-2

Broen kopierer data mellem de to segmenter

Virker som en forstærker på signalet, men mere intelligent

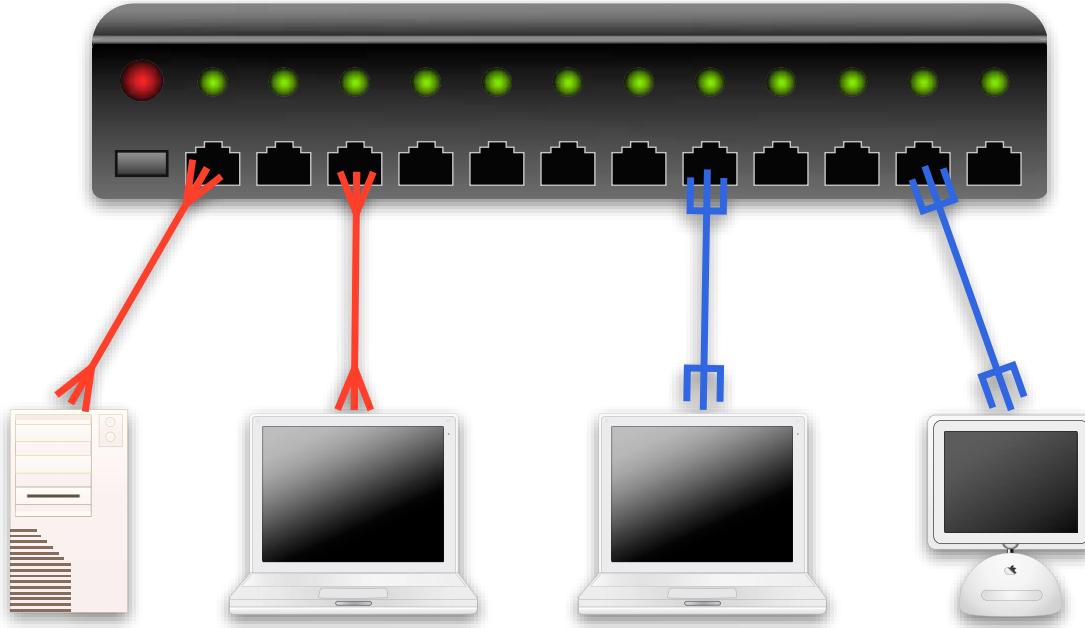
Den intelligente bro kender MAC adresserne på hver side

Broen kopierer kun hvis afsender og modtager er på hver sin side

Kilde: For mere information søger efter Aloha-net

<http://en.wikipedia.org/wiki/ALOHAnet>

# En switch

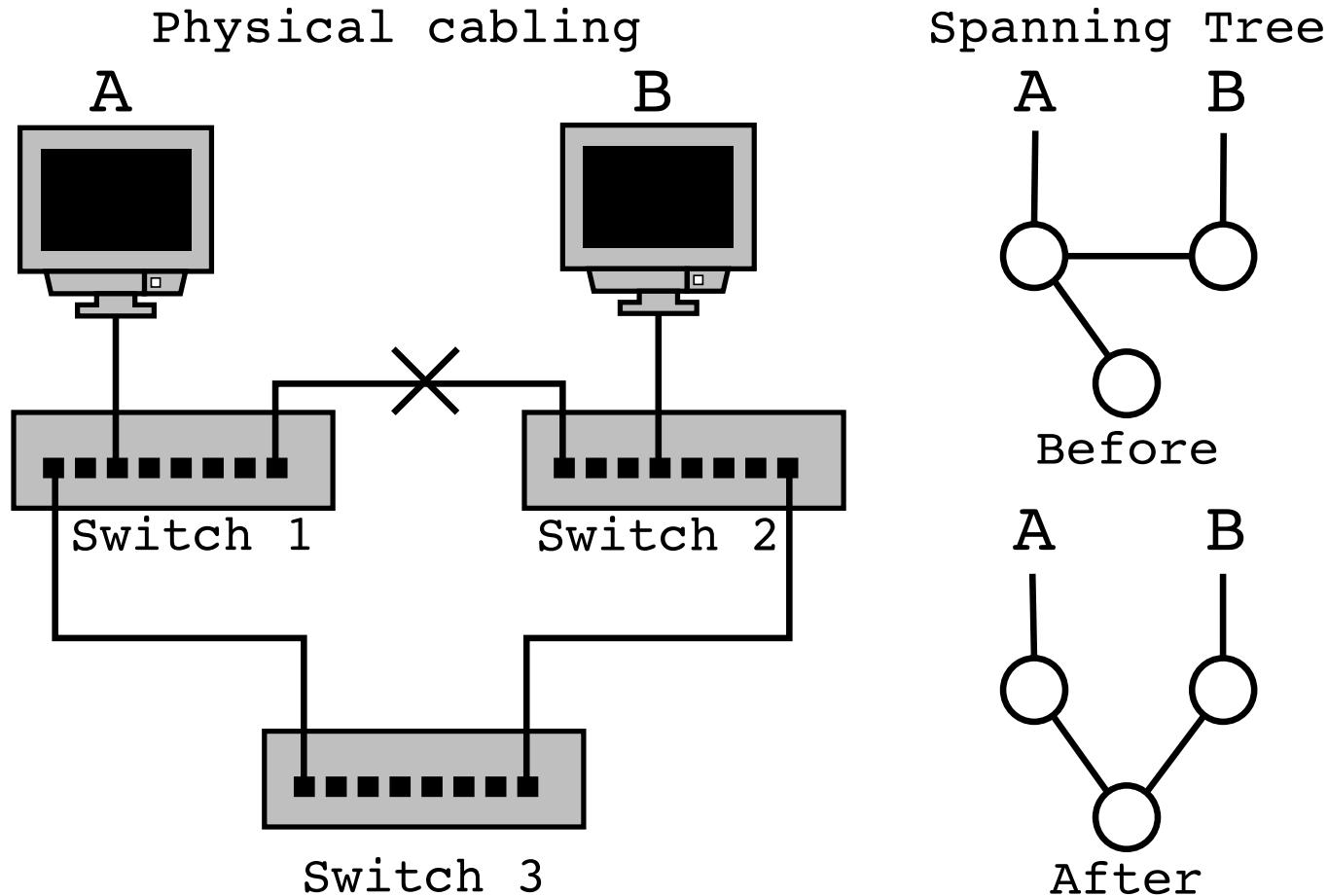


Ved at fortsætte udviklingen kunne man samle broer til en switch

En switch idag kan sende og modtage på flere porte samtidig, og med full-duplex

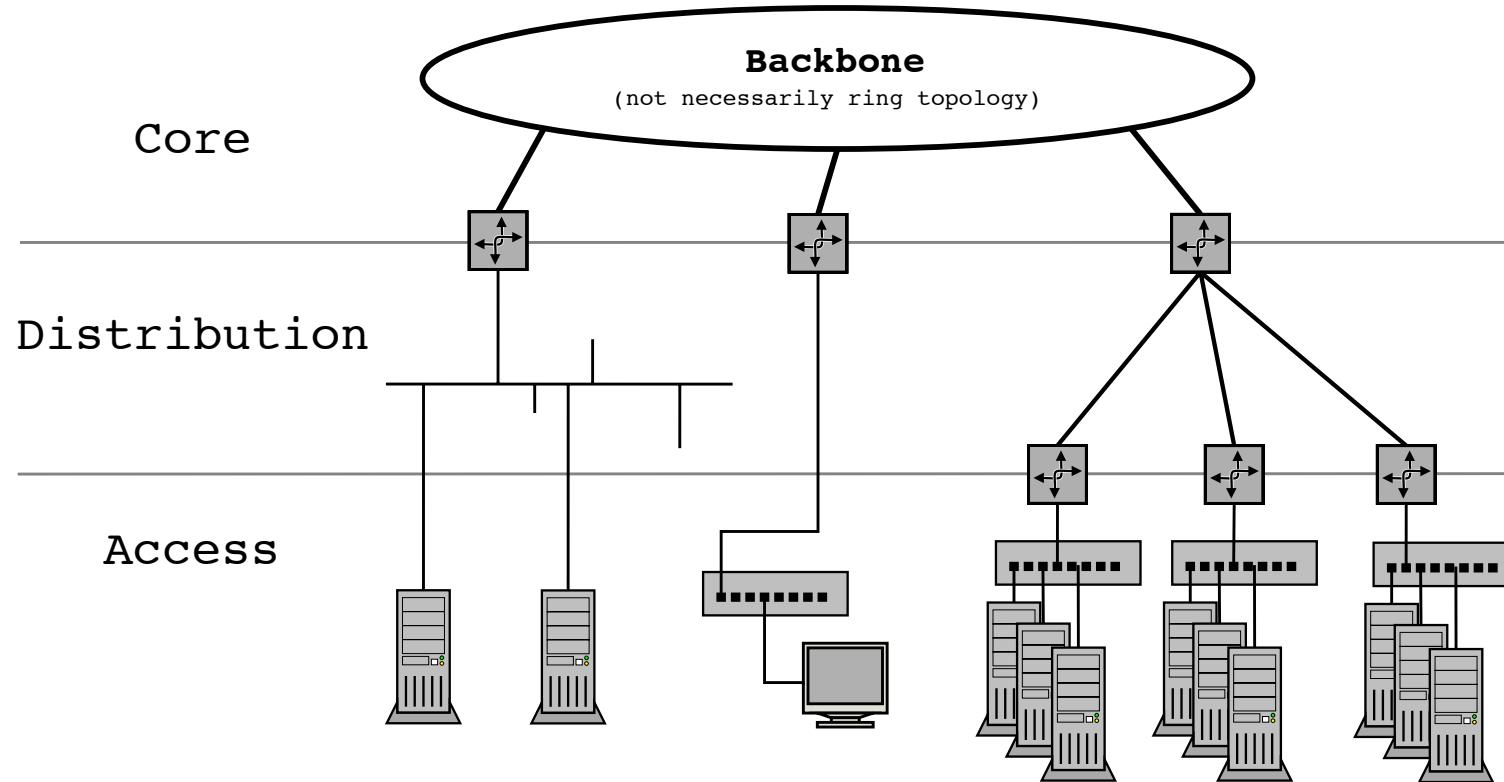
Bemærk performance begrænses af backplane i switchen

# Topologier og Spanning Tree Protocol



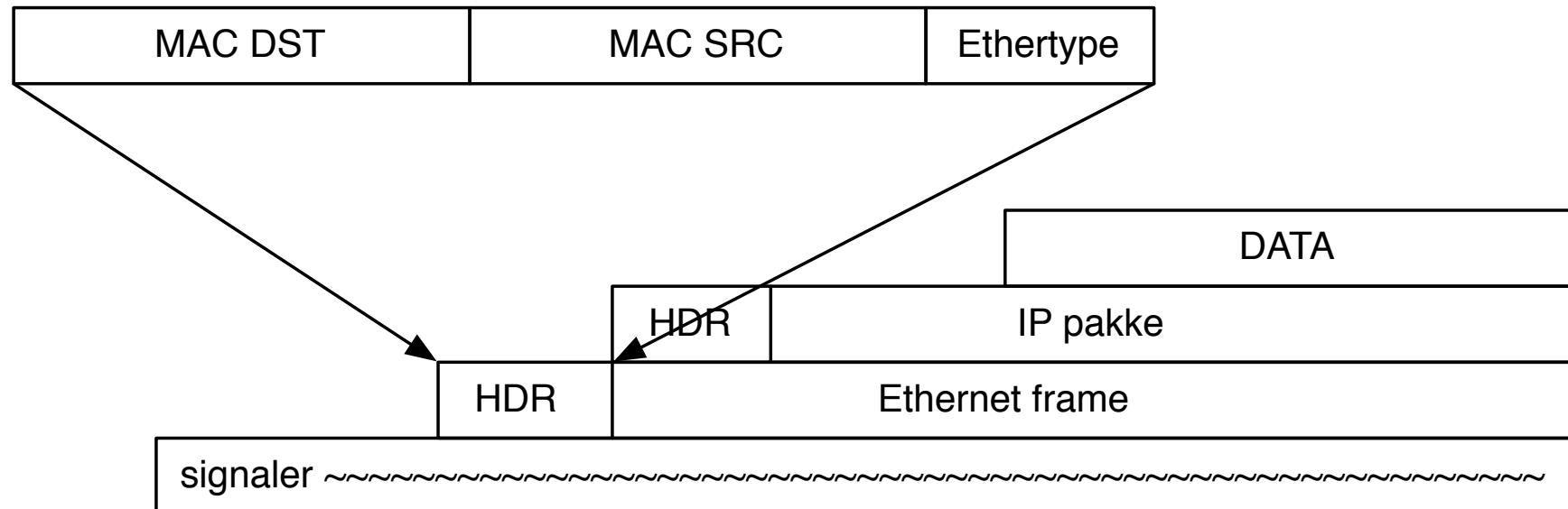
Se mere i bogen af Radia Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*

# Core, Distribution og Access net



Det er ikke altid man har præcis denne opdeling, men den er ofte brugt

# Pakker i en datastrøm

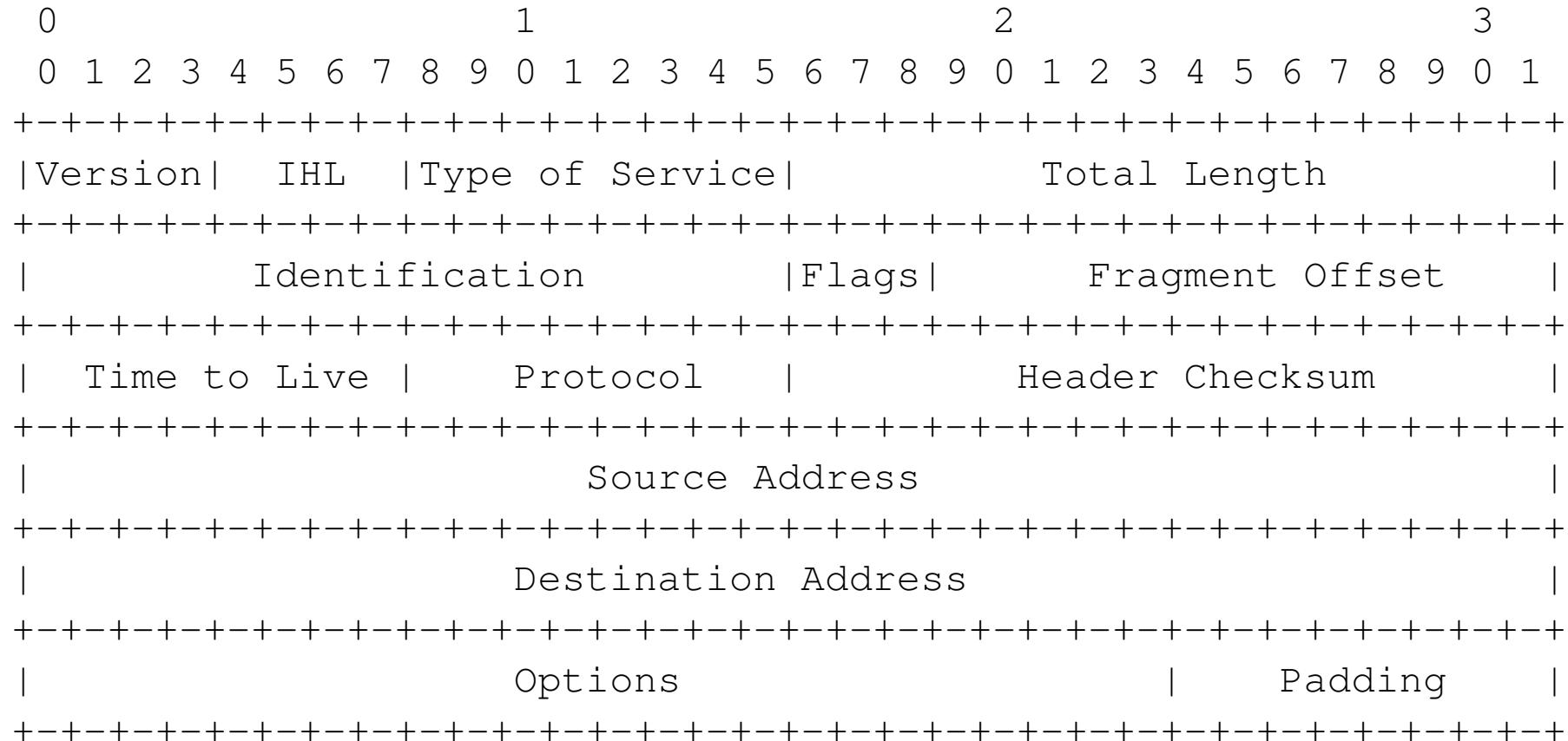


Ser vi data som en datastrøm er pakkerne blot et mønster lagt henover data

Netværksteknologien definerer start og slut på en frame

Fra et lavere niveau modtager vi en pakke, eksempelvis 1500-bytes fra Ethernet driver

# IPv4 pakken - header - RFC-791



Example Internet Datagram Header

## IP karakteristik

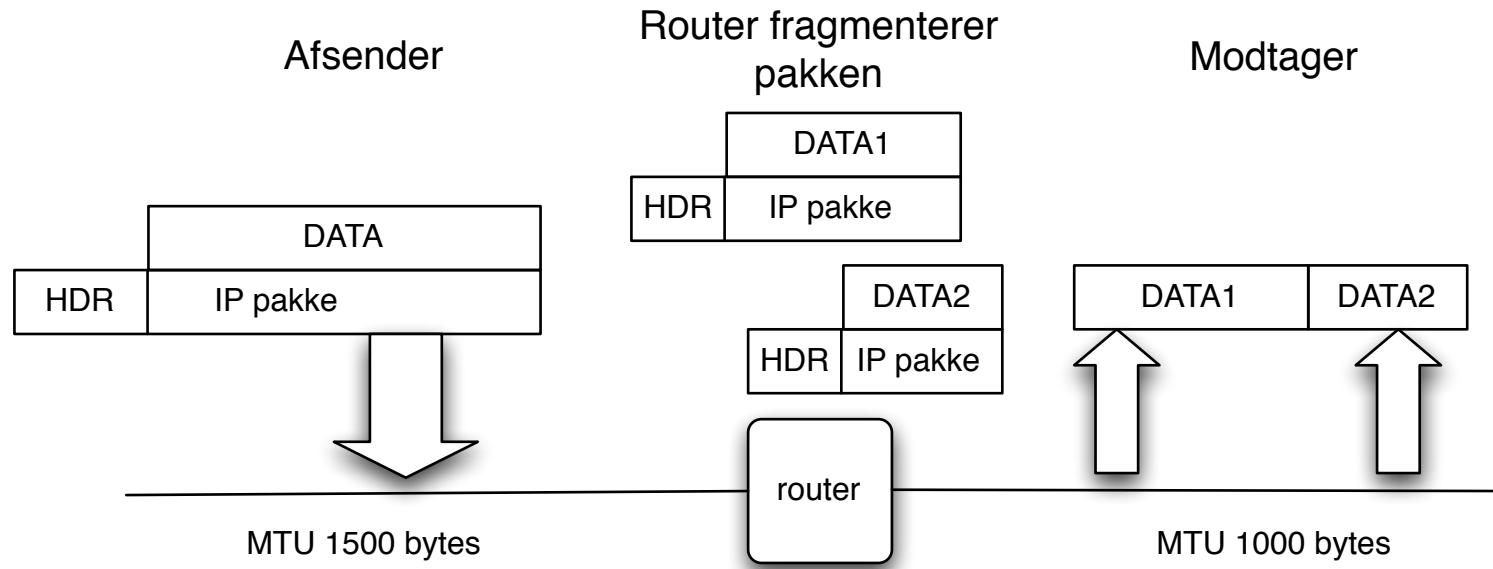
Fælles adresserum

Best effort - kommer en pakke fra er det fint, hvis ikke må højere lag klare det

Kræver ikke mange services fra underliggende teknologi *dumt netværk*

Defineret gennem åben standardiseringsprocess og RFC-dokumenter

# Fragmentering og PMTU



Hidtil har vi antaget at der blev brugt Ethernet med pakkestørrelse på 1500 bytes

Pakkestørrelsen kaldes MTU Maximum Transmission Unit

Skal der sendes mere data opdeles i pakker af denne størrelse, fra afsender

Men hvad hvis en router på vejen ikke bruger 1500 bytes, men kun 1000

Kontrolprotokol og fejlmeldinger

Nogle af de mest almindelige beskedtyper

- echo
- netmask
- info

Bruges generelt til *signalering*

Defineret i RFC-792

**NB: nogle firewall-administratorer blokerer alt ICMP - det er forkert!**

# ICMP beskedtyper

## Type

- 0 = net unreachable;
- 1 = host unreachable;
- 2 = protocol unreachable;
- 3 = port unreachable;
- 4 = fragmentation needed and DF set;
- 5 = source route failed.

Ved at fjerne ALT ICMP fra et net fjerner man nødvendig funktionalitet!

## Tillad ICMP types:

- 3 Destination Unreachable
- 4 Source Quench Message
- 11 Time Exceeded
- 12 Parameter Problem Message

# Hvordan virker ARP?

Server



10.0.0.1

IP adresser

00:30:65:22:94:a1



MAC adresser - Ethernet

Client



10.0.0.2

10.0.0.2

## Hvordan virker ARP? - 2

**ping 10.0.0.2** udført på server medfører

ARP Address Resolution Protocol request/reply:

- ARP request i broadcast - Who has 10.0.0.2 Tell 10.0.0.1
- ARP reply (fra 10.0.0.2) 10.0.0.2 is at 00:40:70:12:95:1c

IP ICMP request/reply:

- Echo (ping) request fra 10.0.0.1 til 10.0.0.2
- Echo (ping) reply fra 10.0.0.2 til 10.0.0.1
- ...

ARP udføres altid på Ethernet før der kan sendes IP trafik

(kan være RARP til udstyr der henter en adresse ved boot)

# ARP cache

```
hlk@bigfoot:hlk$ arp -an
? (10.0.42.1) at 0:0:24:c8:b2:4c on en1 [ethernet]
? (10.0.42.2) at 0:c0:b7:6c:19:b on en1 [ethernet]
```

ARP cache kan vises med kommandoen `arp -an`

`-a` viser alle

`-n` viser kun adresserne, prøver ikke at slå navne op - typisk hurtigere

ARP cache er dynamisk og adresser fjernes automatisk efter 5-20 minutter hvis de ikke bruges mere

Læs mere med `man 4 arp`

# Manualsystemet

It is a book about a Spanish guy called Manual. You should read it. – Dilbert

Manualsystemet i UNIX er utroligt stærkt!

Det SKAL altid installeres sammen med værktøjerne!

Det er næsten identisk på diverse UNIX varianter!

man –k søger efter keyword, se også apropos

Prøv man crontab og man 5 crontab

kommando [options] [argumenter]

\$ cal -j 2005

CAL(1)

BSD General Commands Manual

CAL(1)

## NAME

cal - displays a calendar

## SYNOPSIS

cal [-jy] [[month] year]

## DESCRIPTION

cal displays a simple calendar. If arguments are not specified, the current month is displayed. The options are as follows:

- j      Display julian dates (days one-based, numbered from January 1).
- y      Display a calendar for the current year.

The Gregorian Reformation is assumed to have occurred in 1752 on the 3rd of September. By this time, most countries had recognized the reformation (although a few did not recognize it until the early 1900's.) Ten days following that date were eliminated by the reformation, so the calendar for that month is a bit unusual.

## HISTORY

A cal command appeared in Version 6 AT&T UNIX.

# Kommandolinien på UNIX

Shells kommandofortolkere:

- sh - Bourne Shell
- bash - Bourne Again Shell
- ksh - Korn shell, lavet af David Korn
- csh - C shell, syntaks der minder om C sproget
- flere andre, zsh, tcsh

Svarer til command.com og cmd.exe på Windows

Kan bruges som komplette programmeringssprog

# Kommandoprompten

```
[hlk@fischer hlk]$ id  
uid=6000(hlk) gid=20(staff) groups=20(staff),  
0(wheel), 80(admin), 160(cvs)  
[hlk@fischer hlk]$
```

```
[root@fischer hlk]# id  
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon),  
2(kmem), 3(sys), 4(tty), 5(operator), 20(staff),  
31(guest), 80(admin)  
[root@fischer hlk]#
```

typisk viser et dollartegn at man er logget ind som almindelig bruge  
mens en havelåge at man er root - superbruger

# Kommandoliniens opbygning

```
echo [-n] [string ...]
```

Kommandoerne der skrives på kommandolinien skrives sådan:

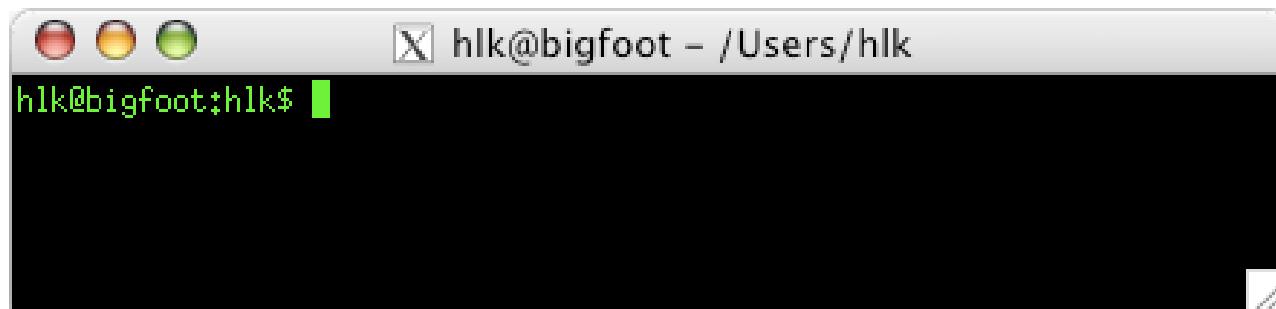
- Starter altid med kommandoen, man kan ikke skrive henrik echo
- Options skrives typisk med bindestreg foran, eksempelvis -n
- Flere options kan sættes sammen, tar -cvf eller tar cvf
- I manualsystemet kan man se valgfrie options i firkantede klammer []
- Argumenterne til kommandoen skrives typisk til sidst (eller der bruges redirection)

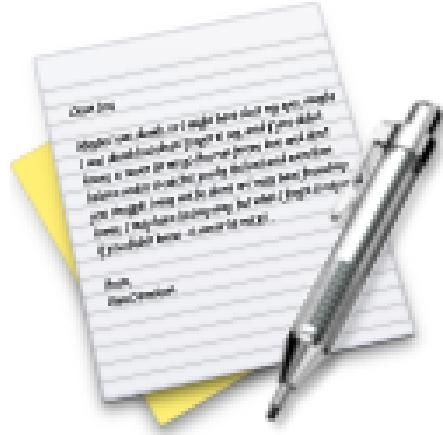


Adgang til UNIX kan ske via grafiske brugergrænseflader

- KDE <http://www.kde.org>
- GNOME <http://www.gnome.org>

eller kommandolinien

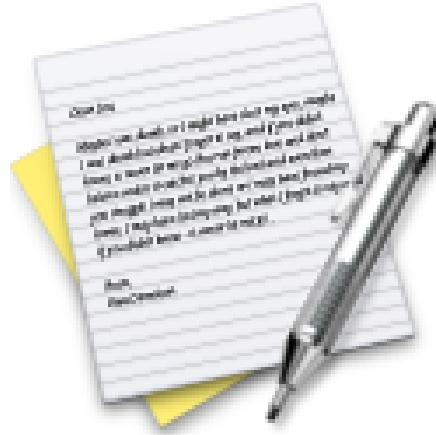




Vi laver nu øvelsen

## Putty installation - Secure Shell login

som er øvelse 1 fra øvelseshæftet.



Vi laver nu øvelsen

## WinSCP installation - Secure Copy

som er øvelse **2** fra øvelseshæftet.



Vi laver nu øvelsen

## Login på UNIX systemerne

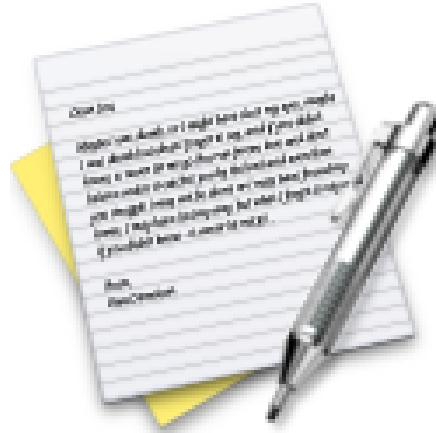
som er øvelse 3 fra øvelseshæftet.



Vi laver nu øvelsen

## Føling med UNIX

som er øvelse **4** fra øvelseshæftet.



Vi laver nu øvelsen

## UNIX - adgang til root

som er øvelse 5 fra øvelseshæftet.



Vi laver nu øvelsen

## UNIX boot CD

som er øvelse **6** fra øvelseshæftet.



Vi laver nu øvelsen

## Basale UNIX kommandoer

som er øvelse 7 fra øvelseshæftet.

# TCP/IP basiskonfiguration

```
ifconfig en0 10.0.42.1 netmask 255.255.255.0  
route add default gw 10.0.42.1
```

konfiguration af interfaces og netværk på UNIX foregår med:

`ifconfig`, `route` **og** `netstat`

- ofte pakket ind i konfigurationsmenuer m.v.

fejlsøgning foregår typisk med `ping` **og** `traceroute`

På Microsoft Windows benyttes ikke `ifconfig`

men kommandoerne `ipconfig` **og** `ipv6`

## Små forskelle

```
$ route add default 10.0.42.1  
uden gw keyword!
```

```
$ route add default gw 10.0.42.1  
Linux kræver gw med
```

**NB: UNIX varianter kan indbyrdes være forskellige!**

## Flere små forskelle

ping eller ping6

Nogle systemer vælger at ping kommandoen kan ping'e både IPv4 og Ipv6

Andre vælger at ping kun benyttes til IPv4, mens IPv6 ping kaldes for ping6

Læg også mærke til jargonen *at pinge*

## Netværkskonfiguration på OpenBSD:

```
# cat /etc/hostname.sk0
inet 10.0.0.23 0xffffffff00 NONE
# cat /etc/mygate
10.0.0.1
# cat /etc/resolv.conf
domain security6.net
lookup file bind
nameserver 212.242.40.3
nameserver 212.242.40.51
```

## Netværkskonfiguration på FreeBSD /etc/rc.conf:

```
# This file now contains just the overrides from /etc/defaults/rc.conf.
hostname="freebsd.security6.net"
#ifconfig_vr0="DHCP"
ifconfig_vr0="inet 10.20.30.75 netmask 255.255.255.0"
router_enable="NO"
defaultrouter="10.20.30.65"
keyrate="fast"
moused_enable="YES"
ntpdate_enable="NO"
ntpdate_flags="none"
saver="blank"
sshd_enable="YES"
usbd_enable="YES"
...
```

# GUI værktøjer - autoconfiguration



Built-in Ethernet

TCP/IP DNS WINS AppleTalk 802.1X Proxies Ethernet

Configure IPv4: Using DHCP

IPv4 Address:

Subnet Mask:

Router:

DHCP Client ID:   
( If required )

Configure IPv6: Automatically

IPv6 Address:

Prefix Length:

# GUI værktøjer - manuel konfiguration



Built-in Ethernet

TCP/IP DNS WINS AppleTalk 802.1X Proxies Ethernet

Configure IPv4: Manually

IPv4 Address: 0.0.0.0

Subnet Mask:

Router:

Configure IPv6: Manually

Router:

IPv6 Address:

Prefix Length:

Advanced

# ifconfig output

```
hlk@bigfoot:hlk$ ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:0a:95:db:c8:b0
        media: autoselect (none) status: inactive
        supported media: none autoselect 10baseT/UTP <half-duplex> 10baseT/UTP
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:0d:93:86:7c:3f
        media: autoselect (<unknown type>) status: inactive
        supported media: autoselect
```

ifconfig output er næsten ens på tværs af UNIX

# Vigtigste protokoller

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

UDP User Datagram Protocol

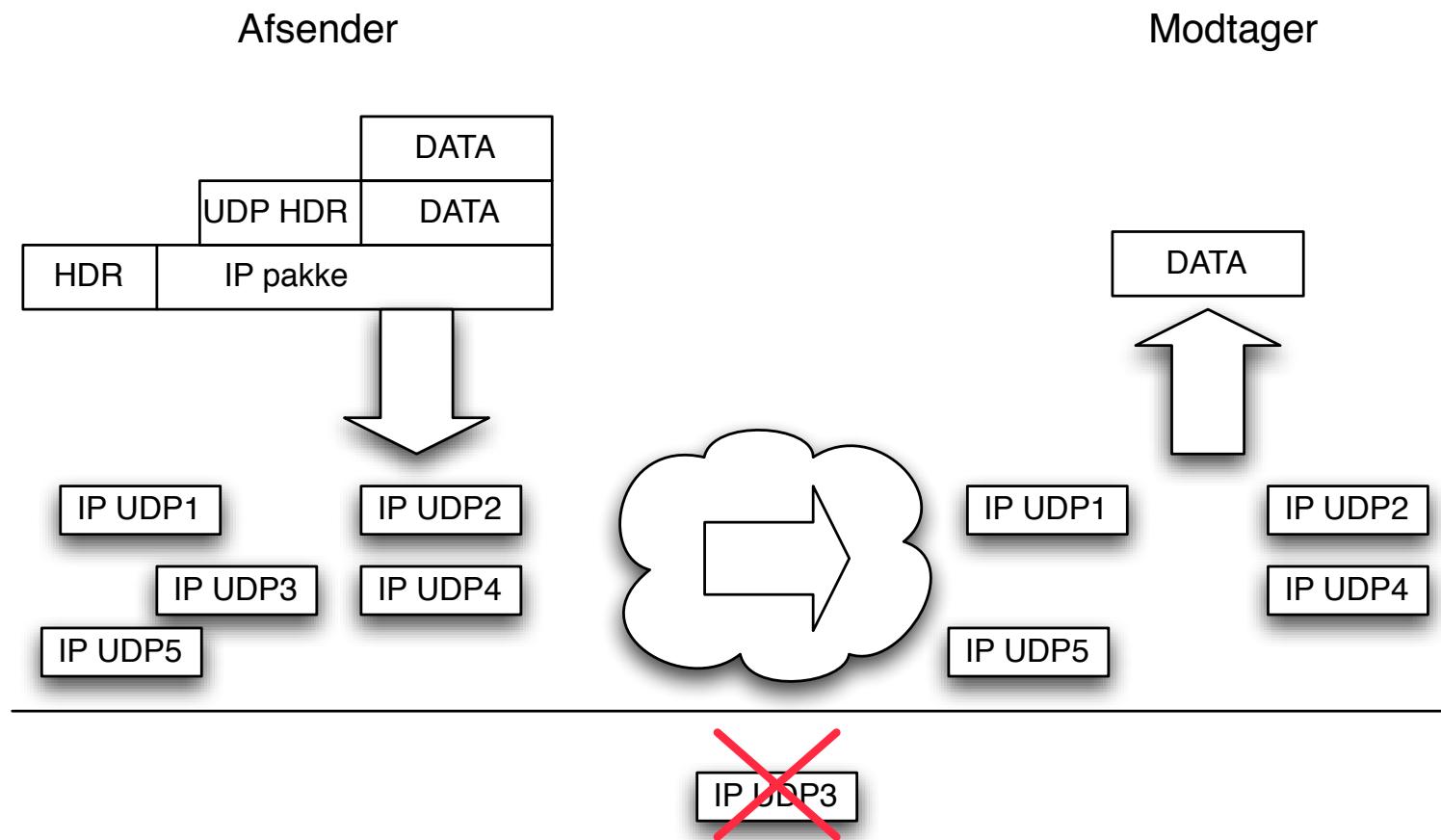
TCP Transmission Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Ovenstående er omtrent minimumskrav for at komme på internet

# UDP User Datagram Protocol



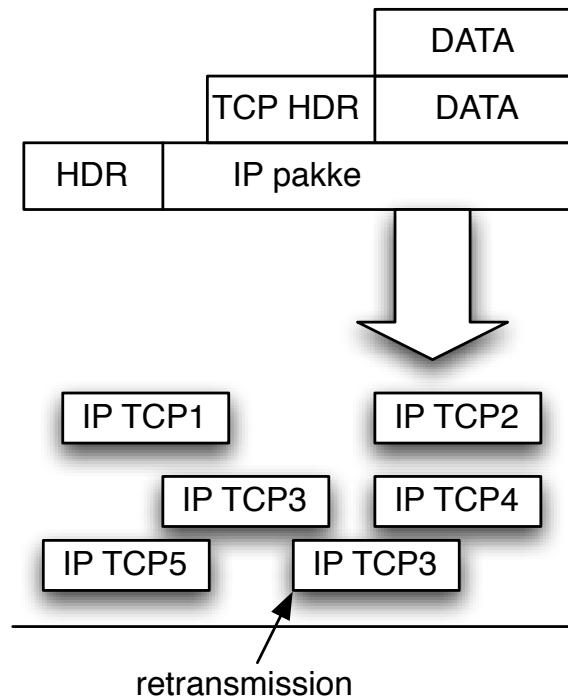
Forbindelsesløs RFC-768, *connection-less* - der kan tabes pakker

Kan benyttes til multicast/broadcast - flere modtagere

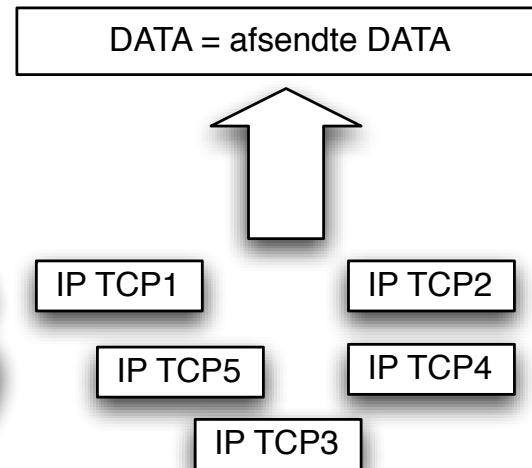
# TCP Transmission Control Protocol



## Afsender



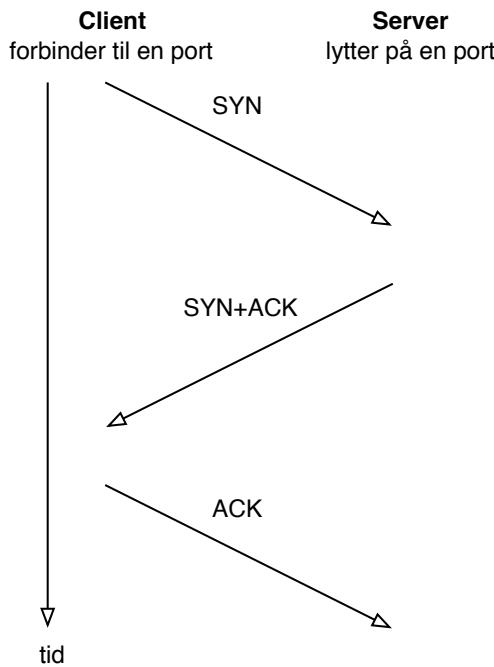
## Modtager



## Forbindelsesorienteret RFC-791 September 1981, *connection-oriented*

Enten overføres data eller man får fejlmeddeelse

# TCP three way handshake



- **TCP SYN half-open scans**
- Tidligere loggede systemer kun når der var etableret en fuld TCP forbindelse - dette kan/kunne udnyttes til *stealth*-scans
- Hvis en maskine modtager mange SYN pakker kan dette fylde tabellen over connections op - og derved afholde nye forbindelser fra at blive oprette - **SYN-flooding**

# Well-known port numbers



IANA vedligeholder en liste over magiske konstanter i IP

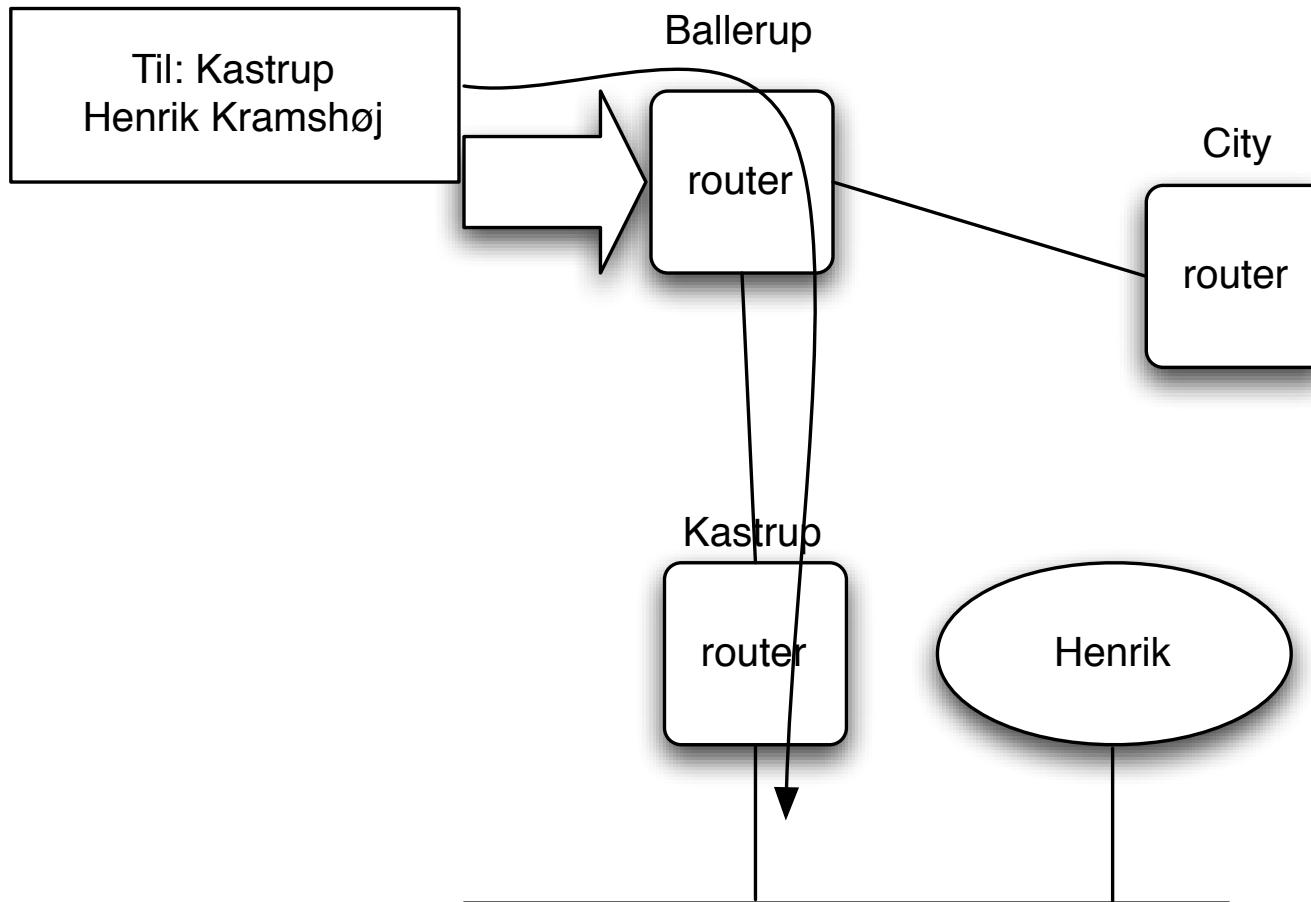
De har lister med hvilke protokoller har hvilke protokol ID m.v.

En liste af interesse er port numre, hvor et par eksempler er:

- Port 25 SMTP Simple Mail Transfer Protocol
- Port 53 DNS Domain Name System
- Port 80 HTTP Hyper Text Transfer Protocol over TLS/SSL
- Port 443 HTTP over TLS/SSL

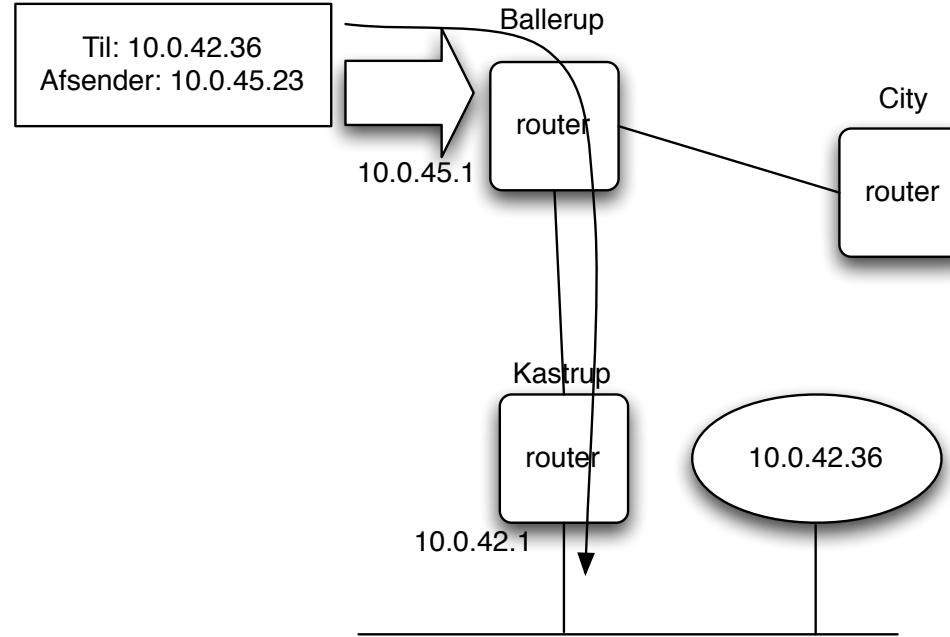
Se flere på <http://www.iana.org>

# Hierarkisk routing



Hvordan kommer pakkerne frem til modtageren

# IP default gateway



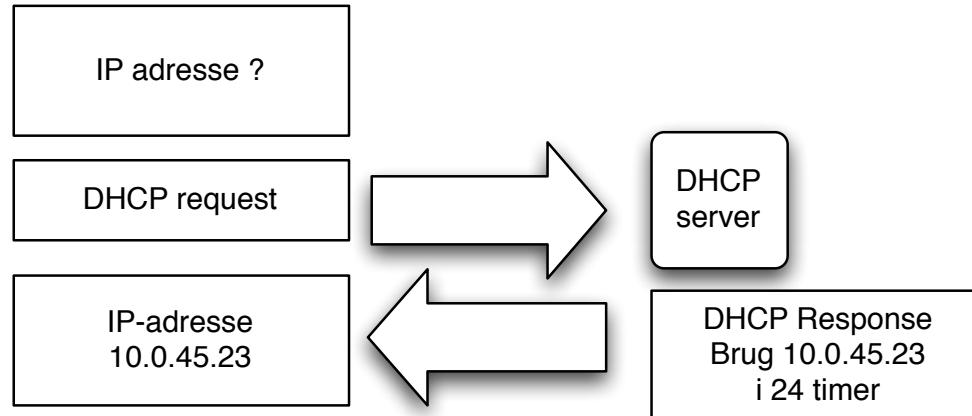
IP routing er nemt

En host kender en default gateway i nærheden

En router har en eller flere upstream routere, få adresser den sender videre til

Core internet har default free zone, kender *alle netværk*

# DHCP Dynamic Host Configuration Protocol



Hvordan får man information om default gateway

Man sender et DHCP request og modtager et svar fra en DHCP server

Dynamisk konfiguration af klienter fra en centralt konfigureret server

Bruges til IP adresser og meget mere

# Routing

routing table - tabel over netværkskort og tilhørende adresser

default gateway - den adresse hvortil man sender *non-local* pakker  
kaldes også default route, gateway of last resort

routing styres enten manuelt - opdatering af route tabellen, eller konfiguration af adresser og subnet maske på netkort

eller automatisk ved brug af routing protocols - interne og eksterne route protokoller

de lidt ældre routing protokoller har ingen sikkerhedsmekanismer

**IP benytter longest match i routing tabeller!**

Den mest specifikke route gælder for forward af en pakke!

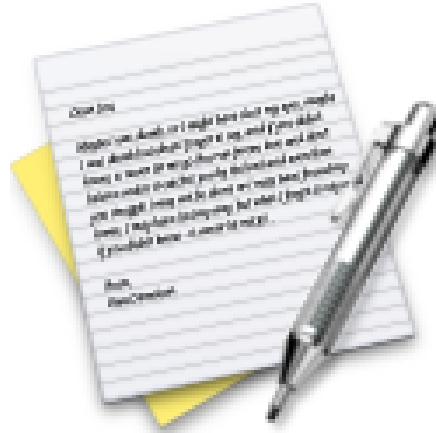
# Routing forståelse

```
$ netstat -rn  
Routing tables
```

Internet:

Destination	Gateway	Flags	Refs	Use	Netif
default	10.0.0.1	UGSc	23	7	en0
10/24	link#4	UCS	1	0	en0
10.0.0.1	0:0:24:c1:58:ac	UHLW	24	18	en0
10.0.0.33	127.0.0.1	UHS	0	1	lo0
10.0.0.63	127.0.0.1	UHS	0	0	lo0
127	127.0.0.1	UCS	0	0	lo0
127.0.0.1	127.0.0.1	UH	4	7581	lo0
169.254	link#4	UCS	0	0	en0

Start med kun at se på Destination, Gateway og Netinterface



Vi laver nu øvelsen

## Netværksinformation: ifconfig

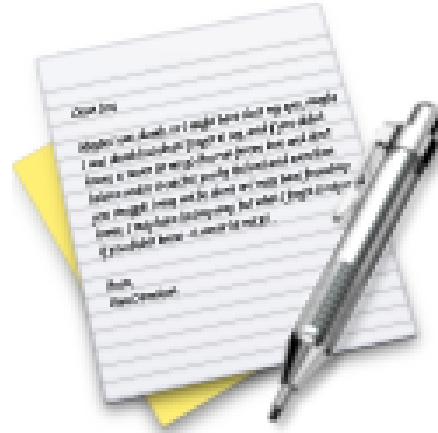
som er øvelse **8** fra øvelseshæftet.



Vi laver nu øvelsen

## Netværksinformation: netstat

som er øvelse **9** fra øvelseshæftet.



Vi laver nu øvelsen

## Netværksinformation: Isof

som er øvelse **10** fra øvelseshæftet.

IP adresserne administreres i dagligdagen af et antal Internet registries, hvor de største er:

- RIPE (Réseaux IP Européens) <http://ripe.net>
- ARIN American Registry for Internet Numbers <http://www.arin.net>
- Asia Pacific Network Information Center <http://www.apnic.net>
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) - Latin America and some Caribbean Islands

disse fire kaldes for Regional Internet Registries (RIRs) i modsætning til Local Internet Registries (LIRs) og National Internet Registry (NIR)

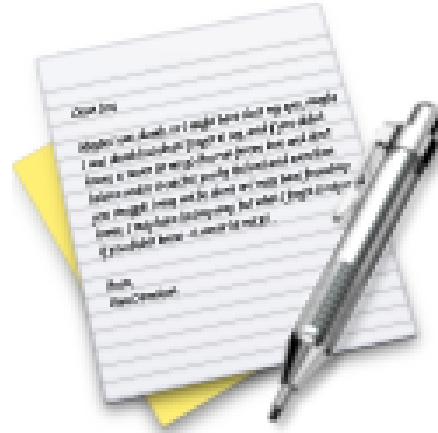
## whois systemet-2



ansvaret for Internet IP adresser ligger hos ICANN The Internet Corporation for Assigned Names and Numbers

<http://www.icann.org>

NB: ICANN må ikke forveksles med IANA Internet Assigned Numbers Authority <http://www.iana.org/> som bestyrer portnumre m.v.



Vi laver nu øvelsen

## Opslag i whois databaser

som er øvelse **11** fra øvelseshæftet.

## ICMP - Internet Control Message Protocol

Benyttes til fejlbeskeder og til diagnosticering af forbindelser

ping programmet virker ved hjælp af ICMP ECHO request og forventer ICMP ECHO reply

```
$ ping 192.168.1.1
```

```
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=150 time=8.849 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=150 time=0.588 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=150 time=0.553 ms
```

## traceroute

traceroute programmet virker ved hjælp af TTL

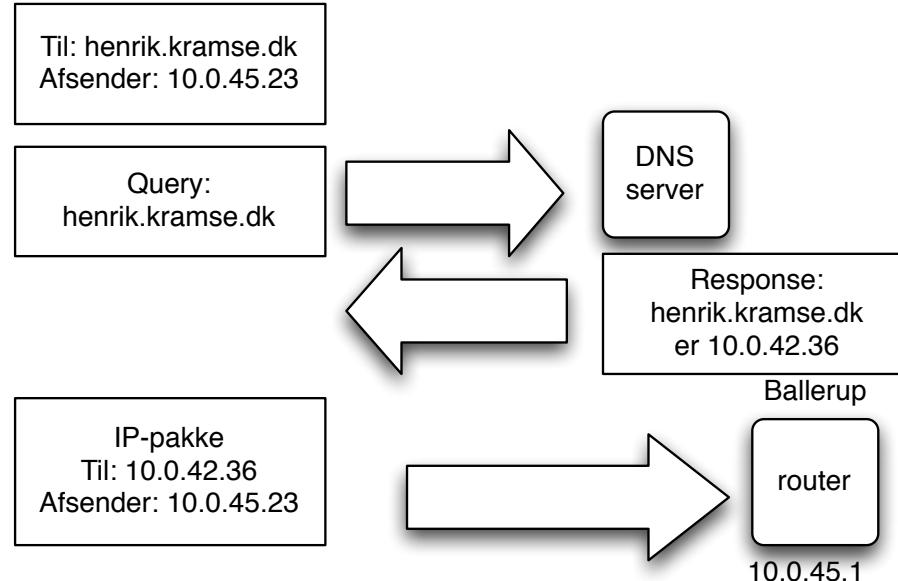
levetiden for en pakke tælles ned i hver router på vejen og ved at sætte denne lavt opnår man at pakken *timer ud* - besked fra hver router på vejen

default er UDP pakker, men på UNIX systemer er der ofte mulighed for at bruge ICMP

```
$ traceroute 217.157.20.129
```

```
traceroute to 217.157.20.129 (217.157.20.129),  
30 hops max, 40 byte packets  
1 safri (10.0.0.11) 3.577 ms 0.565 ms 0.323 ms  
2 router (217.157.20.129) 1.481 ms 1.374 ms 1.261 ms
```

# Domain Name System



Gennem DHCP får man typisk også information om DNS servere

En DNS server kan slå navne, domæner og adresser op

Foregår via query og response med datatyper kaldet resource records

DNS er en distribueret database, så opslag kan resultere i flere opslag

# DNS systemet

Security

.net

navneopslag på Internet

tidligere brugte man en **hosts** fil

hosts filer bruges stadig lokalt til serveren - IP-adresser

UNIX: /etc/hosts

Windows c:\windows\system32\drivers\etc\hosts

Eksempel: www.security6.net har adressen 217.157.20.131

skrives i database filer, zone filer

ns1	IN	A	217.157.20.130
	IN	AAAA	2001:618:433::1
www	IN	A	217.157.20.131
	IN	AAAA	2001:618:433::14

# Mere end navneopslag

består af resource records med en type:

- adresser A-records
- IPv6 adresser AAAA-records
- autoritative navneservere NS-records
- post, mail-exchanger MX-records
- flere andre: md , mf , cname , soa , mb , mg , mr , null , wks , ptr , hinfo , minfo , mx ....

IN	MX	10	mail.security6.net.
IN	MX	20	mail2.security6.net.

# Basal DNS opsætning på klienter

/etc/resolv.conf

NB: denne fil kan hedde noget andet på UNIX varianter!

eksempelvis /etc/netsvc.conf

typisk indhold er domænenavn og IP-adresser for navneservere

```
domain security6.net
nameserver 212.242.40.3
nameserver 212.242.40.51
```

# DNS root servere

Root-servere - 13 stk geografisk distribueret fordelt på Internet

I.ROOT-SERVERS.NET.	3600000	A	192.36.148.17
E.ROOT-SERVERS.NET.	3600000	A	192.203.230.10
D.ROOT-SERVERS.NET.	3600000	A	128.8.10.90
A.ROOT-SERVERS.NET.	3600000	A	198.41.0.4
H.ROOT-SERVERS.NET.	3600000	A	128.63.2.53
C.ROOT-SERVERS.NET.	3600000	A	192.33.4.12
G.ROOT-SERVERS.NET.	3600000	A	192.112.36.4
F.ROOT-SERVERS.NET.	3600000	A	192.5.5.241
B.ROOT-SERVERS.NET.	3600000	A	128.9.0.107
J.ROOT-SERVERS.NET.	3600000	A	198.41.0.10
K.ROOT-SERVERS.NET.	3600000	A	193.0.14.129
L.ROOT-SERVERS.NET.	3600000	A	198.32.64.12
M.ROOT-SERVERS.NET.	3600000	A	202.12.27.33

bestyrer .dk TLD - top level domain

man registrerer ikke .dk-domæner hos DK-hostmaster, men hos en registrator

Et domæne bør have flere navneservere og flere postservere

autoritativ navneserver - ved autoritativt om IP-adresse for maskine.domæne.dk findes

ikke-autoritativ - har på vegne af en klient slået en adresse op

Det anbefales at overveje en service som <http://www.gratisdns.dk> der har 5 navneservere distribueret over stor geografisk afstand - en udenfor Danmark

# Navngivning af servere

Hvordan skal vi kunne huske og administrere servere?

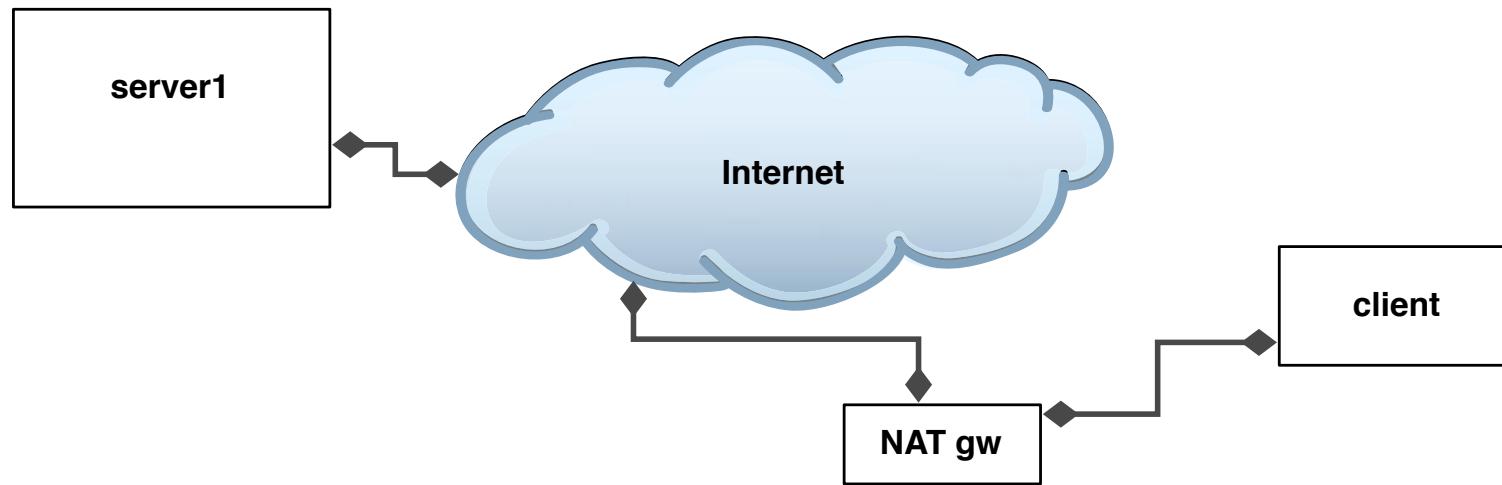
Det er ikke nemt at navngive hverken brugere eller servere!

Selvom det lyder smart med A01S13, som forkortelse af Afdeling 01's Server nr 13, er det umuligt at huske

... men måske nødvendigt i de største netværk

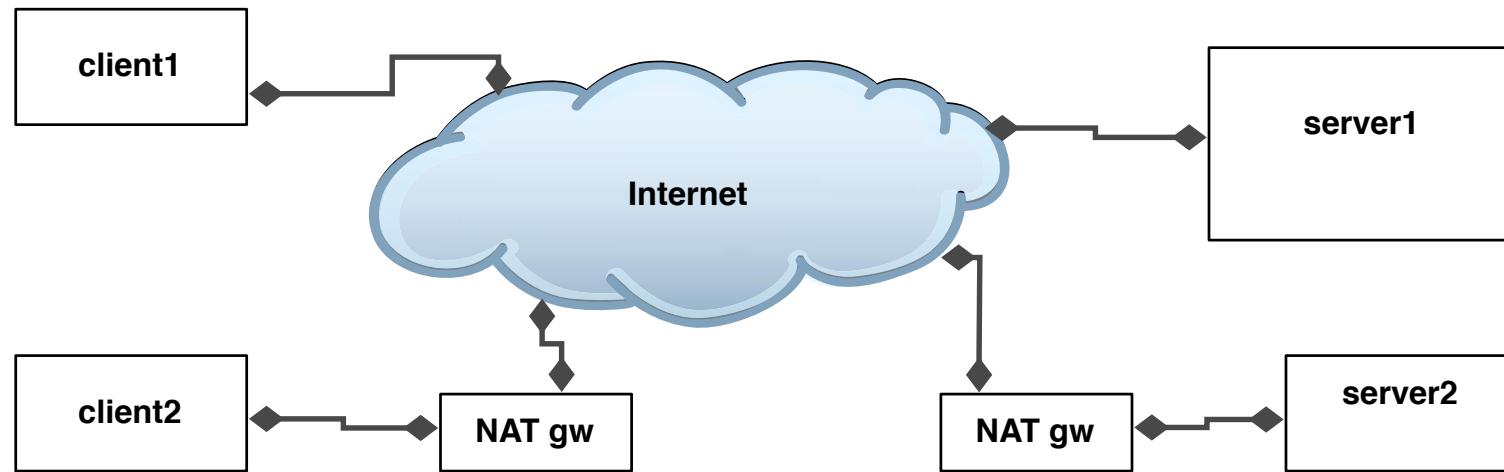
- Windows serveren er domænecontroller - skal hedde:
- Linux server som er terminalserver - skal hedde:
- PC-system med NetBSD skal måske være vores ene server - skal hedde: ?
- PC-system 1 med en Linux server - skal hedde:
- PC-system 2 med en Linux server - skal hedde:

# NAT Network Address Translation

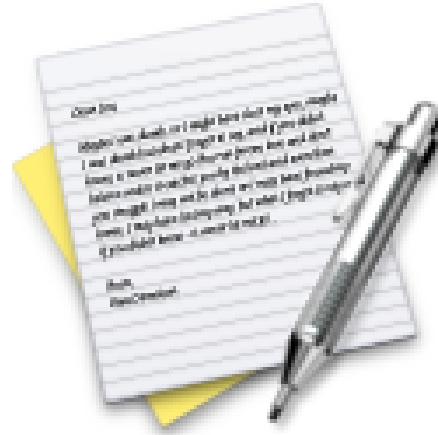


- NAT bruges til at forbinde et privat net (RFC-1918 adresser) med internet
- NAT gateway udskifter afsender adressen med sin egen
- En quick and dirty fix der vil forfølge os for resten af vores liv
- Ødelægger en del protokoller :-)
- Lægger state i netværket - ødelægger fate sharing

# NAT is BAD



- NAT ødelægger end-to-end transparency!
- Problemer med servere bagved NAT
- ”løser” problemet ”godt nok” (tm) for mange
- Men idag ser vi multilevel NAT! - eeeeeeeewwwwww!
- Se RFC-2775 Internet Transparency for mere om dette emne



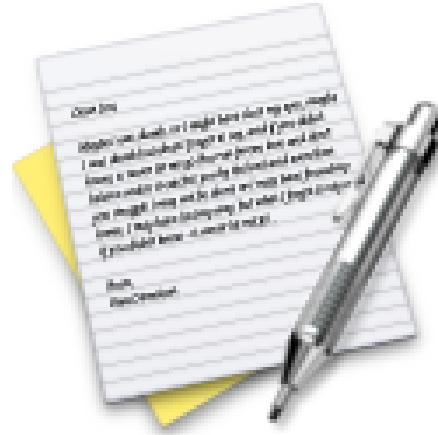
Vi laver nu øvelsen  
**ping og traceroute**  
som er øvelse **12** fra øvelseshæftet.



Vi laver nu øvelsen

## **ICMP tool - icmpush**

som er øvelse **13** fra øvelseshæftet.

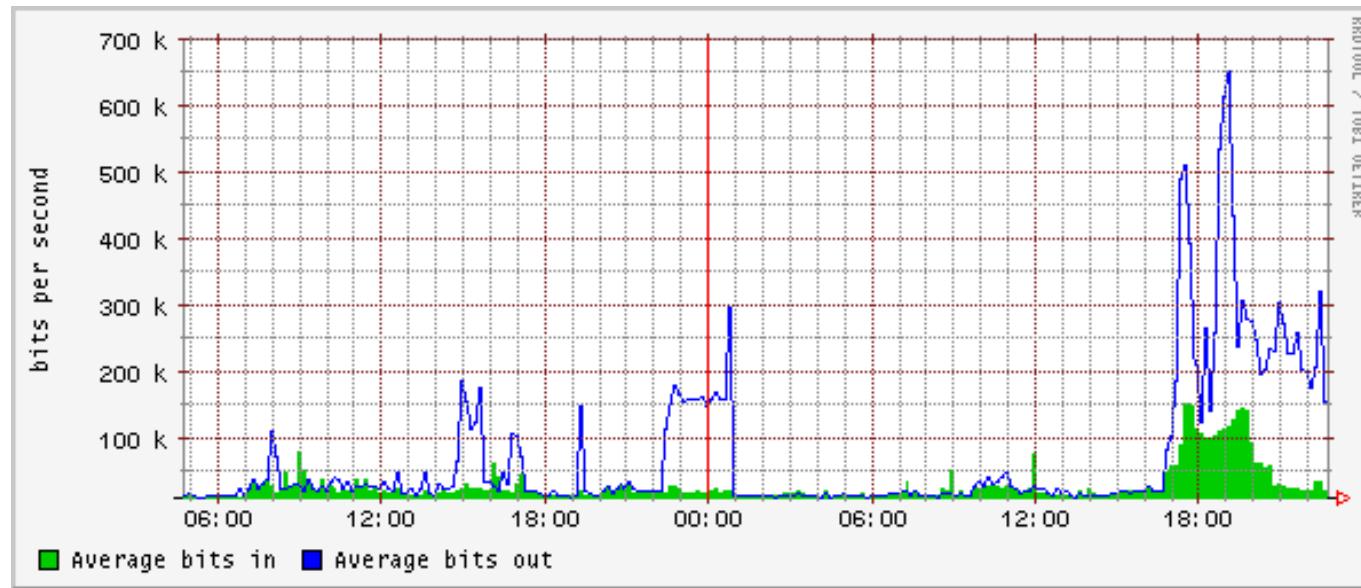


Vi laver nu øvelsen

## DNS og navneopslag

som er øvelse **14** fra øvelseshæftet.

# Dag 2 IPv6, Management, diagnosticering



# IPv4 Adresserummet er ved at løbe ud

Adresserummet er ved at løbe ud! faktum!

32-bit - der ikke kan udnyttes fuldt ud

Tidligere brugte man begreberne A,B og C klasser af IP-adresser

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Address Allocation for Private Internets RFC-1918 adresserne!

Husk at idag benyttes Classless Inter-Domain Routing CIDR

[http://en.wikipedia.org/wiki/Classless\\_Inter-Domain\\_Routing](http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing)

Notation: 192.168.1.0/24

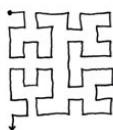
det sædvanlige hjemmenet med subnet maske 255.255.255.0

# Status idag



THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING--ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.

0	1	14	15	16	19	→
3	2	13	12	17	18	
4	7	8	11			
5	6	9	10			



= UNALLOCATED  
BLOCK

# Tidslinie for IPv6 (forkortet)

- 1990 Vancouver IETF meeting det estimeres at klasse B vil løbe ud ca. marts 1994
- 1990 ultimo initiativer til at finde en afløser for IPv4
- 1995 januar RFC-1752 Recommendation for the IP NG Protocol
- 1995 september RFC-1883, RFC-1884, RFC-1885, RFC-1886 1. generation
- 1998 10. august "core" IPv6 dokumenter bliver Draft Standard
- Kilde: RFC-2460, RFC-2461, RFC-2463, RFC-1981 - m.fl.

# IPv6: Internet redesigned? - nej!

Målet var at bevare de gode egenskaber

- basalt set Internet i gamle dage
- back to basics!
- fate sharing
- kommunikationen afhænger ikke af state i netværket
- end-to-end transparency

Idag er Internet blevet en nødvendighed for mange!

**IP er en forretningskritisk ressource**

IPv6 basis i RFC-1752 The Recommendation for the IP Next Generation Protocol

# KAME - en IPv6 reference implementation



<http://www.kame.net>

- Er idag at betragte som en reference implementation
  - i stil med BSD fra Berkeley var det
- KAME har været på forkant med implementation af draft dokumenter
- KAME er inkluderet i OpenBSD, NetBSD, FreeBSD og BSD/OS - har været det siden version 2.7, 1.5, 4.0 og 4.2
- Projektet er afsluttet, men nye projekter fortsætter i WIDE regi <http://www.wide.ad.jp/>
- Der er udkommet to bøger som i detaljer gennemgår IPv6 protokollerne i KAME

# Hvordan bruger man IPv6



[www.inet6.dk](http://www.inet6.dk)

[hlk@inet6.dk](mailto:hlk@inet6.dk)

www.inet6.dk

hlk@inet6.dk

DNS AAAA record tilføjes

www	IN A	91.102.91.17
	IN AAAA	2001:16d8:ff00:12f::2
mail	IN A	91.102.91.17
	IN AAAA	2001:16d8:ff00:12f::2

# IPv6 addresser og skrivemåde

subnet prefix	interface identifier
---------------	----------------------

2001:16d8:ff00:012f:0000:0000:0000:0002  
2001:16d8:ff00:12f::2

- 128-bit adresser, subnet prefix næsten altid 64-bit
- skrives i grupper af 4 hexcifre ad gangen adskilt af kolon :
- foranstillede 0 i en gruppe kan udelades, en række 0 kan erstattes med ::
- dvs 0:0:0:0:0:0 er det samme som  
0000:0000:0000:0000:0000:0000:0000:0000
- Dvs min webservers IPv6 adresse kan skrives som: 2001:16d8:ff00:12f::2
- Specielle adresser: ::1 localhost/loopback og :: default route
- Læs mere i RFC-3513

# IPv6 addresser - prefix notation

CIDR Classless Inter-Domain Routing RFC-1519

Aggregatable Global Unicast

2001::/16 RIR subTLA space

- 2001:200::/23 APNIC
- 2001:400::/23 ARIN
- 2001:600::/23 RIPE

2002::/16 6to4 prefix

3ffe::/16 6bone allocation

link-local unicast addresses

fe80::/10 genereres ud fra MAC addresserne EUI-64

## IPv6 addresser - multicast

Unicast - identifierer ét interface pakker sendes til en modtager

Multicast - identifierer flere interfaces pakker sendes til flere modtagere

Anycast - indentificerer en "gruppe" en pakke sendes til et vilkårligt interface med denne adresse typisk det nærmeste

Broadcast? er væk, udeladt, finito, gone!

Husk også at site-local er deprecated, se RFC-3879

# IPv6 pakken - header - RFC-2460

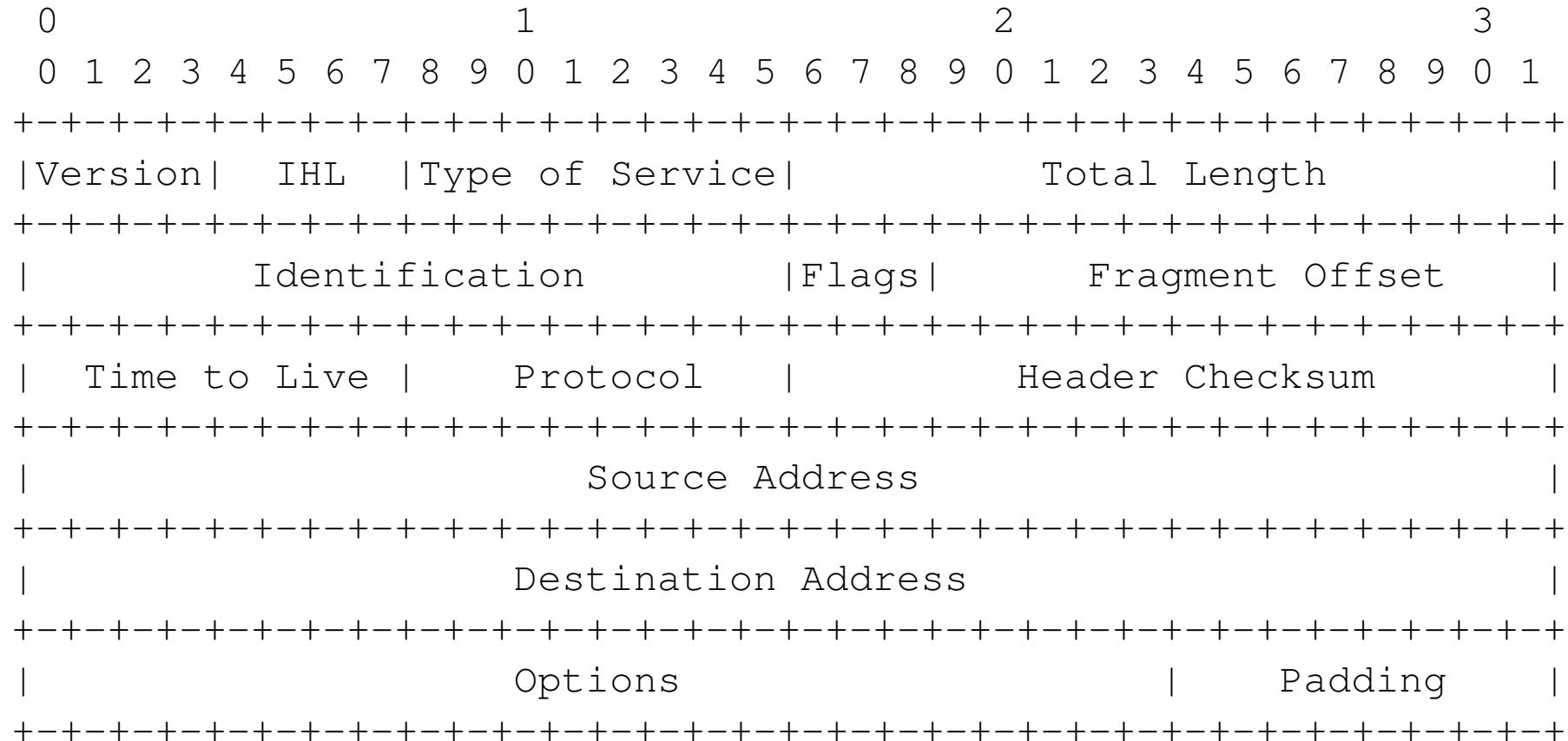
- Simplere - fixed size - 40 bytes
- Sjældent brugte felter (fra v4) udeladt (kun 6 vs 10 i IPv4)
- Ingen checksum!
- Adresser 128-bit
- 64-bit aligned, alle 6 felter med indenfor første 64

Mindre kompleksitet for routere på vejen medfører mulighed for flere pakker på en given router

# IPv6 pakken - header - RFC-2460

Version	Traffic Class	Flow Label	
Payload Length	Next Header	Hop Limit	
Source Address			
Destination Address			

# IPv4 pakken - header - RFC-791



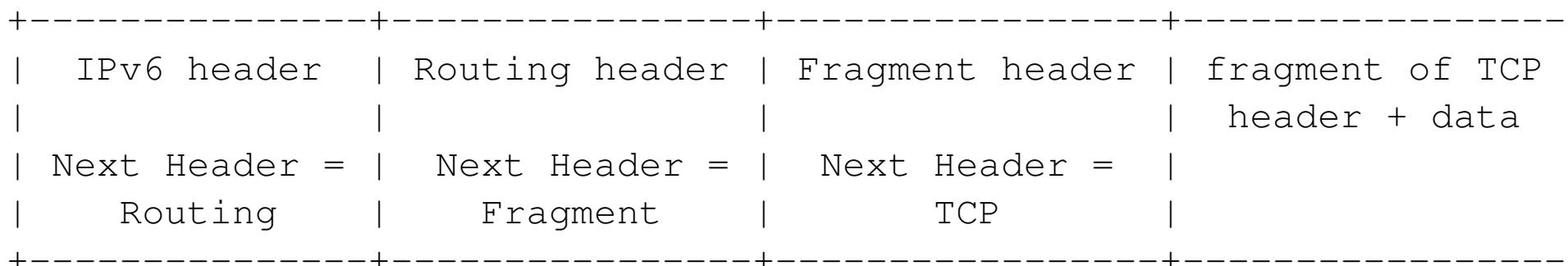
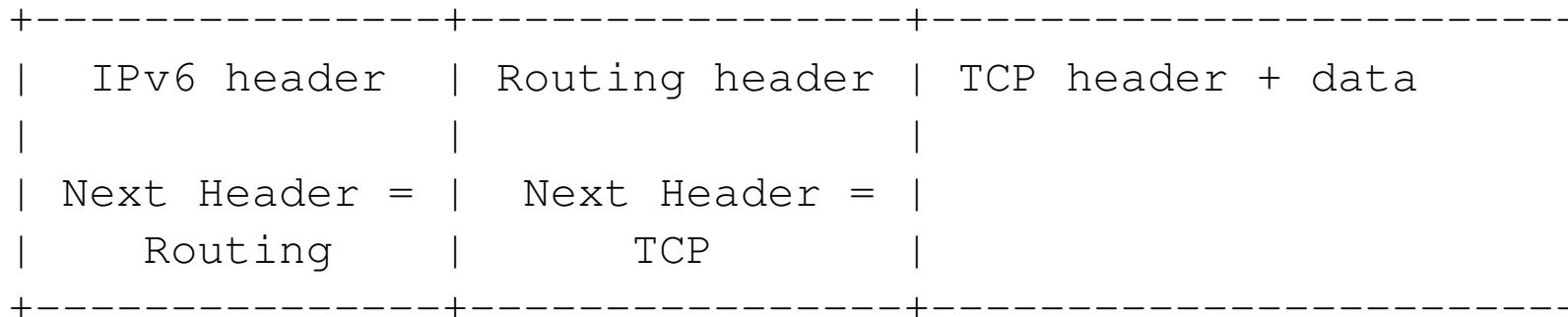
Example Internet Datagram Header

Fuld IPv6 implementation indeholder:

- Hop-by-Hop Options
- Routing (Type 0)
- Fragment - fragmentering KUN i end-points!
- Destination Options
- Authentication
- Encapsulating Security Payload

Ja, IPsec er en del af IPv6!

# Placering af extension headers



# IPv6 configuration - kom igang

Router bagved NAT skal blot kunne forwarde protokoltype 0x41

Cisco 677: set nat entry add 10.1.2.3 0 41

Teredo - the Shipworm er også en mulighed og benyttes aktivt på Windows Vista idag

Officiel IPv4 adresse kan bruges med 6to4 til at lave prefix og router

DNS nameserver anbefales!! tænk på om den skal svare IPv6 AAAA record OG svare over IPv6 sockets - er måske ikke nødvendigt

IPv6-only netværk er sikkert sjældne indtil videre men det er muligt at lave dem nu

Jeg bruger <http://www.sixxs.net> som har vejledninger til diverse operativsystemer

# IPv6 configuration - klienter

```
$ ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.254 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.23 ms
^C
--- ::1 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.230/0.242/0.254 ms
```

Microsoft Windows XP ipv6 install fra kommandolinien eller brug kontrolpanelet

ipv6 giver mulighed for at konfigurere tunnel svarer omtrent til 'ifconfig' på Unix

Migrering er vigtigt i IPv6! Hvis I aktiverer IPv6 nu på en router, vil I pludselig have IPv6 på alle klienter ;-)

Se evt. appendix F Enabling IPv6 functionality i <http://inet6.dk/thesis.pdf>

## ifconfig med ipv6 - Unix

Næsten ingen forskel på de sædvanlige kommandoer ifconfig, netstat,

```
root# ifconfig en1 inet6 2001:1448:81:beef::1
root# ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::230:65ff:fe17:94d1 prefixlen 64 scopeid 0x5
        inet6 2001:1448:81:beef::1 prefixlen 64
        inet 169.254.32.125 netmask 0xffff0000 broadcast 169.254.255.255
        ether 00:30:65:17:94:d1
        media: autoselect status: active
        supported media: autoselect
```

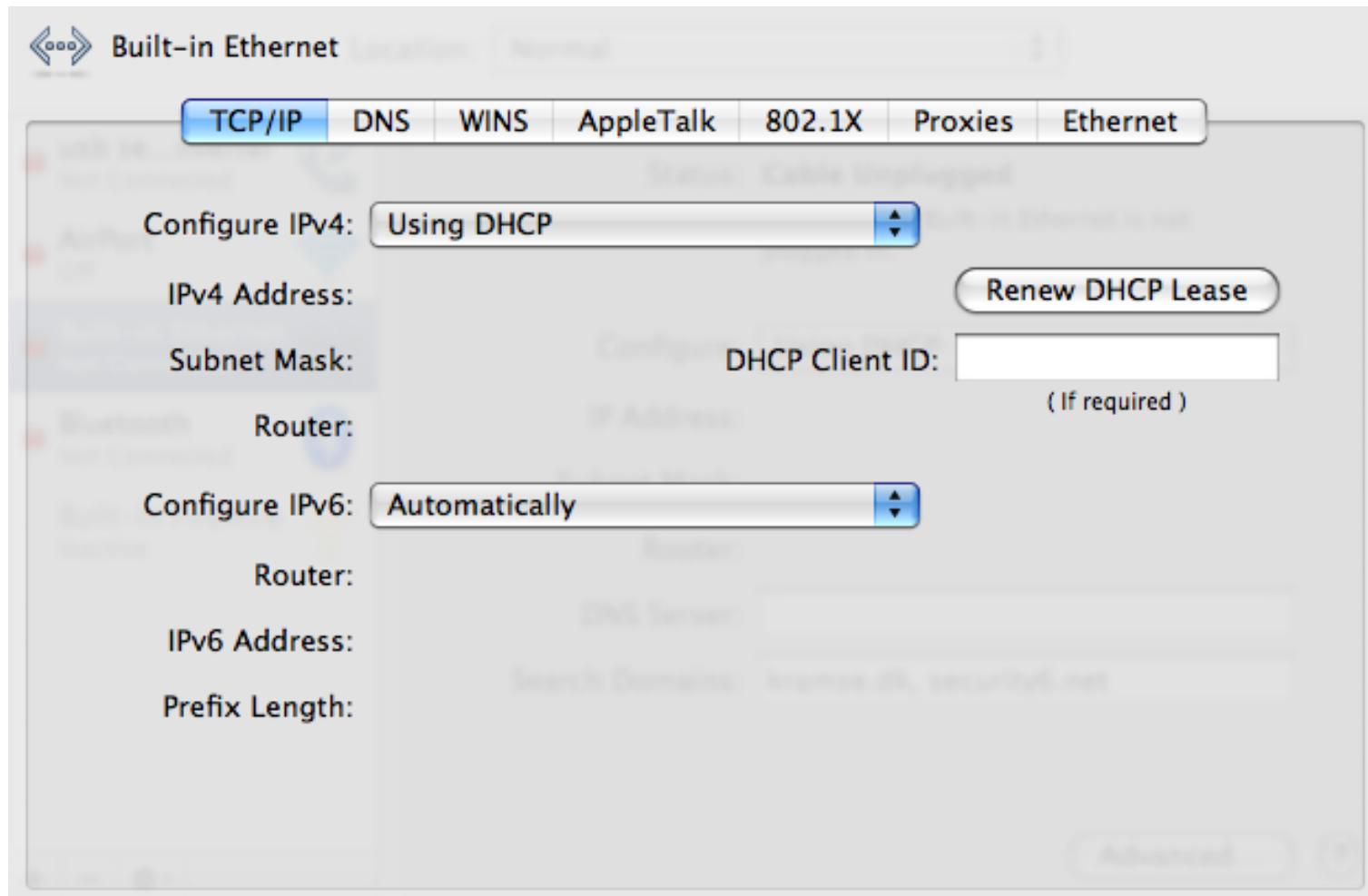
Fjernes igen med:

```
ifconfig en1 inet6 -alias 2001:1448:81:beef::1
```

Prøv også:

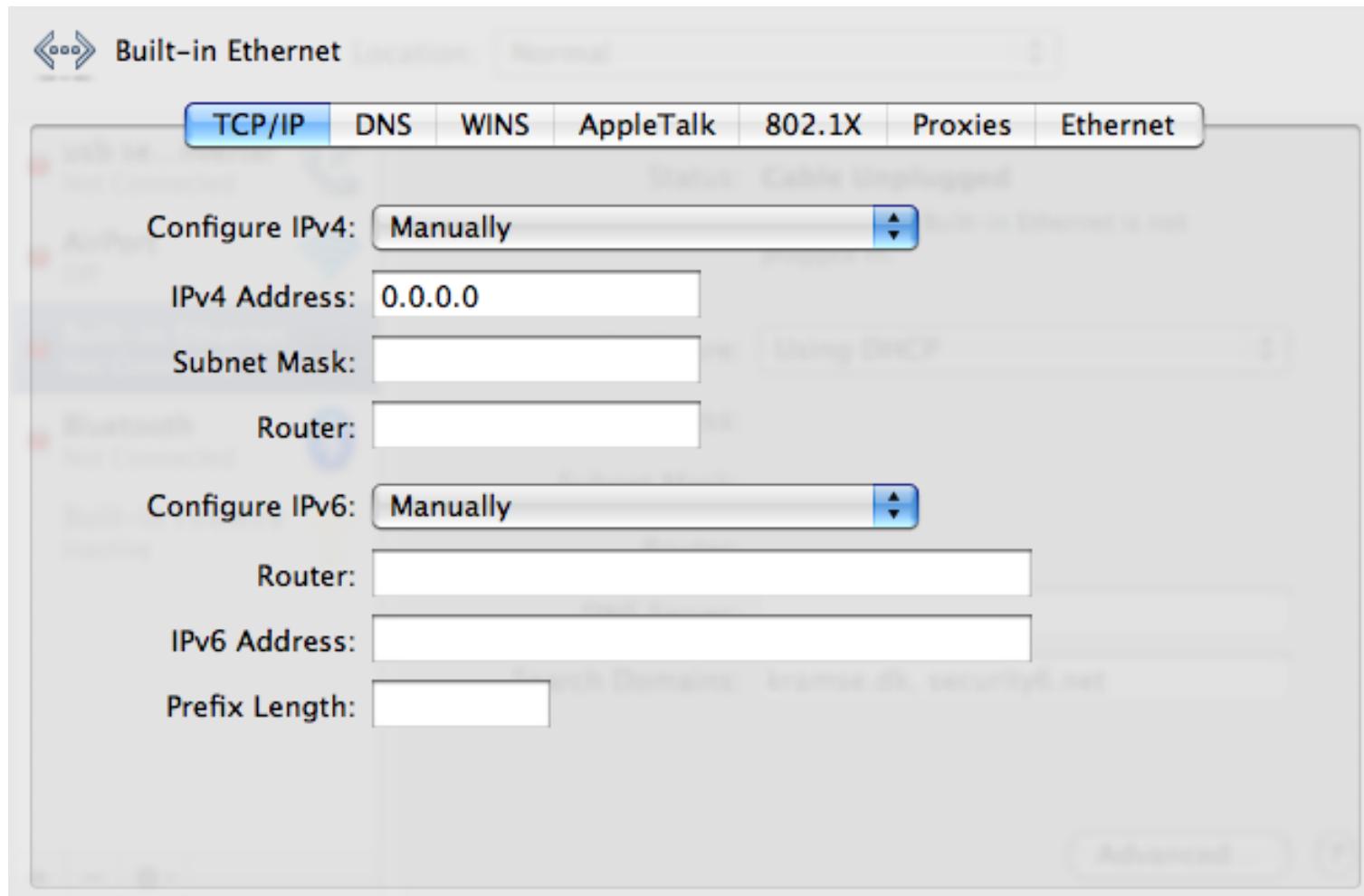
```
ifconfig en1 inet6
```

# GUI værktøjer - autoconfiguration



De fleste moderne operativsystemer er efterhånden opdateret med menuer til IPv6

# GUI værktøjer - manuel konfiguration



Bemærk hvorledes subnetmaske nu blot er en prefix length

# ping til IPv6 adresser

```
root# ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.312 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.319 ms
^C
--- localhost ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.312/0.316/0.319 ms
```

Nogle operativsystemer kalder kommandoen ping6, andre bruger blot ping

# ping6 til global unicast adresse

```
root# ping6 2001:1448:81:beef:20a:95ff:fef5:34df
PING6(56=40+8+8 bytes) 2001:1448:81:beef::1 --> 2001:1448:81:beef:20a:95ff:fef5:34df
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=0 hlim=64 time=10.639 ms
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=1 hlim=64 time=1.615 ms
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=2 hlim=64 time=2.074 ms
^C
--- 2001:1448:81:beef:20a:95ff:fef5:34df ping6 statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.615/4.776/10.639 ms
```

# ping6 til specielle adresser

```
root# ping6 -I en1 ff02::1
PING6(56=40+8+8 bytes) fe80::230:65ff:fe17:94d1 --> ff02::1
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=0 hlim=64 time=0.483 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=0 hlim=64 time=982.932 ms
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=1 hlim=64 time=0.582 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=1 hlim=64 time=9.6 ms
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=2 hlim=64 time=0.489 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=2 hlim=64 time=7.636 ms
^C
--- ff02::1 ping6 statistics ---
4 packets transmitted, 4 packets received, +4 duplicates, 0% packet loss
round-trip min/avg/max = 0.483/126.236/982.932 ms
```

- ff02::1 ipv6-allnodes
- ff02::2 ipv6-allrouters
- ff02::3 ipv6-allhosts

## Stop - tid til leg

Der findes et trådløst netværk med IPv6

Join med en laptop og prøv at pinge lidt

1. Virker `ping6 ::1` eller `ping ::1`, fortsæt
2. Virker kommando svarende til: `ping6 -I en1 ff02::1`  
- burde vise flere maskiner
3. Kig på dine egne adresser med: `ipv6` (Windows) eller `ifconfig` (Unix)
4. Prøv at trace i netværket

Hvordan fik I IPv6 adresser?

# router advertisement daemon

```
/etc/rtadvd.conf:  
en0:  
    :addrs#1:addr="2001:1448:81:b00f::":prefixlen#64:  
en1:  
    :addrs#1:addr="2001:1448:81:beef::":prefixlen#64:  
  
root# /usr/sbin/rtadvd -Df en0 en1  
root# sysctl -w net.inet6.ip6.forwarding=1  
net.inet6.ip6.forwarding: 0 -> 1
```

Stateless autoconfiguration er en stor ting i IPv6

Kommandoen starter den i debug-mode og i forgrunden  
- normalt vil man starte den fra et script

Typisk skal forwarding aktiveres, som vist med BSD sysctl kommando

# IPv6 og andre services

```
root# netstat -an | grep -i listen
```

```
tcp46 0 0 *.80 *.* LISTEN
tcp4 0 0 *.6000 *.* LISTEN
tcp4 0 0 127.0.0.1.631 *.* LISTEN
tcp4 0 0 *.25 *.* LISTEN
tcp4 0 0 *.20123 *.* LISTEN
tcp46 0 0 *.20123 *.* LISTEN
tcp4 0 0 127.0.0.1.1033 *.* LISTEN
```

ovenstående er udført på Mac OS X

# IPv6 output fra kommandoer - inet6 family

```
root# netstat -an -f inet6
```

Active Internet connections (including servers)

Proto	Recv	Send	Local	Foreign	(state)
tcp46	0	0	*.80	*.*	LISTEN
tcp46	0	0	*.22780	*.*	LISTEN
udp6	0	0	*.5353	*.*	
udp6	0	0	*.5353	*.*	
udp6	0	0	*.514	*.*	
icm6	0	0	*.*	*.*	
icm6	0	0	*.*	*.*	
icm6	0	0	*.*	*.*	

ovenstående er udført på Mac OS X og tilrettet lidt

# IPv6 er default for mange services

```
root# telnet localhost 80
```

```
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Thu, 19 Feb 2004 09:22:34 GMT
Server: Apache/2.0.43 (Unix)
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
...
```

# IPv6 er default i OpenSSH

```
hlk$ ssh -v localhost -p 20123
OpenSSH_3.6.1p1+CAN-2003-0693, SSH protocols 1.5/2.0, OpenSSL 0x0090702f
debug1: Reading configuration data /Users/hlk/.ssh/config
debug1: Applying options for *
debug1: Reading configuration data /etc/ssh_config
debug1: Rhosts Authentication disabled, originating port will not be trusted.
debug1: Connecting to localhost [::1] port 20123.
debug1: Connection established.
debug1: identity file /Users/hlk/.ssh/id_rsa type -1
debug1: identity file /Users/hlk/.ssh/id_dsa type 2
debug1: Remote protocol version 2.0, remote software version OpenSSH_3.6.1p1+CA
debug1: match: OpenSSH_3.6.1p1+CAN-2003-0693 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_3.6.1p1+CAN-2003-0693
```

# Apache access log

```
root# tail -f access_log
::1 - - [19/Feb/2004:09:05:33 +0100] "GET /images/IPv6ready.png
HTTP/1.1" 304 0
::1 - - [19/Feb/2004:09:05:33 +0100] "GET /images/valid-html401.png
HTTP/1.1" 304 0
::1 - - [19/Feb/2004:09:05:33 +0100] "GET /images/snowflake1.png
HTTP/1.1" 304 0
::1 - - [19/Feb/2004:09:05:33 +0100] "GET /~hlk/security6.net/images/logo-1.png
HTTP/1.1" 304 0
2001:1448:81:beef:20a:95ff:fef5:34df - - [19/Feb/2004:09:57:35 +0100]
"GET / HTTP/1.1" 200 1456
2001:1448:81:beef:20a:95ff:fef5:34df - - [19/Feb/2004:09:57:35 +0100]
"GET /apache_pb.gif HTTP/1.1" 200 2326
2001:1448:81:beef:20a:95ff:fef5:34df - - [19/Feb/2004:09:57:36 +0100]
"GET /favicon.ico HTTP/1.1" 404 209
2001:1448:81:beef:20a:95ff:fef5:34df - - [19/Feb/2004:09:57:36 +0100]
"GET /favicon.ico HTTP/1.1" 404 209
```

Apache konfigureres nemt til at lytte på IPv6

Mange bruger HTTPD fra Apache projektet  
<http://httpd.apache.org> - netcraft siger omkring 70%

konfigureres gennem httpd.conf

```
Listen 0.0.0.0:80
Listen [::]:80
...
Allow from 127.0.0.1
Allow from 2001:1448:81:0f:2d:9ff:f86:3f
Allow from 217.157.20.133
```

# OpenBSD fast IPv6 adresse

Netværkskonfiguration på OpenBSD - flere filer:

```
# cat /etc/hostname.sk0
inet 10.0.0.23 0xffffffff00 NONE
inet6 2001:1448:81:30::2
# cat /etc/mygate
10.0.0.1
# grep 2001 /etc/rc.local
route add -inet6 default 2001:1448:81:30::1
# cat /etc/resolv.conf
domain security6.net
lookup file bind
nameserver 212.242.40.3
nameserver 212.242.40.51
nameserver 2001:1448:81:30::10
```

# Basal DNS opsætning

```
domain security6.net
nameserver 212.242.40.3
nameserver 212.242.40.51
nameserver 2001:1448:81::2
```

/etc/resolv.conf angiver navneservere og søgedomæner  
typisk indhold er domænenavn og IP-adresser for navneservere  
Filten opdateres også automatisk på DHCP klienter

**Husk at man godt kan slå AAAA records op over IPv4**

NB: denne fil kan hedde noget andet på UNIX varianter!

eksempelvis /etc/netsvc.conf

# DNS systemet

Navneopslag på Internet - centralet for IPv6

Tidligere brugte man en **hosts** fil  
hosts filer bruges stadig lokalt til serveren - IP-adresser

UNIX: /etc/hosts

Windows c:\windows\system32\drivers\etc\hosts

Eksempel: www.security6.net har adressen 217.157.20.131

skrives i database filer, zone filer

ns1	IN	A	217.157.20.130
	IN	AAAA	2001:618:433::1
www	IN	A	217.157.20.131
	IN	AAAA	2001:618:433::14

# Mere end navneopslag

består af resource records med en type:

- adresser A-records
- IPv6 adresser AAAA-records
- autoritative navneservere NS-records
- post, mail-exchanger MX-records
- flere andre: md , mf , cname , soa , mb , mg , mr , null , wks , ptr , hinfo , minfo , mx ....

IN	MX	10	mail.security6.net.
IN	MX	20	mail2.security6.net.

# BIND DNS server

Berkeley Internet Name Daemon server

Mange bruger BIND fra Internet Systems Consortium - altså Open Source

konfigureres gennem named.conf

det anbefales at bruge BIND version 9

- *DNS and BIND*, Paul Albitz & Cricket Liu, O'Reilly, 4th edition Maj 2001
- *DNS and BIND cookbook*, Cricket Liu, O'Reilly, 4th edition Oktober 2002

Kilde: <http://www.isc.org>

# BIND konfiguration - et udgangspunkt

```
acl internals { 127.0.0.1; ::1; 10.0.0.0/24; };
options {
    // the random device depends on the OS !
    random-device "/dev/random"; directory "/namedb";
    listen-on-v6 any;
    port 53; version "Dont know"; allow-query { any; };
};

view "internal" {
    match-clients { internals; }; recursion yes;
    zone "." {
        type hint; file "root.cache"; };
    // localhost forward lookup
    zone "localhost." {
        type master; file "internal/db.localhost";   };
    // localhost reverse lookup from IPv4 address
    zone "0.0.127.in-addr.arpa" {
        type master; file "internal/db.127.0.0"; notify no;   };
    ...
}
```

# Routing forståelse - IPv6

```
$ netstat -f inet6 -rn
```

Routing tables

Internet6:

Destination	Gateway	Flags	Netif
default	fe80::200:24ff:fec1:58ac	UGc	en0
::1	::1	UH	lo0
2001:1448:81:cf0f::/64	link#4	UC	en0
2001:1448:81:cf0f::1	0:0:24:c1:58:ac	UHLW	en0
fe80::/64	fe80::1	Uc	lo0
fe80:::1	link#1	UHL	lo0
fe80::/64	link#4	UC	en0
fe80::20d:93ff:fe28:2812	0:d:93:28:28:12	UHL	lo0
fe80::/64	link#5	UC	en1
fe80::20d:93ff:fe86:7c3f	0:d:93:86:7c:3f	UHL	lo0
ff01::/32	::1	U	lo0
ff02::/32	::1	UC	lo0
ff02::/32	link#4	UC	en0
ff02::/32	link#5	UC	en1

# IPv6 neighbor discovery protocol (NDP)

OSI	IPv4	IPv6
Network	IP / ICMP	IPv6 / ICMPv6
Link	ARP	
Physical	Physical	Physical

ARP er væk

NDP erstatter og udvider ARP, Sammenlign arp -an med ndp -an

Til dels erstatter ICMPv6 således DHCP i IPv6, DHCPv6 findes dog

**NB: bemærk at dette har stor betydning for firewallregler!**

# ARP vs NDP

```
h1k@bigfoot:basic-ipv6-new$ arp -an
```

```
? (10.0.42.1) at 0:0:24:c8:b2:4c on en1 [ethernet]  
? (10.0.42.2) at 0:c0:b7:6c:19:b on en1 [ethernet]
```

```
h1k@bigfoot:basic-ipv6-new$ ndp -an
```

Neighbor	Linklayer Address	Netif	Expires	St	Flgs	Prbs
::1	(incomplete)		lo0	permanent	R	
2001:16d8:fffd2:cf0f:21c:b3ff:fec4:e1b6	0:1c:b3:c4:e1:b6	en1	permanent	R		
fe80::1%lo0	(incomplete)		lo0	permanent	R	
fe80::200:24ff:fec8:b24c%en1	<b>0:0:24:c8:b2:4c</b>	en1	8h54m51s	S	R	
fe80::21c:b3ff:fec4:e1b6%en1	0:1c:b3:c4:e1:b6	en1	permanent	R		

Det er sagt mange gange at nu skal vi igang med IPv6

Der er sket store ændringer fra starten af 2007 til nu

Hvor det i starten af 2007 var status quo er flere begyndt at presse på

Selv på version2.dk omtales det <http://www.version2.dk/artikel/6147>  
*Seks DNS-rodservere tænder for IPv6 ICANN har nu aktiveret IPv6 på seks af inter-  
nettets 13 rodservere. Med det rigtige udstyr kan man nu køre helt uden om IPv4.*

# Hvorfor implementere IPv6 i jeres netværk?

## Addresserummet

- end-to-end transparency
- nemmere administration

## Autoconfiguration

- stateless autoconfiguration
- automatisk routerconfiguration! (router renumbering)

## Performance

- simplere format
- kortere behandlingstid i routere

## Fleksibilitet - generelt

## Sikkerhed

- IPsec er et krav!
- Afsender IP-adressen ændres ikke igennem NAT!

# Hvorfor migrere til IPv6?

IPv4 er mere end 25 år gammel - fra 1981 til idag

Idag har folk ønsker/krav til kommunikationen

- båndbredde
- latency
- Quality-of-service
- sikkerhed

Meget af dette er, eller kan, implementeres med IPv4 - men det bliver lappeløsninger

NB: IPv6 er designet til at løse SPECIFIKKE problemer

## The Internet has done this before!

Because all hosts can not be converted to TCP simultaneously, and some will implement only IP/TCP, it will be necessary to provide temporarily for communication between NCP-only hosts and TCP-only hosts. To do this certain hosts which implement both NCP and IP/TCP will be designated as relay hosts. These relay hosts will support Telnet, FTP, and Mail services on both NCP and TCP. These relay services will be provided beginning in November 1981, and will be fully in place in January 1982.

Initially there will be many NCP-only hosts and a few TCP-only hosts, and the load on the relay hosts will be relatively light. As time goes by, and the conversion progresses, there will be more TCP capable hosts, and fewer NCP-only hosts, plus new TCP-only hosts. But, presumably most hosts that are now NCP-only will implement IP/TCP in addition to their NCP and become "dual protocol" hosts. So, while the load on the relay hosts will rise, it will not be a substantial portion of the total traffic.

NCP/TCP Transition Plan November 1981 RFC-801

## Er IPv6 klar? - Korte svar - ja

Det bruges idag aktivt, især i dele af verden der ikke har store dele af v4 adresserummet

Kernen af IPv6 er stabil

IPv6 er inkluderet i mange operativsystemer idag

AIX, Solaris, BSD'erne, Linux, Mac OS X og Windows XP Cisco IOS, Juniper Networks  
Juniper har haft hardware support for IPv6 i mange år!

IPv6 TCP/IP stakke til indlejrede systemer er klar

prøv at lave `ping6 ::1` på jeres maskiner - det er IPv6

Se listen over IPv6 implementationer på <http://playground.sun.com/ipv6/ipng-implementations.html>

# IPv6 bruges idag

Listen over brugere vokser konstant

Store nye netværk designes alle med IPv6 en liste kan eksempelvis ses på addressen:  
<http://www.ipv6.ac.uk/gtpv6/eu.html>

Andre links kan vise statistik for internet og IPv4/IPv6

<http://www.bgpexpert.com/addrspace2007.php>

<https://wiki.caida.org/wiki/iic/bin/view/Main/WebHome>

<http://bgp.he.net/ipv6-progress-report.cgi>

Se også: <http://www.eu.ipv6tf.org/>

# 5 dårlige argumenter for ikke at bruge IPv6 nu

Det er ikke færdigt

- IPv4 har ALDRIG været færdigt ;-)

Ikke nødvendigt

- man kan stikke hovedet i busken

NAT løser alle problemer og er meget sikkert ...

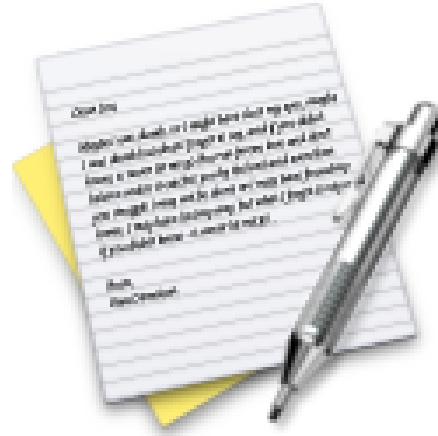
- NAT er en lappeløsning

Udskiftning af HELE infrastrukturen er for dyrt

- man opgraderer/udskifter jævnligt udstyr

Vent til det er færdigt!

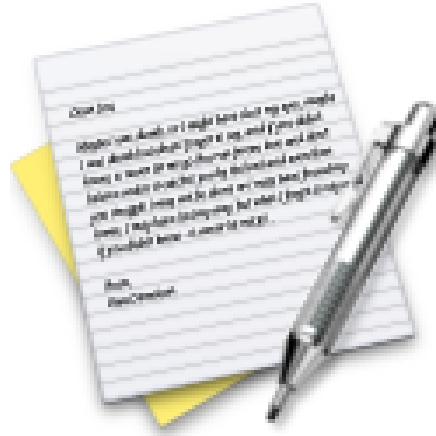
- man mister muligheden for at påvirke resultatet!



Vi laver nu øvelsen

## ping6 og traceroute6

som er øvelse **15** fra øvelseshæftet.



Vi laver nu øvelsen

## DNS og navneopslag - IPv6

som er øvelse **16** fra øvelseshæftet.

# Færdig med IPv6

I resten af kurset vil vi ikke betragte IPv6 eller IPv4 som noget specielt

Vi vil indimellem bruge det ene, indimellem det andet

Alle subnets er konfigureret ens på IPv4 og IPv6

Subnets som i IPv4 hedder prefix.45 vil således i IPv6 hedde noget med prefix:45:

At have ens routing på IPv4 og IPv6 vil typisk IKKE være tilfældet i praksis

Man kan jo lige så godt forbedre netværket mens man går over til IPv6 :-)

# Nu skal vi til management og diagnosticering

Always check the spark plugs!

Når man skal spore fejl i netværk er det essentielt at starte fra bunden:

- Er der link?
- Er der IP-adresse?
- Er der route?
- Modtager systemet pakker
- Er der en returvej fra systemet! Den her kan snyde mange!
- Lytter serveren på den port man vil forbinde til, UDP/TCP

Hvis der ikke er link vil man aldrig få svar fra databasen/webserveren/postserveren

De vigtigste kommandoer til udtræk af netværkskonfigurationen:

- cat - til at vise tekstfiler
- ifconfig - interface configuration
- netstat - network statistics
- lsof - list open files

# Basale testværktøjer TCP - Telnet og OpenSSL

Telnet blev tidligere brugt til login og er en klartekst forbindelse over TCP

Telnet kan bruges til at teste forbindelsen til mange ældre serverprotokoller som benytter ASCII kommandoer

- telnet mail.kramse.dk 25 laver en forbindelse til port 25/tcp
- telnet www.kramse.dk 80 laver en forbindelse til port 80/tcp

Til krypterede forbindelser anbefales det at teste med openssl

- openssl s\_client -host www.kramse.dk -port 443  
laver en forbindelse til port 443/tcp med SSL
- openssl s\_client -host mail.kramse.dk -port 993  
laver en forbindelse til port 993/tcp med SSL

Med OpenSSL i client-mode kan services tilgås med samme tekstkommandoer som med telnet

## Basale testværktøjer UDP

UDP er lidt drilsk, for de fleste services er ikke *ASCII protokoller*

Der findes dog en række testprogrammer, a la ping

- nsPing - name server ping
- dhcpping - dhcp server ping
- ...

Derudover kan man bruge de sædvanlige programmer som host til navneopslag osv.



IP har eksisteret mange år

Vi har udskiftet langsomme forbindelser med hurtige forbindelser

Vi har udskiftet langsomme MHz maskiner med Quad-core GHz maskiner

IP var tidligere meget konservativt, for ikke at overbelaste modtageren

Billedet er en HP arbejdsstation med 19" skærm og en 60MHz HP PA-RISC processor

# Anbefalet netværkstuning - hvad skal tunes

Der er visse indstillinger som tidligere var standard, de bør idag slås fra

En del er allerede tunet i nyere versioner af IP-stakkene, men check lige

Ideer til ting som skal slås fra:

- broadcast ICMP, undgå smurfing
- Source routing, kan måske omgå firewalls og filtre

Ideer til ting som skal slås til/ændres:

- Bufferstørrelser - hvorfor have en buffer på 65535 bytes på en maskine med 32GB ram?
- Nye funktioner som RFC-1323 TCP Extensions for High Performance

Det anbefales at finde leverandørens vejledning til hvad der kan tunes

# Netværkskonfiguration med sysctl

```
# tuning
net.inet.tcp.recvspace=65535
net.inet.tcp.sendspace=65535
net.inet.udp.recvspace=65535
net.inet.udp.sendspace=32768
# postgresql tuning
kern.seminfo.semnni=256
kern.seminfo.semnnm=2048
kern.shminfo.shmmax=50331648
```

På mange UNIX varianter findes et specielt tuningsprogram, sysctl

Findes blandt andet på alle BSD'erne: FreeBSD, OpenBSD, NetBSD og Darwin/OSX

Ændringerne skrives ind i filen /etc/sysctl.conf

På Linux erstatter det til dels konfiguration med echo

```
echo 1 > /proc/net/ip/forwarding
```

På AIX benyttes kommandoen network options no

Hvad er flaskehalsen for programmet?

I/O bundet - en enkelt disk eller flere

CPU bundet - regnekraften

Netværket - 10Mbit half-duplex adapter

Memory - begynder systemet at *swappe* eller *thrasher*

brug top og andre statistikprogrammer til at se disse data

## Måling af throughput

Når der skal tunes er det altid nødvendigt med en baseline

Man kan ikke begynde at tune ud fra subjektive målinger

*Det kører langsomt, Svartiden er for høj*

Målinger der giver præcise tal er nødvendige, før og efter målinger!

Der findes et antal værktøjer til, blandt andet Iperf

# Målinger med Iperf

```
hlk@fluffy:hlk$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
[ 4] local 10.0.42.23 port 5001 connected with 10.0.42.67 port 51148
[ 4] 0.0-10.2 sec 6.95 MBytes 5.71 Mbits/sec
[ 4] local 10.0.42.23 port 5001 connected with 10.0.42.67 port 51149
[ 4] 0.0-10.2 sec 7.02 MBytes 5.76 Mbits/sec
```

Ovenstående er set fra server, client kaldes med iperf -c fluffy

# Stop - vi prøver i fællesskab Iperf



Vi prøver lige Iperf sammen

hvis alle prøver samtidig giver det stor variation i resultaterne

# Apache benchmark og andre programmer



```
hlk@bigfoot:hlk$ ab -n 100 http://www.kramse.dk/
This is ApacheBench, Version 2.0.41-dev <$Revision: 1.121.2.12 $> apache-2.0
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright (c) 2006 The Apache Software Foundation, http://www.apache.org/
```

Benchmarking www.kramse.dk (be patient) ...

...

Der findes specialiserede værktøjer til mange protokoller

Eksempelvis følger der et apache benchmark med Apache HTTPD serveren

Mange andre værktøjer til at simulere flere samtidige brugere

# Apache Benchmark output - 1

```
Server Software:           Apache
Server Hostname:          www.kramse.dk
Server Port:              80

Document Path:            /
Document Length:          7547 bytes

Concurrency Level:        1
Time taken for tests:    13.84924 seconds
Complete requests:        100
Failed requests:          0
Write errors:             0
Total transferred:        778900 bytes
HTML transferred:         754700 bytes
Requests per second:      7.64 #/sec (mean)
Time per request:         130.849 ms (mean)
Time per request:         130.849 ms (mean, across all concurrent requests)
Transfer rate:            58.08 Kbytes/sec received
```

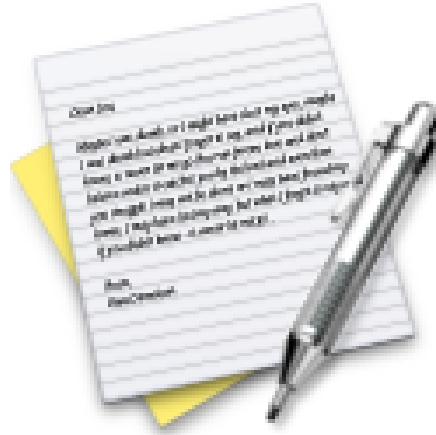
# Apache Benchmark output - 3

## Connection Times (ms)

	min	mean	+/-sd	median	max
Connect:	22	24	4.0	24	58
Processing:	96	105	33.0	99	421
Waiting:	63	71	32.7	65	386
Total:	119	130	33.5	124	446

## Percentage of the requests served within a certain time (ms)

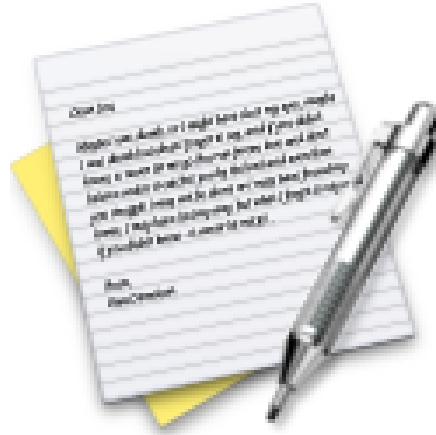
50%	124
66%	126
75%	128
80%	130
90%	143
95%	153
98%	189
99%	446
100%	446 (longest request)



Vi laver nu øvelsen

## Netværksinformation: sysctl

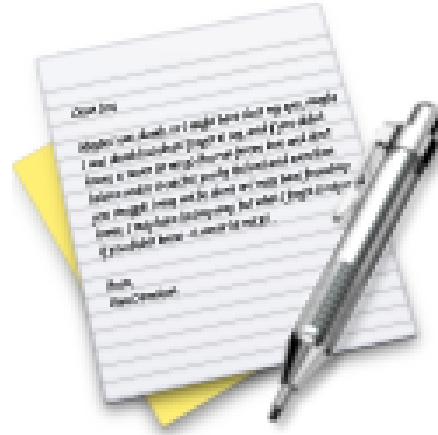
som er øvelse **17** fra øvelseshæftet.



Vi laver nu øvelsen

## Performance tool - iperf

som er øvelse **18** fra øvelseshæftet.



Vi laver nu øvelsen

## Afprøv Apache Benchmark programmet

som er øvelse **19** fra øvelseshæftet.

## Antal pakker per sekund

Til tider er det ikke båndbredden som sådan man vil måle

Specielt for routere er det vigtigt at de kan behandle mange pakker per sekund, pps

Til dette kan man lege med det indbyggede Ping program i flooding mode

Når programmet kaldes (som systemadministrator) med `ping -f server` vil den sende ping pakker så hurtigt som netkortet tillader

Programmer der kan teste pakker per sekund kaldes generelt for blaster tools

## traceroute

traceroute programmet virker ved hjælp af TTL

levetiden for en pakke tælles ned i hver router på vejen og ved at sætte denne lavt opnår man at pakken *timer ud* - besked fra hver router på vejen

default er UDP pakker, men på UNIX systemer er der ofte mulighed for at bruge ICMP

```
$ traceroute 217.157.20.129
```

```
traceroute to 217.157.20.129 (217.157.20.129),  
30 hops max, 40 byte packets  
1 safri (10.0.0.11) 3.577 ms 0.565 ms 0.323 ms  
2 router (217.157.20.129) 1.481 ms 1.374 ms 1.261 ms
```

# traceroute - med UDP

```
# tcpdump -i en0 host 217.157.20.129 or host 10.0.0.11
tcpdump: listening on en0
23:23:30.426342 10.0.0.200.33849 > router.33435: udp 12 [ttl 1]
23:23:30.426742 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.436069 10.0.0.200.33849 > router.33436: udp 12 [ttl 1]
23:23:30.436357 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437117 10.0.0.200.33849 > router.33437: udp 12 [ttl 1]
23:23:30.437383 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437574 10.0.0.200.33849 > router.33438: udp 12
23:23:30.438946 router > 10.0.0.200: icmp: router udp port 33438 unreachable
23:23:30.451319 10.0.0.200.33849 > router.33439: udp 12
23:23:30.452569 router > 10.0.0.200: icmp: router udp port 33439 unreachable
23:23:30.452813 10.0.0.200.33849 > router.33440: udp 12
23:23:30.454023 router > 10.0.0.200: icmp: router udp port 33440 unreachable
23:23:31.379102 10.0.0.200.49214 > safri.domain: 6646+ PTR?

200.0.0.10.in-addr.arpa. (41)
23:23:31.380410 safri.domain > 10.0.0.200.49214: 6646 NXDomain* 0/1/0 (93)
14 packets received by filter
0 packets dropped by kernel
```

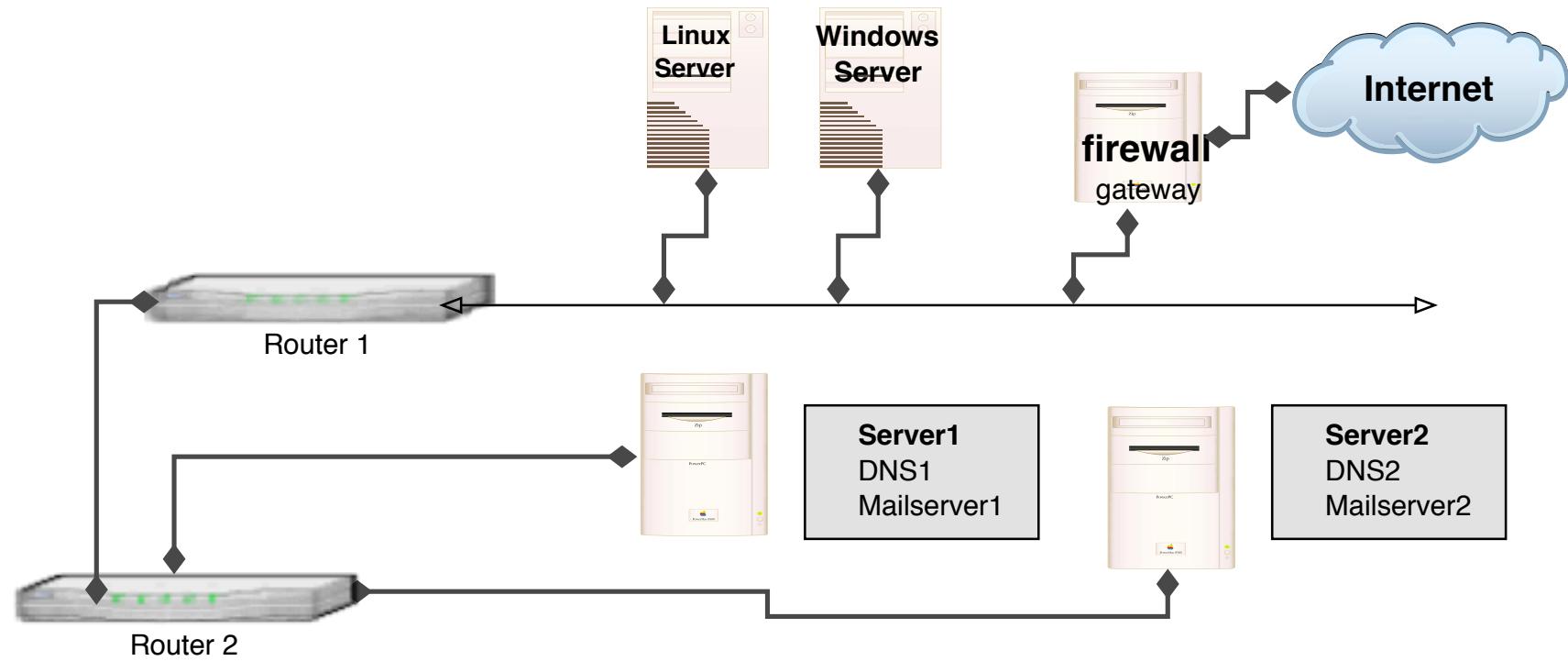
Diagnosticering af netværksproblemer - formålet med traceroute

Indblik i netværkets opbygning!

Svar fra hosts - en modtaget pakke fremfor et *sort hul*

Traceroute er ikke et angreb - det er også vigtigt at kunne genkende normal trafik!

# Network mapping



Ved brug af traceroute og tilsvarende programmer kan man ofte udlede topologien i det netværk man undersøger

## Flere traceprogrammer

mtr My traceroute - grafisk <http://www.bitwizard.nl/mtr/>

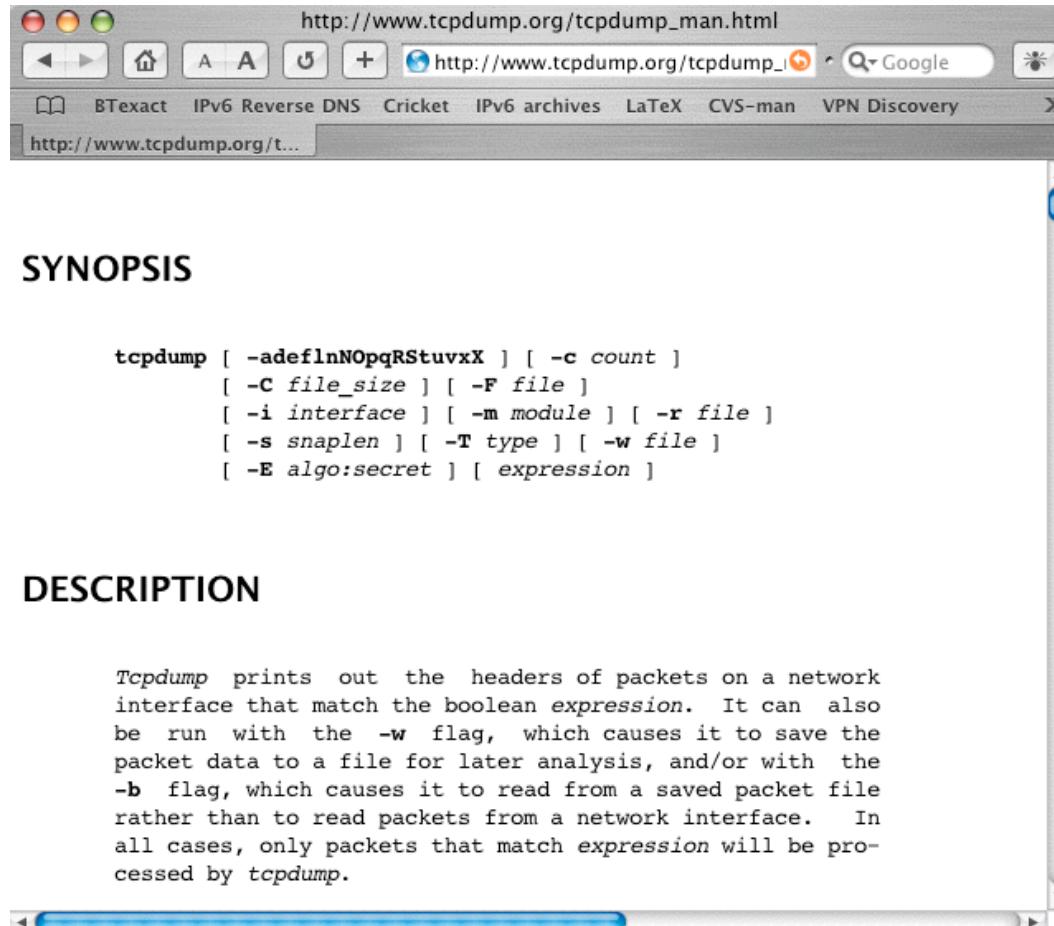
Ift - *layer four trace* benytter TCP SYN og FIN prober

trace ved hjælp af TCP og andre protokoller findes

paratrace - *Parasitic Traceroute via Established TCP Flows and IPID Hopcount*

Der findes webservices hvor man kan trace fra, eksempelvis: <http://www.samspade.org>

# TCPDUMP - protokolanalyse pakkesniffer



```
tcpdump [ -adeflnNOpqRStuvxxX ] [ -c count ]
[ -C file_size ] [ -F file ]
[ -i interface ] [ -m module ] [ -r file ]
[ -s snaplen ] [ -T type ] [ -w file ]
[ -E algo:secret ] [ expression ]
```

## DESCRIPTION

*Tcpdump* prints out the headers of packets on a network interface that match the boolean expression. It can also be run with the **-w** flag, which causes it to save the packet data to a file for later analysis, and/or with the **-b** flag, which causes it to read from a saved packet file rather than to read packets from a network interface. In all cases, only packets that match expression will be processed by *tcpdump*.

<http://www.tcpdump.org> - både til Windows og UNIX

# tcpdump - normal brug

- tekstmode
- kan gemme netværkspakker i filer
- kan læse netværkspakker fra filer
- er de-facto standarden for at gemme netværksdata i filer

```
[root@otto hlk]# tcpdump -i en0
tcpdump: listening on en0
13:29:39.947037 fe80::210:a7ff:fe0b:8a5c > ff02::1: icmp6: router advertisement
13:29:40.442920 10.0.0.200.49165 > dns1.cybercity.dk.domain: 1189+[|domain]
13:29:40.487150 dns1.cybercity.dk.domain > 10.0.0.200.49165: 1189 NXDomain*[|domain]
13:29:40.514494 10.0.0.200.49165 > dns1.cybercity.dk.domain: 24765+[|domain]
13:29:40.563788 dns1.cybercity.dk.domain > 10.0.0.200.49165: 24765 NXDomain*[|domain]
13:29:40.602892 10.0.0.200.49165 > dns1.cybercity.dk.domain: 36485+[|domain]
13:29:40.648288 dns1.cybercity.dk.domain > 10.0.0.200.49165: 36485 NXDomain*[|domain]
13:29:40.650596 10.0.0.200.49165 > dns1.cybercity.dk.domain: 4101+[|domain]
13:29:40.694868 dns1.cybercity.dk.domain > 10.0.0.200.49165: 4101 NXDomain*[|domain]
13:29:40.805160 10.0.0.200 > mail: icmp: echo request
13:29:40.805670 mail > 10.0.0.200: icmp: echo reply
...
```

# TCPDUMP syntaks - udtryk

filtrer til husbehov

- type - host, net og port
- src pakker med afsender IP eller afsender port
- dst pakker med modtager IP eller modtager port
- host - afsender eller modtager
- proto - protokol: ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp og udp

IP adresser kan angives som dotted-decimal eller navne

porte kan angives med numre eller navne

komplekse udtryk opbygges med logisk and, or, not

## tcpdump udtryk eksempler

Host 10.1.2.3

Alle pakker hvor afsender eller modtager er 10.1.2.3

host 10.2.3.4 and not host 10.3.4.5

Alle pakker til/fra 10.2.3.4 undtagen dem til/fra 10.3.4.5

- meget praktisk hvis man er logget ind på 10.2.3.4 via netværk fra 10.3.4.5

host foo and not port ftp and not port ftp-data

trafik til/fra maskine *foo* undtagen hvis det er FTP trafik

# Wireshark - grafisk pakkesniffer



The screenshot shows the official Wireshark website. At the top is a large blue header with the word "WIRESHARK" in white. Below it is a black navigation bar with links for "HOME", "ABOUT", "WHAT'S NEW", "DOWNLOAD", and "FAQ". To the right of the header is a background image of a shark swimming in water. On the left side, there's a sidebar with a dark blue background containing links under categories like "Get It", "Get Help", "Develop", and "Products". The main content area has several sections: "Sniffing Problems A Mile Away" (explaining the name change from Ethereal), "News" (about the release of version 0.99.3), and a "Download Now" section for version 0.99.3. There's also a Q&A section about capturing 802.11 traffic.

**Sniffing Problems A Mile Away**

The Ethereal network protocol analyzer has changed its name to Wireshark.

The name might be new, but the software is the same. Wireshark's powerful features make it the tool of choice for network troubleshooting, protocol development, and education worldwide.

Wireshark was written by networking experts around the world, and is an example of the power of open source. It runs on Windows, Linux, UNIX, and other platforms.

**News**

**Wireshark 0.99.3 Released**

Aug 23, 2006

Wireshark 0.99.3 has been released. Security-related vulnerabilities in the SCSI, DHCP, ESP, and Q.2931 dissectors have been fixed. See the [advisory](#) for details.

**Download Now**

0.99.3

**Q:**  
How do I capture 802.11 traffic on Windows?

**A:**  
**AirPcap**

<http://www.wireshark.org>

både til Windows og UNIX, tidligere kendt som Ethereal

# Programhygiejne!

Download, installer - kør! - farligt!

Sådan gøres det:

- download program OG signaturfil/MD5
- verificer signatur eller MD5
- installer
- brug programmet
- hold programmet opdateret!

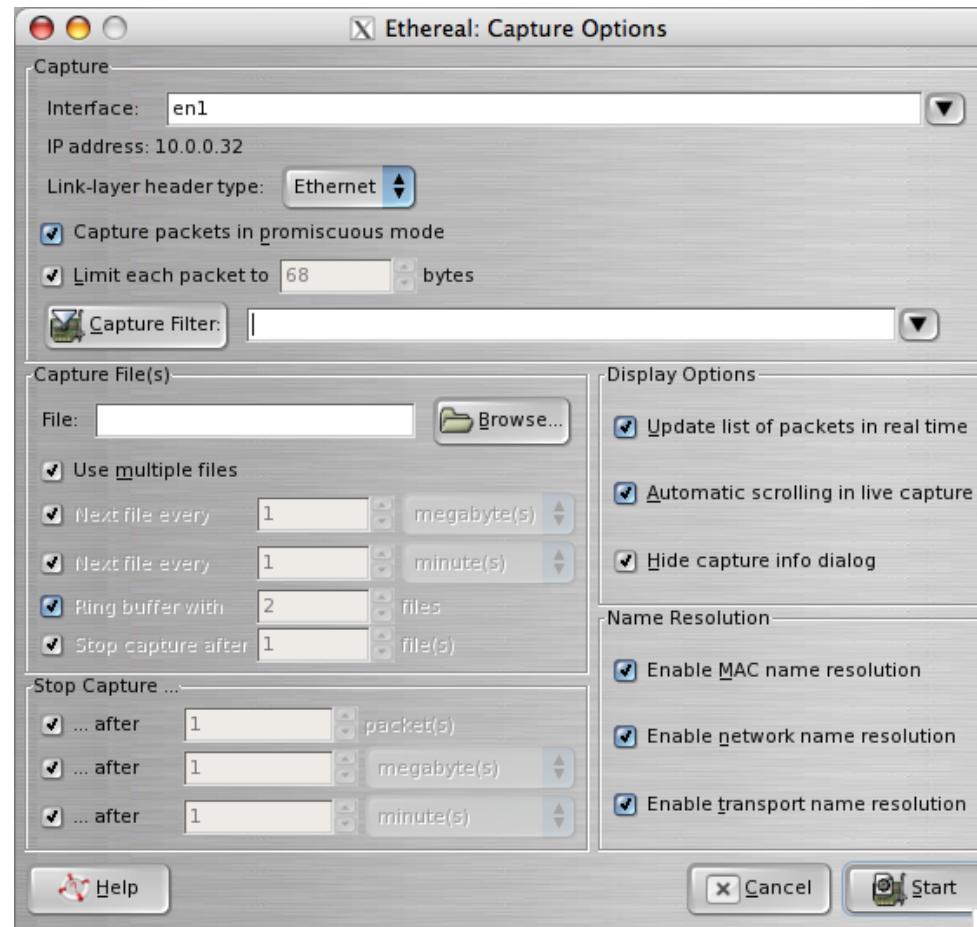
Se eksempelvis teksten på hjemmesiden:

*Wireshark 0.99.2 has been released. Several security-related vulnerabilities have been fixed and several new features have been added.*

NB: ikke alle programmer har signaturer :(

MD5 er en envejs hash algoritme - mere om det senere

# Brug af Wireshark



Man starter med Capture - Options

# Brug af Wireshark



The screenshot shows the Wireshark interface with the title bar "X (Untitled) - Ethereal". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The toolbar contains various icons for file operations, search, and analysis. The main pane displays a list of network frames. Frame 563 is selected, showing details: Source 10.0.0.32, Destination sunny.kramse.dk, Protocol TCP, and Info 54021 > imaps [ACK] Seq=426 Ack=11106 Win=65!. Below the list is a detailed description of the selected frame, including its type (Ethernet II), source and destination MAC addresses, and protocol stack information (Internet Protocol, Transmission Control Protocol). The bottom pane shows the raw hex and ASCII data of the selected frame.

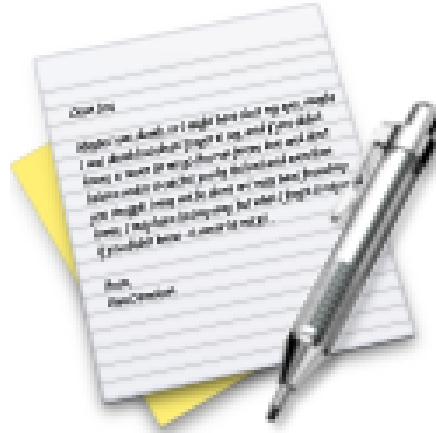
No.	Time	Source	Destination	Protocol	Info
561	0.700947	10.0.0.32	sunny.kramse.uk	TCP	54021 > imaps [ACK] Seq=426 Ack=11106 Win=65!
562	6.763144	sunny.kramse.dk	10.0.0.32	TLS	Continuation Data, [Unreassembled Packet]
563	6.820037	10.0.0.32	sunny.kramse.dk	TCP	54021 > imaps [ACK] Seq=426 Ack=11106 Win=65!
564	6.919635	10.0.0.32	sunny.kramse.dk	TCP	54023 > imaps [SYN] Seq=0 Ack=0 Win=65535 Len
565	6.921708	sunny.kramse.dk	10.0.0.32	TCP	imaps > 54023 [SYN, ACK] Seq=0 Ack=1 Win=1638
566	6.921794	10.0.0.32	sunny.kramse.dk	TCP	54023 > imaps [ACK] Seq=1 Ack=1 Win=65535 Len
567	6.922614	10.0.0.32	sunny.kramse.dk	TLS	Client Hello

Frame 563 (66 bytes on wire, 66 bytes captured)  
Ethernet II, Src: AppleCom\_86:7c:3f (00:0d:93:86:7c:3f), Dst: Olicom\_c3:57:d8 (00:00:24:c3:57:d8)  
Internet Protocol, Src: 10.0.0.32 (10.0.0.32), Dst: sunny.kramse.dk (217.157.20.131)  
Transmission Control Protocol, Src Port: 54021 (54021), Dst Port: imaps (993), Seq: 426, Ack: 11106, Len: 0

0000 00 00 24 c3 57 d8 00 0d 93 86 7c 3f 08 00 45 00 ..\$.W...|?.E.  
0010 00 34 7e 8b 40 00 40 06 c3 f8 0a 00 00 20 d9 9d .4~.@@. .... ..  
0020 14 83 d3 05 03 e1 cd 31 c9 ea 0d 7b a2 bf 80 10 .....1 ...{....  
0030 ff ff 32 0d 00 00 01 01 08 0a 62 e0 c3 42 bb e3 ..2.....b..B..

Filter: Expression... Clear Apply File: "/var/tmp/ether0ARkxt..."

Læg mærke til filtermulighederne



Vi laver nu øvelsen

## TCP/IP og sniffere

som er øvelse **20** fra øvelseshæftet.

# syslog

Security

.net

syslog er system loggen på UNIX og den er effektiv

- man kan definere hvad man vil se og hvor man vil have det dirigeret hen
- man kan samle det i en fil eller opdele alt efter programmer og andre kriterier
- man kan ligeledes bruge named pipes - dvs filer i filesystemet som tunneller fra chroot'ed services til syslog i det centrale system!
- man kan nemt sende data til andre systemer

Hvis man vil lave en centraliseret løsning er følgende link vigtigt:

Tina Bird, Counterpane

<http://loganalysis.org>

## syslogd.conf eksempel

```
*.err;kern.debug;auth.notice;authpriv.none;mail.crit      /dev/console
*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none   /var/log/messages
kern.debug;user.info;syslog.info                           /var/log/messages
auth.info                                                 /var/log/authlog
authpriv.debug                                           /var/log/secure
...
# Uncomment to log to a central host named "loghost".
#*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none   @loghost
#kern.debug,user.info,syslog.info                          @loghost
#auth.info,authpriv.debug,daemon.info                     @loghost
```

# Andre syslogs syslog-ng

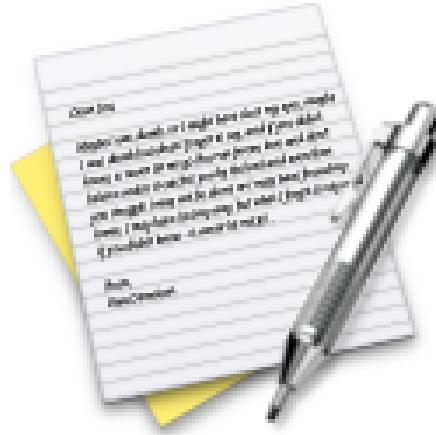
der findes andre syslog systemer eksempelvis syslog-ng

konfigureres gennem /etc/syslog-nginx/syslog-nginx.conf

Eksempel på indholdet af filen kunne være:

```
options {
    long_hostnames(off);
    sync(0);
    stats(43200);
};

source src unix-stream("/dev/log"); internal(); pipe("/proc/kmsg"); ;
destination messages file("/var/log/messages"); ;
destination console_all file("/dev/console"); ;
log source(src); destination(messages); ;
log source(src); destination(console_all); ;
```



Vi laver nu øvelsen

## Logning med syslogd og syslog.conf

som er øvelse **21** fra øvelseshæftet.

# Logfiler og computer forensics

Logfiler er en nødvendighed for at have et transaktionsspor

Logfiler er desuden nødvendige for at fejlfinde

Det kan være relevant at sammenholde logfiler fra:

- routere
- firewalls
- intrusion detection systemer
- adgangskontrolsystemer
- ...

Husk - tiden er vigtig! Network Time Protocol (NTP) anbefales

Husk at logfilerne typisk kan slettes af en angriber - hvis denne får kontrol med systemet

# Simple Network Management Protocol



SNMP er en protokol der supporteres af de fleste professionelle netværksenheder, såsom switcher, routere

hosts - skal slås til men følger som regel med

SNMP bruges til:

- *network management*
- statistik
- rapportering af fejl - SNMP traps

**sikkerheden baseres på community strings der sendes som klartekst ...**

det er nemmere at brute-force en community string end en brugerid/kodeord kombination

## Simple Network Management Protocol

sikkerheden afhænger alene af en Community string SNMPv2

typisk er den nem at gætte:

- public - default til at aflæse statistik
- private - default når man skal ændre på enheden, skrive
- cisco
- ...

Der findes lister og ordbøger på nettet over kendte default communities

## Systemer med SNMP

kan være svært at finde ... det er UDP 161

Hvis man finder en så prøv at bruge **snmpwalk** programmet - det kan vise alle tilgængelige SNMP oplysninger fra den pågældende host

det kan være en af måderne at identificere uautoriserede WLAN Access Points på - sweep efter port 161/UDP

snmpwalk er et af de mest brugte programmer til at hente snmp oplysninger - i forbindelse med hackning og penetrationstest

## snmpwalk

Typisk brug er:

```
snmpwalk -v 1 -c secret switch1
```

```
snmpwalk -v 2c -c secret switch1
```

Eventuelt bruges snmpget og snmpset

Ovenstående er en del af Net-SNMP pakken, <http://net-snmp.sourceforge.net/>



Vi laver nu øvelsen

## SNMP walk

som er øvelse **22** fra øvelseshæftet.

hvad betyder bruteforcing?  
afprøvning af alle mulighederne

Hydra v2.5 (c) 2003 by van Hauser / THC <vh@thc.org>

Syntax: hydra [[[ -l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]]  
[-o FILE] [-t TASKS] [-g TASKS] [-T SERVERS] [-M FILE] [-w TIME]  
[-f] [-e ns] [-s PORT] [-S] [-vV] server service [OPT]

Options:

-S	connect via SSL
-s PORT	if the service is on a different default port, define it here
-l LOGIN	or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS	or -P FILE try password PASS, or load several passwords from FILE
-e ns	additional checks, "n" for null password, "s" try login as pass
-C FILE	colon seperated "login:pass" format, instead of -L/-P option
-M FILE	file containing server list (parallizes attacks, see -T)
-o FILE	write found login/password pairs to FILE instead of stdout

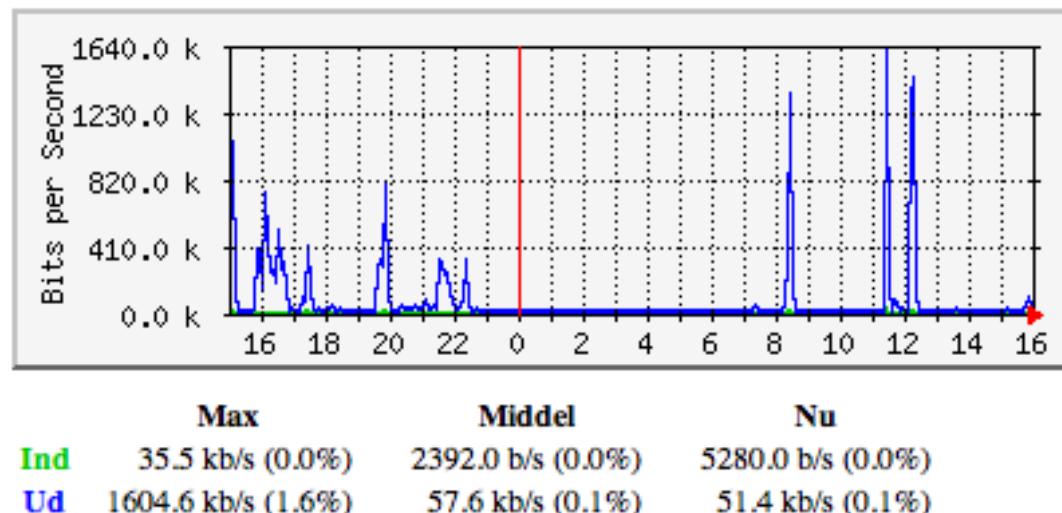
...

# Eksempler på SNMP og management

Ofte foregår administration af netværksenheder via HTTP, Telnet eller SSH

- små dumme enheder er idag ofte web-enablet
- bedre enheder giver både HTTP og kommandolinieadgang
- de bedste giver mulighed for SSH, fremfor Telnet

## 'Daglig' graf (5 minuts Middel)



Monitorering af SNMP enheder og grafer

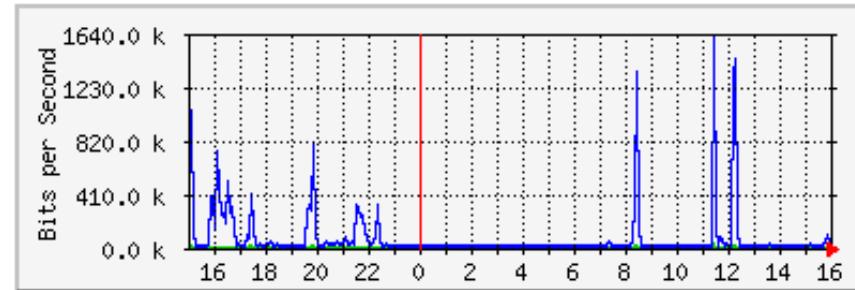
Inkluderer en nem configmaker og benytter idag RRDTool til data

Hjemmesiden: <http://oss.oetiker.ch/mrtg/>

# RRDTool Round Robin Database Tool



'Daglig' graf (5 minuts Middel)



	Max	Middel	Nu
Ind	35.5 kb/s (0.0%)	2392.0 b/s (0.0%)	5280.0 b/s (0.0%)
Ud	1604.6 kb/s (1.6%)	57.6 kb/s (0.1%)	51.4 kb/s (0.1%)

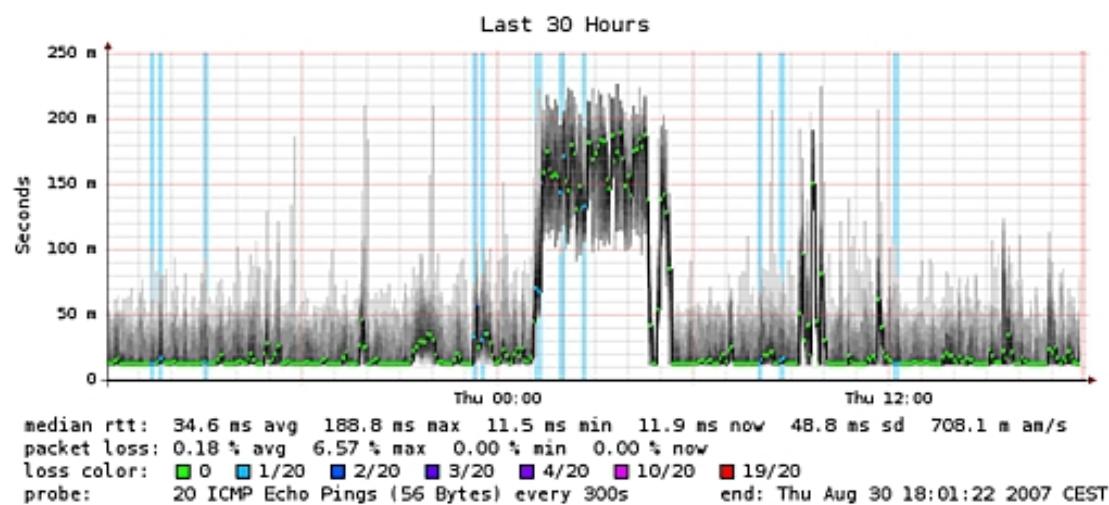
Round Robin Database Tool er en måde at gemme data på

Med RRDTool kan man derefter få lavet grafer

Typisk bruger man et andet værktøj som benytter RRDTool til data

<http://oss.oetiker.ch/rrdtool/doc/index.en.html>

Kan bruges til temperaturmålinger og alt muligt andet



Måling af latency for netværksservice

Understøtter et stort antal prober: ICMP, DNS, HTTP, LDAP, SMTP, ...

Min SmokePing server <http://pumba.kramse.dk/smokeping/>

Hjemmesiden for SmokePing <http://oss.oetiker.ch/smokeping/>

Lavet af Tobias Oetiker og Niko Tyni

Overvågningsværktøj der giver godt overblik

- Monitoring af diverse services (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Monitoring af host resources (processor load, disk and memory usage, running processes, log files, etc.)
- Monitoring af andre ressourcer som temperatur
- Simpel plugin design som gør det nemt at udvide
- Kan sende e-mail, SMS m.v.

Benyttes mange steder

Hjemmesiden for Nagios <http://www.nagios.org/>

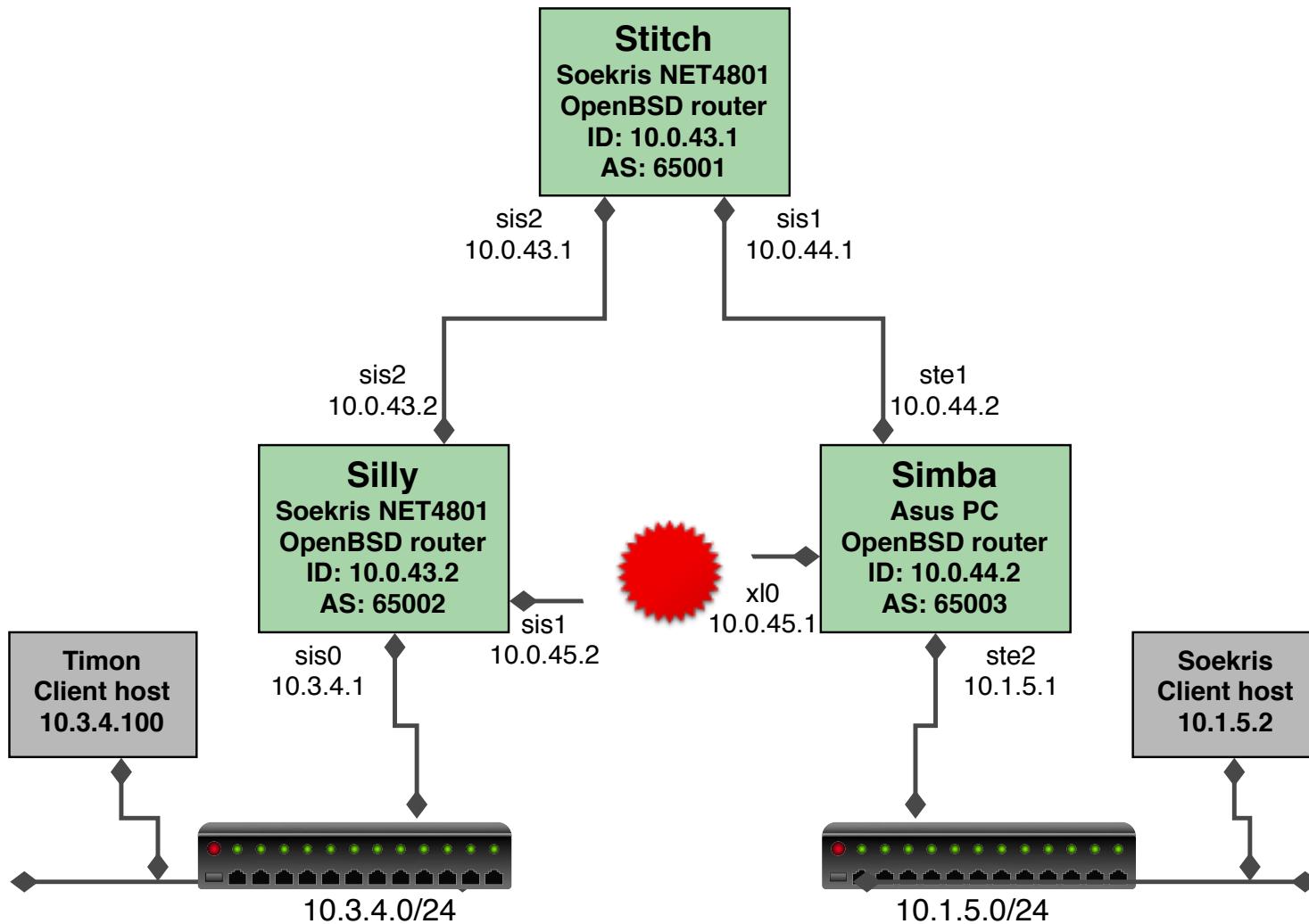
# Stop - overvågningsværktøjer

Brug lidt tid på at se på vores netværk

Valgfrit om I vil se på Administrationsinterface på switcher, SNMP indstillinger eksempelvis

Eller Nagios og SmokePing på mine servere

# Dag 3 Dynamiske protokoller og services



802.11 er arbejdsgruppen under IEEE

De mest kendte standarder idag indenfor trådløse teknologier:

- 802.11b 11Mbps versionen
- 802.11g 54Mbps versionen

Der er proprietære versioner 22Mbps og den slags

- det anbefales IKKE at benytte disse da det giver vendor lock-in - man bliver låst fast

Kilde: <http://grouper.ieee.org/groups/802/11/index.html>

## 802.11 modes og frekvenser

Access point kører typisk i *access point mode* også kaldet infrastructure mode - al trafik går via AP

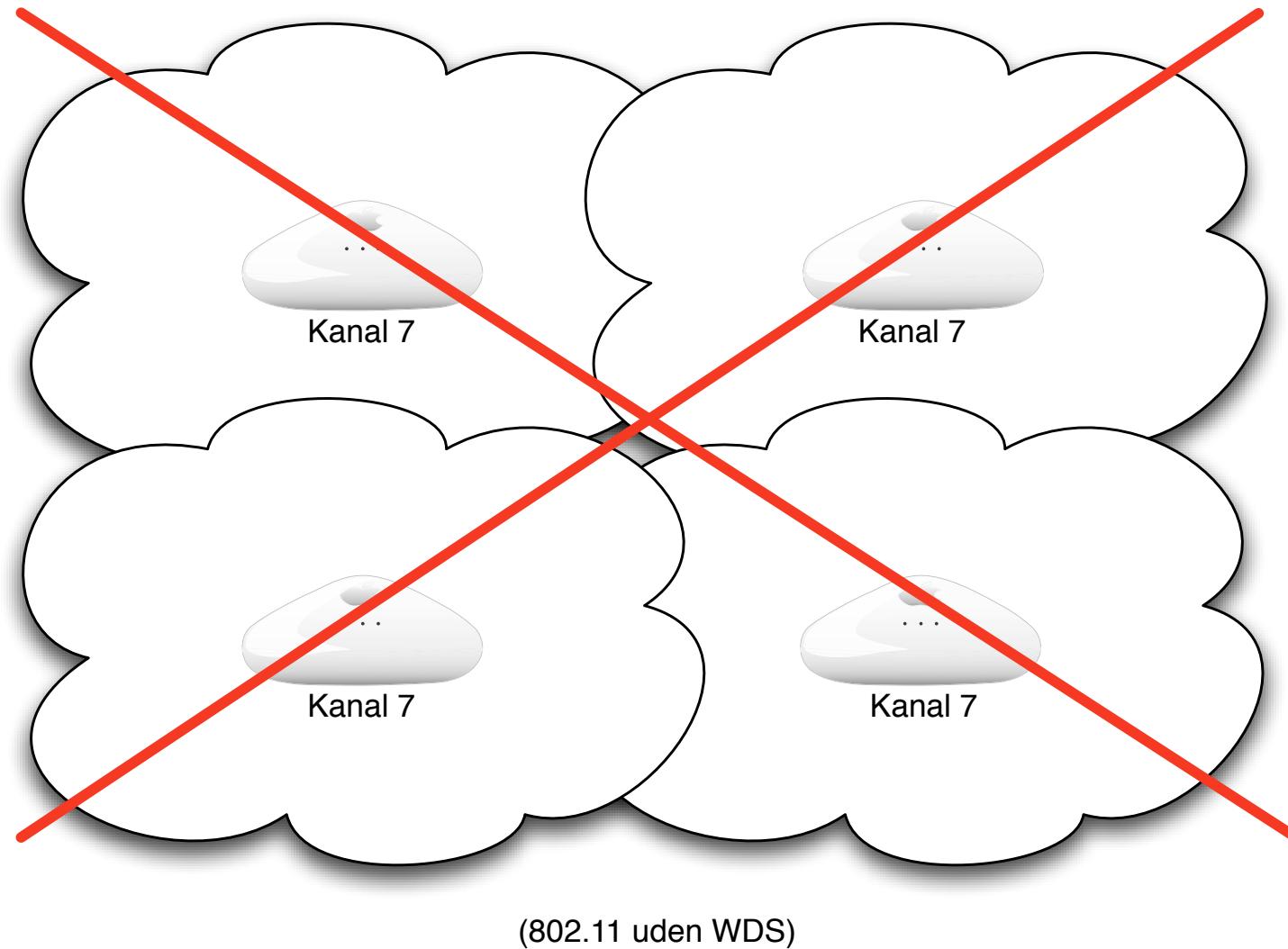
Alternativt kan wireless kort oprette ad-hoc netværk - hvor trafikken går direkte mellem netkort

Frekvenser op til kanal 11 og 12+13 i DK/EU

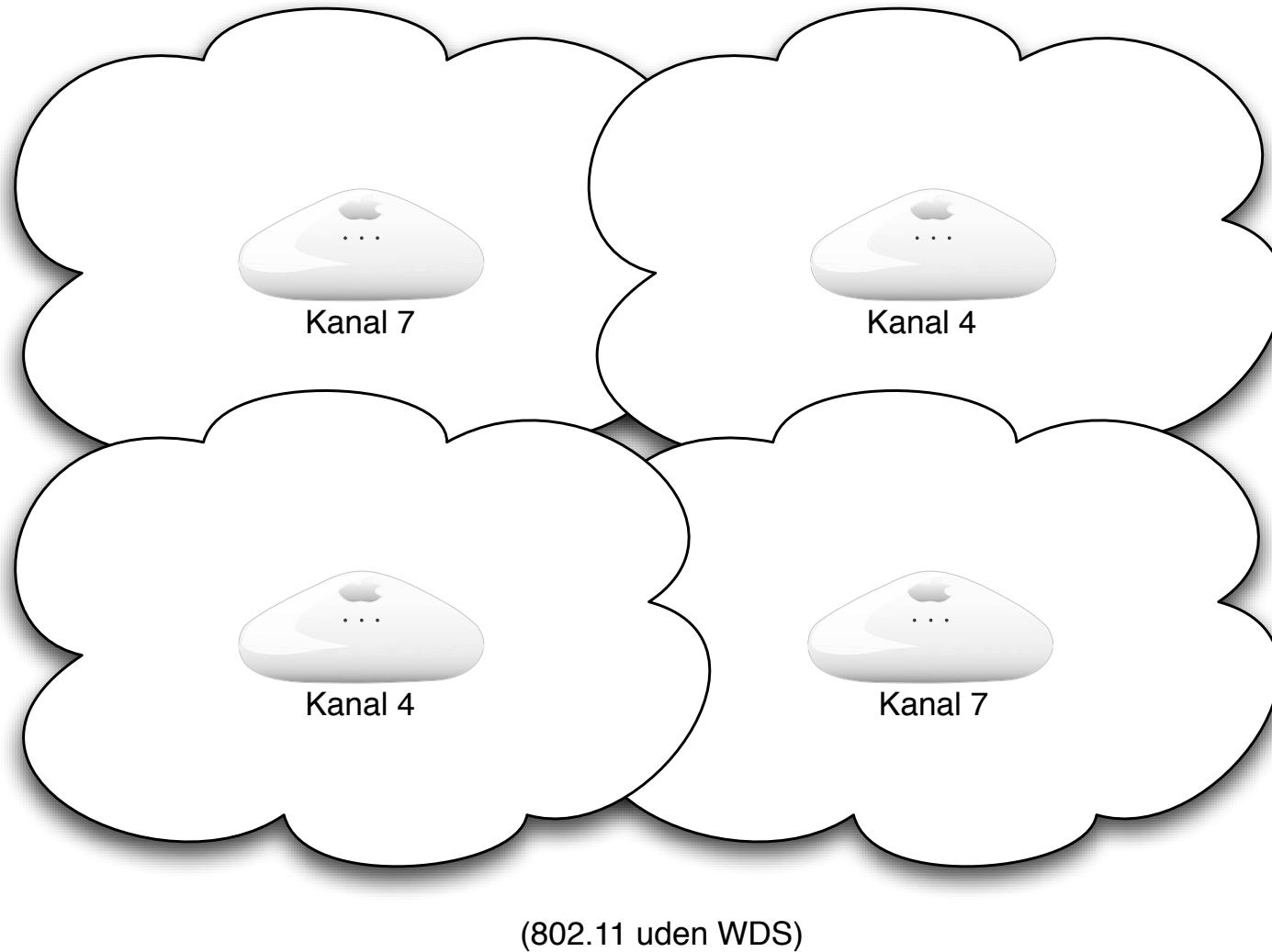
Høgst 2 kanaler spring for 802.11b AP der placeres indenfor rækkevidde

Høgst 4 kanaler spring for 802.11g AP der placeres indenfor rækkevidde

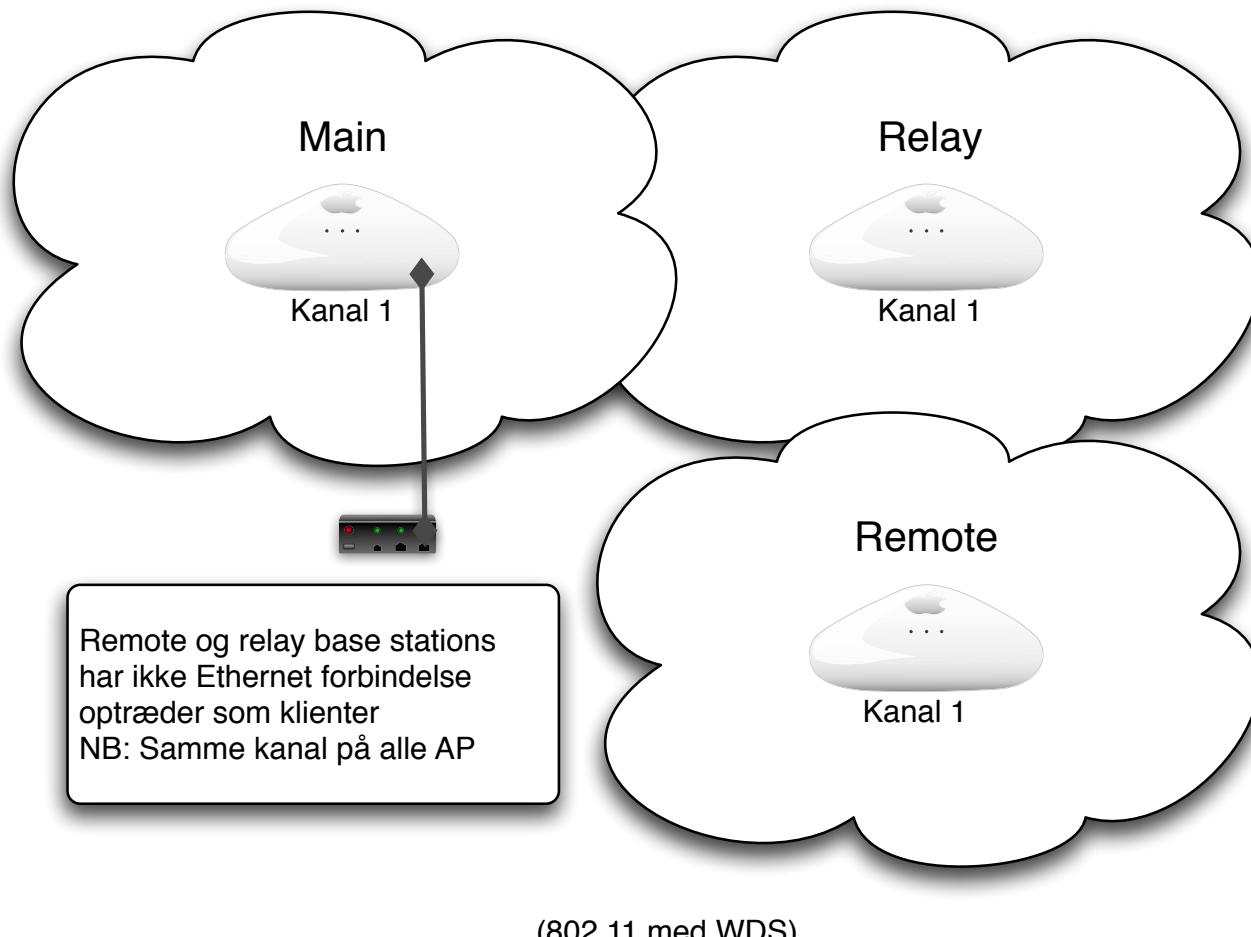
# Eksempel på netværk med flere AP'er



# Eksempel på netværk med flere AP'er



# Wireless Distribution System WDS



Se også: [http://en.wikipedia.org/wiki/Wireless\\_Distribution\\_System](http://en.wikipedia.org/wiki/Wireless_Distribution_System)

# Er trådløse netværk interessante?

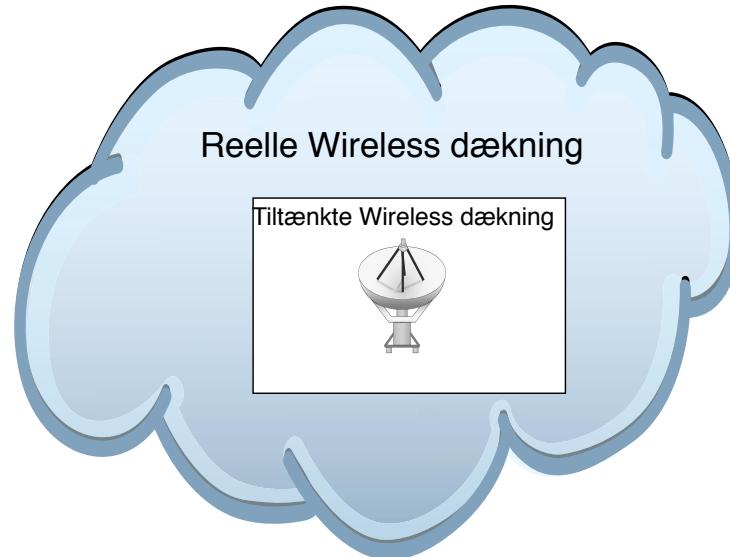


Sikkerhedsproblemer i de trådløse netværk er mange

- Fra lavt niveau - eksempelvis ARP, 802.11
- dårlige sikringsmekanismer - WEP
- dårligt udstyr - mange fejl
- usikkkerhed om implementering og overvågning

Trådløst udstyr er blevet meget billigt!

Det er et krav fra brugerne - trådløst er lækkert



- Værre end Internetangreb - anonymt
- Kræver ikke fysisk adgang til lokationer
- Konsekvenserne ved sikkerhedsbrud er generelt større
- Typisk får man direkte LAN eller Internet adgang!

Alle bruger nogenlunde de samme værktøjer, måske forskellige mærker

- Wirelessscanner - Kismet og netstumbler
- Wireless Injection - typisk på Linux
- ...
- Aircrack-ng

Jeg anbefaler Auditor Security Collection og BackTrack boot CD'erne

# Konsulentens udstyr wireless

Laptop med PC-CARD slot

Trådløse kort Atheros, de indbyggede er ofte ringe ;-)

Access Points - jeg anbefaler Airport Express

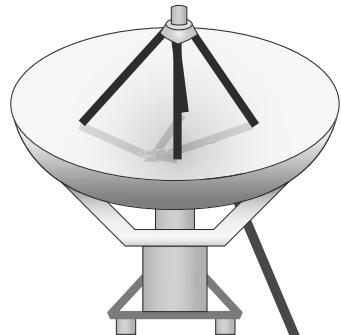
Antenner hvis man har lyst

Bøger:

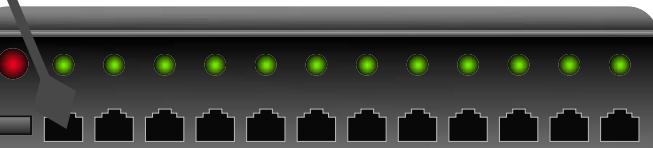
- *Real 802.11 security*
- Se oversigter over bøger og værktøjer igennem præsentationen:

Internetressourcer:

- Auditor Security Collection - CD image med Linux+værktøjer
- Packetstorm wireless tools <http://packetstormsecurity.org/wireless/>
- *Beginner's Guide to Wireless Auditing* David Maynor <http://www.securityfocus.com/infocus/1877?ref=rss>



## Wireless Access Point



## netværket - typisk Ethernet

et access point - forbindes til netværket

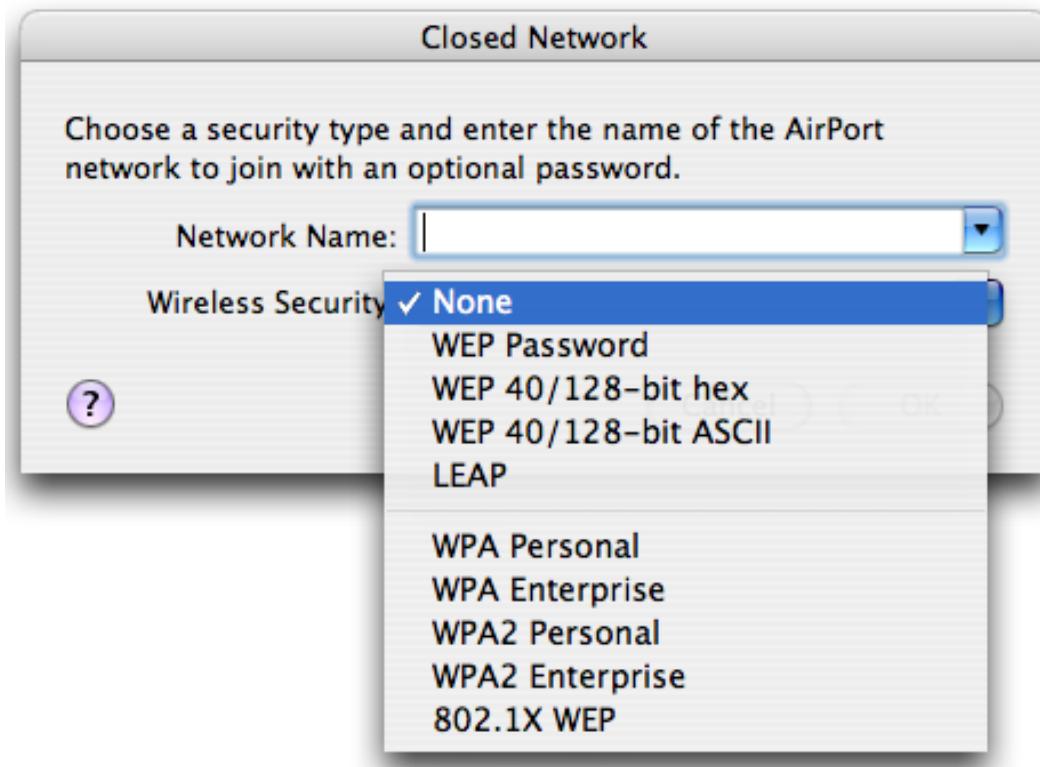
# Basal konfiguration

Når man tager fat på udstyr til trådløse netværk opdager man:

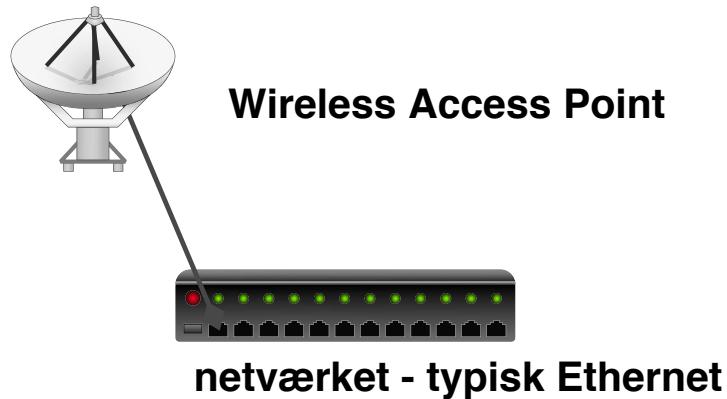
SSID - nettet skal have et navn

frekvens / kanal - man skal vælge en kanal, eller udstyret vælger en automatisk  
der er nogle forskellige metoder til sikkerhed

# Trådløs sikkerhed



- Trådløs sikkerhed - WPA og WPA2
- Nem konfiguration
- Nem konfiguration af Access Point



Sikkerheden er baseret på nogle få forudsætninger

- SSID - netnavnet
- WEP kryptering - Wired Equivalent Privacy
- måske MAC flitrering, kun bestemte kort må tilgå accesspoint

Til gengæld er disse forudsætninger ofte ikke tilstrækkelige ...

- WEP er måske *ok* til visse små hjemmenetværk
- WEP er baseret på en DELT hemmelighed som alle stationer kender
- nøglen ændres sjældent, og det er svært at distribuere en ny

# Forudsætninger

Til gengæld er disse forudsætninger ofte ikke tilstrækkelige ...

Hvad skal man beskytte?

Hvordan kan sikkerheden omgås?

Mange firmaer og virksomheder stille forskellige krav til sikkerheden - der er ikke en sikkerhedsmekanisme der passer til alle

## SSID - netnavnet

Service Set Identifier (SSID) - netnavnet

32 ASCII tegn eller 64 hexadecimale cifre

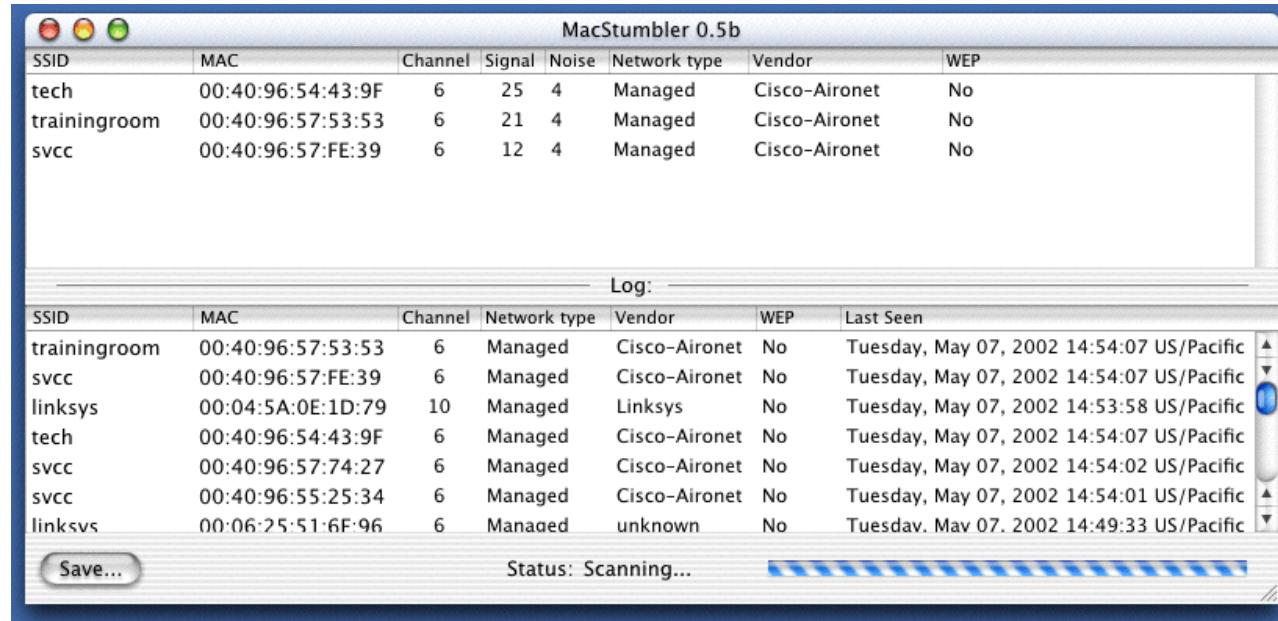
Udstyr leveres typisk med et standard netnavn

- Cisco - tsunami
- Linksys udstyr - linksys
- Apple Airport, 3Com m.fl. - det er nemt at genkende dem

SSID kaldes også for NWID - network id

SSID broadcast - udstyr leveres oftest med broadcast af SSID

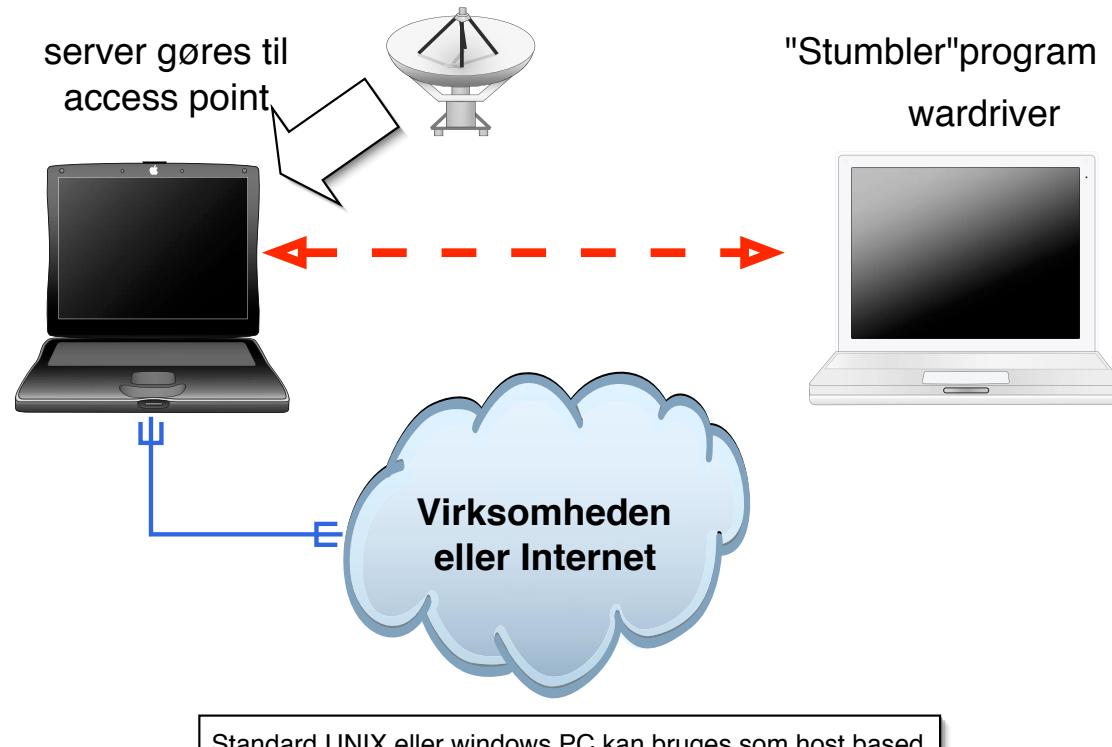
# Demo: wardriving med stumbler programmer



man tager et trådløst netkort og en bærbar computer og noget software:

- Netstumbler - Windows <http://www.netstumbler.com>
- dstumbler - UNIX <http://www.dachb0den.com/projects/dstumbler.html>
- iStumbler - Mac <http://www.istumbler.net/>
- Kismet ... mange andre

# Start på demo - wardriving



- Almindelige laptops bruges til demo
- Der startes et *access point*

## MAC filtrering

De fleste netkort tillader at man udskifter sin MAC adresse

MAC adressen på kortene er med i alle pakker der sendes

MAC adressen er aldrig krypteret, for hvordan skulle pakken så nå frem?

MAC adressen kan derfor overtages, når en af de tilladte stationer forlader området ...

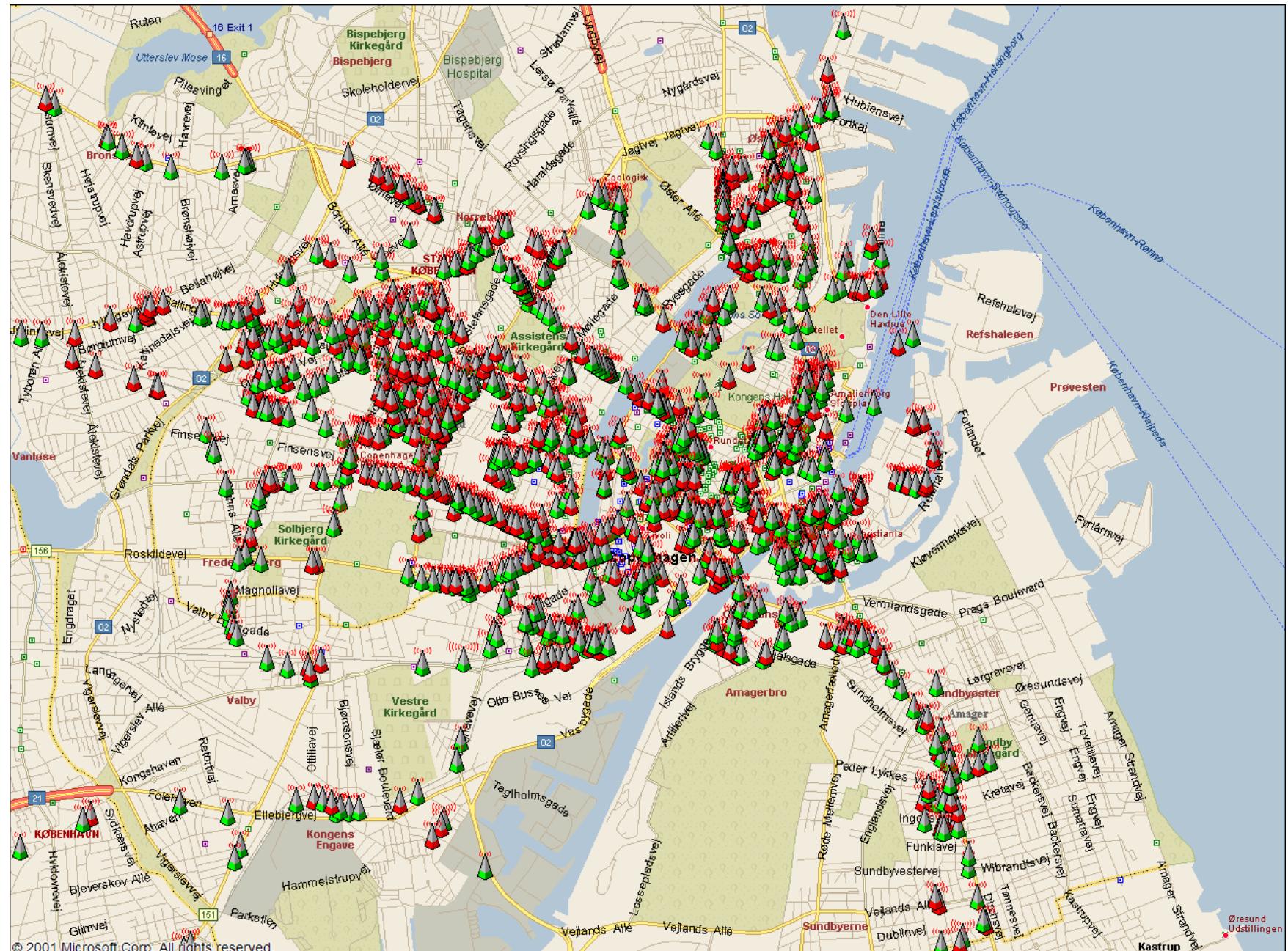
# Resultater af wardriving

Hvad opdager man ved wardriving?

- at WEP IKKE krypterer hele pakken
- at alle pakker indeholder MAC adressen
- WEP nøglen skifter sjældent
- ca. 2/3 af de netværk man finder har ikke WEP slået til - og der er fri og uhindret adgang til Internet

Man kan altså lytte med på et netværk med WEP, genbruge en anden maskines MAC adresse - og måske endda bryde WEP krypteringen.

Medmindre man kender virksomheden og WEP nøglen ikke er skiftet ... det er besværligt at skifte den, idet alle stationer skal opdateres.



© 2001 Microsoft Corp. All rights reserved.

# Informationsindsamling

Det vi har udført er informationsindsamling

Indsamlingen kan være aktiv eller passiv indsamling i forhold til målet for angrebet  
passiv kunne være at lytte med på trafik eller søge i databaser på Internet  
aktiv indsamling er eksempelvis at sende ICMP pakker og registrere hvad man får af  
svar

# WEP kryptering

WEP *kryptering* - med nøgler der specificeres som tekst eller hexadecimale cifre typisk 40-bit, svarende til 5 ASCII tegn eller 10 hexadecimale cifre eller 104-bit 13 ASCII tegn eller 26 hexadecimale cifre

WEP er baseret på RC4 algoritmen der er en *stream cipher* lavet af Ron Rivest for RSA Data Security

## De første fejl ved WEP

Oprindeligt en dårlig implementation i mange Access Points

Fejl i krypteringen - rettet i nyere firmware

WEP er baseret på en DELT hemmelighed som alle stationer kender

Nøglen ændres sjældent, og det er svært at distribuere en ny



WEP er *ok* til et privat hjemmenetværk

WEP er for simpel til et større netværk - eksempelvis 20 brugere

Firmaer bør efter min mening bruge andre sikkerhedsforanstaltninger

Hvordan udelukker man en bestemt bruger?

# WEP sikkerhed



AirSnort Homepage



AirSnort is a wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered.

802.11b, using the Wired Equivalent Protocol (WEP), is crippled with numerous security flaws. Most damning of these is the weakness described in "Weaknesses in the Key Scheduling Algorithm of RC4" by Scott Fluhrer, Itsik Mantin and Adi Shamir. Adam Stubblefield was the first to implement this attack, but he has not made his software public. AirSnort, along with WEPCrack, which was released about the same time as AirSnort, are the first publicly available implementations of this attack. <http://airsnort.shmoo.com/>

## major cryptographic errors

weak keying - 24 bit er allerede kendt - 128-bit = 104 bit i praksis

small IV - med kun 24 bit vil hver IV blive genbrugt oftere

CRC-32 som intergritetscheck er ikke *stærkt* nok kryptografisk set

Authentication gives pad - giver fuld adgang - hvis der bare opdages *encryption pad* for en bestemt IV. Denne IV kan så bruges til al fremtidig kommunikation

**Konklusion: Kryptografi er svært**

# WEP cracking - airodump og aircrack



airodump - opsamling af krypterede pakker

aircrack - statistisk analyse og forsøg på at finde WEP nøglen

Med disse værktøjer er det muligt at knække *128-bit nøgler!*

Blandt andet fordi det reelt er 104-bit nøgler ☺

tommelfingerregel - der skal opsamles mange pakker ca. 500.000 er godt

Links:

<http://www.cr0.net:8040/code/network/aircrack/> aircrack

<http://www.securityfocus.com/infocus/1814> WEP: Dead Again

# airodump afvikling



Når airodump kører opsamles pakkerne  
samtidig vises antal initialisationsvektorer IV's:

BSSID	CH	MB	ENC	PWR	Packets	LAN IP / # IVs	ESSID
00:03:93:ED:DD:8D	6	11		209	<b>801963</b>	<b>540180</b>	wanlan

NB: dataopsamlingen er foretaget på 100% opdateret Mac udstyr

# aircrack - WEP cracker



```
$ aircrack -n 128 -f 2 aftendump-128.cap
                           aircrack 2.1
* Got 540196! unique IVs | fudge factor = 2
* Elapsed time [00:00:22] | tried 12 keys at 32 k/m
KB      depth      votes
 0      0/   1      CE( 45) A1( 20) 7E( 15) 98( 15) 72( 12) 82( 12)
 1      0/   2      62( 43) 1D( 24) 29( 15) 67( 13) 94( 13) F7( 13)
 2      0/   1      B6( 499) E7( 18) 8F( 15) 14( 13) 1D( 12) E5( 10)
 3      0/   1      4E( 157) EE( 40) 29( 39) 15( 30) 7D( 28) 61( 20)
 4      0/   1      93( 136) B1( 28) 0C( 15) 28( 15) 76( 15) D6( 15)
 5      0/   2      E1( 75) CC( 45) 39( 31) 3B( 30) 4F( 16) 49( 13)
 6      0/   2      3B( 65) 51( 42) 2D( 24) 14( 21) 5E( 15) FC( 15)
 7      0/   2      6A( 144) 0C( 96) CF( 34) 14( 33) 16( 33) 18( 27)
 8      0/   1      3A( 152) 73( 41) 97( 35) 57( 28) 5A( 27) 9D( 27)
 9      0/   1      F1( 93) 2D( 45) 51( 29) 57( 27) 59( 27) 16( 26)
10     2/   3      5B( 40) 53( 30) 59( 24) 2D( 15) 67( 15) 71( 12)
11     0/   2      F5( 53) C6( 51) F0( 21) FB( 21) 17( 15) 77( 15)
12     0/   2      E6( 88) F7( 81) D3( 36) E2( 32) E1( 29) D8( 27)
```

**KEY FOUND! [ CE62B64E93E13B6A3AF15BF5E6 ]**

# Hvor lang tid tager det?

Opsamling a data - ca. en halv time på 802.11b ved optimale forhold

Tiden for kørsel af aircrack fra auditor CD på en Dell CPi 366MHz Pentium II laptop:

```
$ time aircrack -n 128 -f 2 aftendump-128.cap
...
real      5m44.180s    user     0m5.902s    sys     1m42.745s
```

## Hvor lang tid tager det?

Opsamling a data - ca. en halv time på 802.11b ved optimale forhold

Tiden for kørsel af aircrack fra auditor CD på en Dell CPi 366MHz Pentium II laptop:

```
$ time aircrack -n 128 -f 2 aftendump-128.cap
...
real      5m44.180s    user     0m5.902s    sys     1m42.745s
```

Tiden for kørsel af aircrack på en moderne 1.6GHz CPU med almindelig laptop disk tager typisk mindre end 60 sekunder

Det anbefales at bruge:

Kendte VPN teknologier eller WPA

baseret på troværdige algoritmer

implementeret i professionelt udstyr

fra troværdige leverandører

udstyr der vedligeholdes og opdateres

Man kan måske endda bruge de eksisterende løsninger - fra hjemmepc adgang, mobil  
adgang m.v.

RADIUS er en protokol til autentificering af brugere op mod en fælles server

Remote Authentication Dial In User Service (RADIUS)

RADIUS er beskrevet i RFC-2865

RADIUS kan være en fordel i større netværk med

- dial-in
- administration af netværksudstyr
- trådløse netværk
- andre RADIUS kompatible applikationer

# Erstatninger for WEP

Der findes idag andre metoder til sikring af trådløse netværk

802.1x Port Based Network Access Control

WPA - Wi-Fi Protected Access)

WPA = 802.1X + EAP + TKIP + MIC

nu WPA2

WPA2 is based on the final IEEE 802.11i amendment to the 802.11 standard and is eligible for FIPS 140-2 compliance.

Kilde: [http://www.wifialliance.org/OpenSection/protected\\_access.asp](http://www.wifialliance.org/OpenSection/protected_access.asp)

## WPA eller WPA2?

WPA2 is based upon the Institute for Electrical and Electronics Engineers (IEEE) 802.11i amendment to the 802.11 standard, which was ratified on July 29, 2004.

Q: How are WPA and WPA2 similar?

A: Both WPA and WPA2 offer a high level of assurance for end-users and network administrators that their data will remain private and access to their network restricted to authorized users. Both utilize 802.1X and Extensible Authentication Protocol (EAP) for authentication. Both have Personal and Enterprise modes of operation that meet the distinct needs of the two different consumer and enterprise market segments.

Q: How are WPA and WPA2 different?

A: WPA2 provides a **stronger encryption mechanism** through **Advanced Encryption Standard (AES)**, which is a requirement for some corporate and government users.

Kilde: <http://www.wifialliance.org> WPA2 Q and A

# WPA Personal eller Enterprise

Personal - en delt hemmelighed, preshared key

Enterprise - brugere valideres op mod fælles server

Hvorfor er det bedre?

- Flere valgmuligheder - passer til store og små
  - WPA skifter den faktiske krypteringsnøgle jævnligt - TKIP
  - Initialisationsvektoren (IV) fordobles 24 til 48 bit
  - Imødekommer alle kendte problemer med WEP!
  - Integrerer godt med andre teknologier - RADIUS
- 
- EAP - Extensible Authentication Protocol - individuel autentifikation
  - TKIP - Temporal Key Integrity Protocol - nøgleskift og integritet
  - MIC - Message Integrity Code - Michael, ny algoritme til integritet

Nu skifter vi så til WPA og alt er vel så godt?

# WPA cracking

Nu skifter vi så til WPA og alt er vel så godt?

Desværre ikke!

Du skal vælge en laaaaaang passphrase, ellers kan man sniffe WPA handshake når en computer går ind på netværket!

Med et handshake kan man med aircrack igen lave off-line bruteforce angreb!

# WPA cracking demo

Vi konfigurerer AP med Henrik42 som WPA-PSK/passphrase

Vi finder netværk kismet eller airodump

Vi starter airodump mod specifik kanal

Vi spoofer deauth og opsamler WPA handshake

Vi knækker WPA :-)

Brug manualsiderne for programmerne i aircrack-ng pakken!

# WPA cracking med aircrack - start

```
slax ~ # aircrack-ng -w dict wlan-test.cap
Opening wlan-test.cap
Read 1082 packets.
```

#	BSSID	ESSID	Encryption
1	00:11:24:0C:DF:97	wlan	WPA (1 handshake)
2	00:13:5F:26:68:D0	Noea	No data - WEP or WPA
3	00:13:5F:26:64:80	Noea	No data - WEP or WPA
4	00:00:00:00:00:00		Unknown

Index number of target network ? **1**

# WPA cracking med aircrack - start

```
[00:00:00] 0 keys tested (0.00 k/s)
```

```
KEY FOUND! [ Henrik42 ]
```

```
Master Key      : 8E 61 AB A2 C5 25 4D 3F 4B 33 E6 AD 2D 55 6F 76  
                  6E 88 AC DA EF A3 DE 30 AF D8 99 DB F5 8F 4D BD
```

```
Transient Key   : C5 BB 27 DE EA 34 8F E4 81 E7 AA 52 C7 B4 F4 56  
                  F2 FC 30 B4 66 99 26 35 08 52 98 26 AE 49 5E D7  
                  9F 28 98 AF 02 CA 29 8A 53 11 EB 24 0C B0 1A 0D  
                  64 75 72 BF 8D AA 17 8B 9D 94 A9 31 DC FB 0C ED
```

```
EAPOL HMAC     : 27 4E 6D 90 55 8F 0C EB E1 AE C8 93 E6 AC A5 1F
```

Min Thinkpad X31 med 1.6GHz Pentium M knækker ca. 150 Keys/sekund

# Tools man bør kende

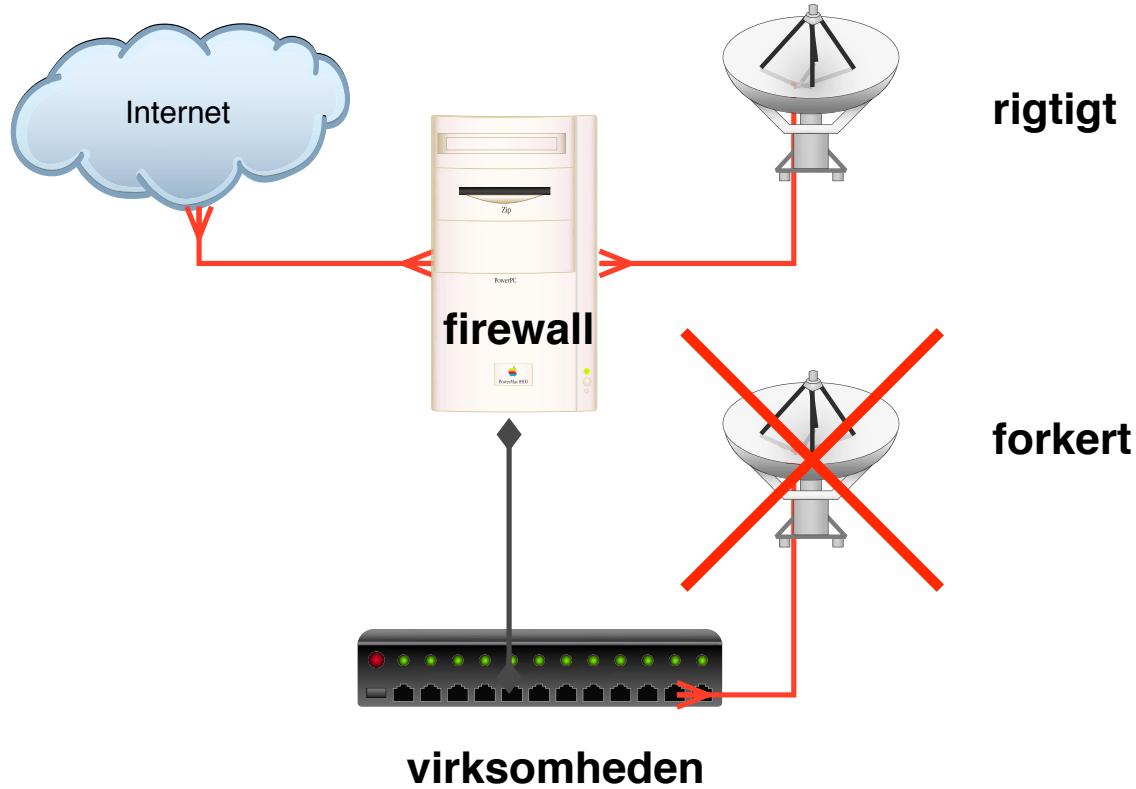


- **BSD Airtools** <http://www.dachb0den.com/projects/bsd-airtools.html>
- **Kismet** <http://www.kismetwireless.net/>
- **Airsnort** <http://airsnort.shmoo.com/> læs pakkerne med WEP kryptering
- **wepcrack** <http://wepcrack.sourceforge.net/> - knæk krypteringen i WEP
- **Airsnarf** <http://airsnarf.shmoo.com/> - lav dit eget AP parallelt med det rigtige og snif hemmeligheder
- **Wireless Scanner** <http://www.iss.net/> - kommersielt
- **Dette er et lille uddrag af programmer**  
Se også <http://packetstormsecurity.org/wireless/>

Så går man igang med de almindelige værktøjer

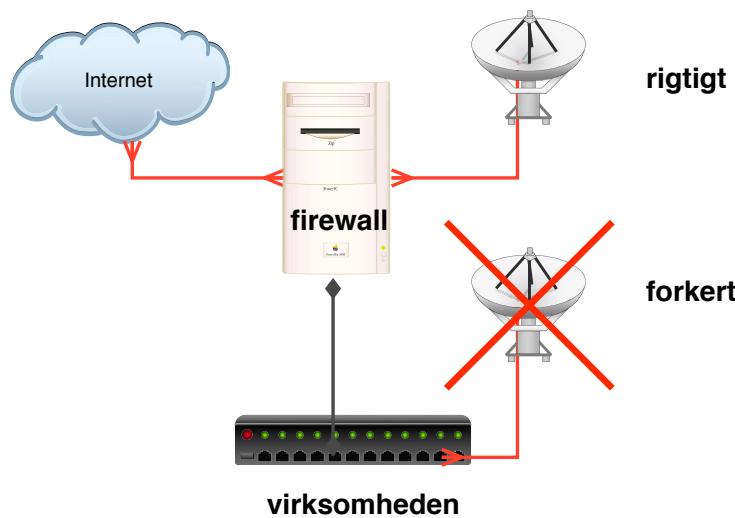
Fyodor Top 100 Network Security Tools <http://www.sectools.org>

Forsvaret er som altid - flere lag af sikkerhed!



Sådan bør et access point forbindes til netværket

# Anbefalinger mht. trådløse netværk



- Brug noget tilfældigt som SSID - netnavnet
- Brug ikke WEP til virksomhedens netværk  
- men istedet en VPN løsning med individuel autentificering eller WPA
- NB: WPA Personal/PSK kræver passphrase på +40 tegn!
- Placer de trådløse adgangspunkter hensigtsmæssigt i netværket - så de kan overvåges
- Lav et sæt regler for brugen af trådløse netværk - hvor må medarbejdere bruge det?
- Se eventuelt pjecerne *Beskyt dit trådløse Netværk* fra Ministeriet for Videnskab, Teknologi og Udvikling  
<http://www.videnskabsministeriet.dk/>

Lad være med at bruge et wireless-kort i en PC til at lave AP, brug et AP

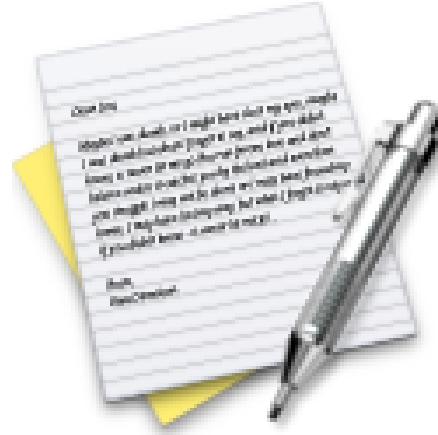
Husk et AP kan være en router, men den kan ofte også blot være en bro

Brug WPA og overvej at lave en decideret DMZ til WLAN

Placer AP hensigtsmæddigt og gerne højt, oppe på et skab eller lignende



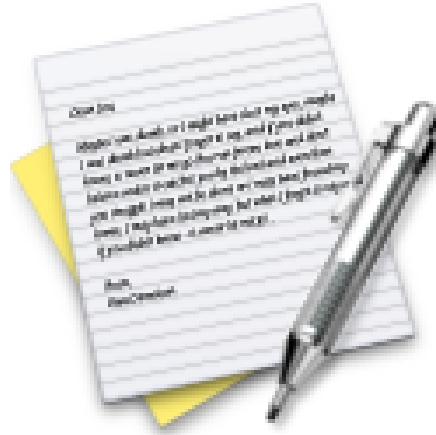
Vi laver nu øvelsen  
**AirPort Extreme**  
som er øvelse **23** fra øvelseshæftet.



Vi laver nu øvelsen

## Wardriving på Windows - netstumbler

som er øvelse **24** fra øvelseshæftet.



Vi laver nu øvelsen

## Wardriving på UNIX - Kismet

som er øvelse **25** fra øvelseshæftet.

## Dynamisk routing

Når netværkene vokser bliver det administrativt svært at vedligeholde

Det skalerer dårligt med statiske routes til netværk

Samtidig vil man gerne have redundante forbindelser

Til dette brug har man STP på switch niveau og dynamisk routing på IP niveau

# BGP Border Gateway Protocol

Er en dynamisk routing protocol som benyttes eksternt

Netværk defineret med AS numre annoncerer hvilke netværk de er forbundet til

Autonomous System (AS) er en samling netværk

BGP version 4 er beskrevet i RFC-4271

BGP routere forbinder sig til andre BGP routere og snakker sammen, *peering*

[http://en.wikipedia.org/wiki/Border\\_Gateway\\_Protocol](http://en.wikipedia.org/wiki/Border_Gateway_Protocol)

Vores setup svarer til dette:

[http://www.kramse.dk/projects/network/openbgpd-basic\\_en.html](http://www.kramse.dk/projects/network/openbgpd-basic_en.html)

# RIP Routing Information Protocol

Gammel routingprotokol som ikke benyttes mere

RIP er en distance vector routing protokol, tæller antal hops

[http://en.wikipedia.org/wiki/Routing\\_Information\\_Protocol](http://en.wikipedia.org/wiki/Routing_Information_Protocol)

# OSPF Open Shortest Path First

Er en dynamisk routing protocol som benyttes til intern routing

OSPF version 3 er beskrevet i RFC-2740

OSPF bruger hverken TCP eller UDP, men sin egen protocol med ID 89

OSPF bruger en metric/cost pr link for at udregne smart routing

[http://en.wikipedia.org/wiki/Open\\_Shortest\\_Path\\_First](http://en.wikipedia.org/wiki/Open_Shortest_Path_First)

Vores setup svarer til OpenBGPD setup, blot med OpenSPFD

Cisco protokol til intern routing, hvis man udelukkende har Cisco udstyr

<http://www.cisco.com>

## Stop - vi gennemgår og tester vores dynamiske routing

Vi gennemgår hvordan vores setup ser ud

Vi laver traceroute før og efter:

Vi fjerner en ledning *link down*

Vi stopper en router og ser de annoncerede netværk forsvinder

Vi bootter en router og ser de annoncerede netværk igen

# Båndbredestyring og policy based routing

Mange routere og firewalls idag kan lave båndbredde allokering til protokoller, porte og derved bestemte services

Mest kendte er i Open Source:

- ALTQ bruges på OpenBSD - integreret i PF
- FreeBSD har dummynet
- Linux har tilsvarende  
ADSL-Bandwidth-Management-HOWTO, ADSL Bandwidth Management HOWTO  
Adv-Routing-HOWTO, Linux Advanced Routing & Traffic Control HOWTO  
<http://www.knowplace.org/shaper/resources.html> Linux resources

Det kaldes også traffic shaping

# Routingproblemer, angreb

falske routing updates til protokollerne

sende redirect til maskiner

source routing - mulighed for at specificere en ønsket vej for pakken

Der findes (igen) specialiserede programmer til at teste og forfalske routing updates, svarende til icmpush programmet

Det anbefales at sikre routere bedst muligt - eksempelvis Secure IOS template der findes på adressen:

<http://www.cymru.com/Documents/secure-ios-template.html>

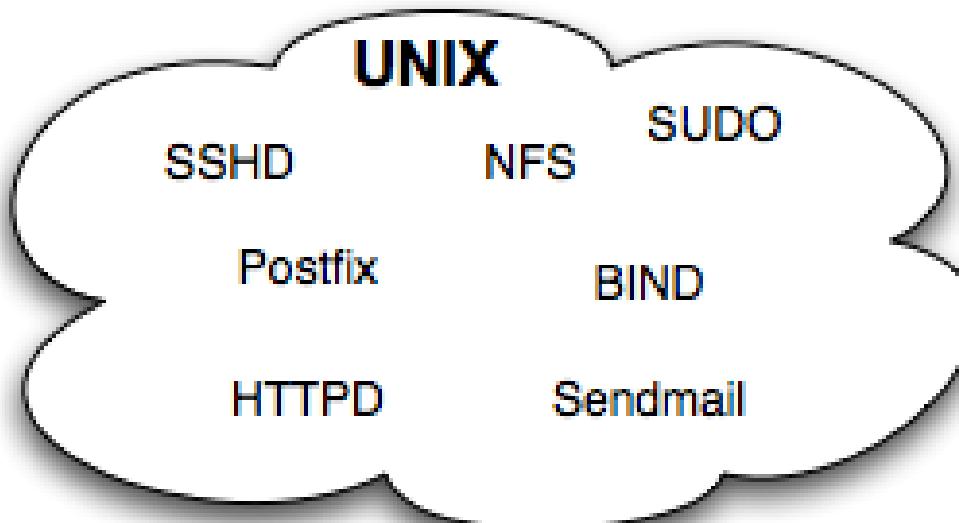
Med UNIX systemer generelt anbefales opdaterede systemer og netværkstuning

## Source routing

Hvis en angriber kan fortælle hvilken vej en pakke skal følge kan det give anledning til sikkerhedsproblemer

maskiner idag bør ikke lytte til source routing, evt. skal de droppe pakkerne

# Formålet med resten af dagen



Vi skal gennemgå gængse internet-serverfunktioner

# Network Services

Flere UNIX varianter har fået mere moderne strukturer til at starte services

SystemV start/stop af services er stadig meget udbredt rc.d katalogstrukturer

Solaris: Service Management Facility SMF

AIX: Subsystem Ressource Controller

Mac OS X: launchd

Windows: services, net stop/start m.fl.

Hjælper med til at køre systemet

udfører jobs

typiske daemoner er:

- ftpd - FTP daemonen giver FTP adgang til filoverførsler
- Telnetd - giver login adgang - NB: ukrypteret!
- tftpd - Trivial file transfer protocol daemon, bruges til boot og opgradering af netværksudstyr - kræver ikke password
- pop3d - POP3 post office protocol, elektronisk post
- sshd - SSH protokol daemonen giver adgang til login via SSH

# inetd en super server

inetd har mange funktioner

istedet for at have 10 programmer der lytter på diverse porte kan inetd lytte på en hel masse, og så give forbindelsen videre til programmerne når der er brug for det:

/etc/inetd.conf

finger	stream	tcp	nowait	nobody	/usr/libexec/tcpd	fingerd -s
ftp	stream	tcp	nowait	root	/usr/libexec/tcpd	ftpd -l
login	stream	tcp	nowait	root	/usr/libexec/tcpd	rlogind
nntp	stream	tcp	nowait	usenet	/usr/libexec/tcpd	nntpd
ntalk	dgram	udp	wait	root	/usr/libexec/tcpd	ntalkd
shell	stream	tcp	nowait	root	/usr/libexec/tcpd	rshd
telnet	stream	tcp	nowait	root	/usr/libexec/tcpd	telnetd
uucpd	stream	tcp	nowait	root	/usr/libexec/tcpd	uucpd
comsat	dgram	udp	wait	root	/usr/libexec/tcpd	comsat
tftp	dgram	udp	wait	nobody	/usr/libexec/tcpd	tftpd /tftpboot

konfigureres med separate filer pr service i kataloget /etc/xinetd.d eksempelvis:  
/etc/xinetd.d/cups-lpd:

```
service printer
{
    socket_type  = stream
    protocol     = tcp
    wait         = no
    user         = lp
    server       = /usr/lib/cups/daemon/cups-lpd
    disable      = yes
}
```



De fleste benytter idag standard kommandoerne:

- `lp` og `lpr` - print files
- `lpq` - show printer queue status
- `lprm` - cancel print jobs
- Mange bruger softwaren Common UNIX Printing System fra <http://www.cups.org>
- Gamle UNIX systemer bruger stadig konfiguration via `/etc/printcap`
- remote print sker gennem Line Printer Daemon LPD protokollen port 515/tcp
- nyere printere understøtter Internet Printing Protocol IPP port 80/tcp

Kilde: billede er fra CUPS

Trivial File Transfer Protocol - uautentificerede filoverførsler

De bruges især til:

- TFTP bruges til boot af netværksklienter uden egen harddisk
- TFTP benytter UDP og er derfor ikke garanteret at data overføres korrekt

TFTP sender alt i klartekst, hverken password

**USER** brugernavn og

**PASS** hemmeligt-kodeord

# FTP File Transfer Protocol

File Transfer Protocol - filoverførsler

Bruges især til:

- FTP - drivere, dokumenter, rettelser - Windows Update? er enten HTTP eller FTP

FTP sender i klartekst

**USER brugernavn** og

**PASS hemmeligt-kodeord**

Der findes varianter som tillader kryptering, men brug istedet SCP/SFTP over Secure Shell protokollen

# FTP Daemon konfiguration

Meget forskelligt!

WU-FTPD er meget udbredt

BSD FTPD ligeså meget anvendt

*anonym ftp* er når man tillader alle at logge ind  
men husk så ikke at tillade upload af filer!

På BSD oprettes blot en bruger med navnet `ftp` så er der åbent!

## NTP opsætning

foregår typisk i /etc/ntp.conf eller /etc/ntpd.conf

det vigtigste er navnet på den server man vil bruge som tidskilde

Brug enten en NTP server hos din udbyder eller en fra <http://www.pool.ntp.org/>

Eksempelvis:

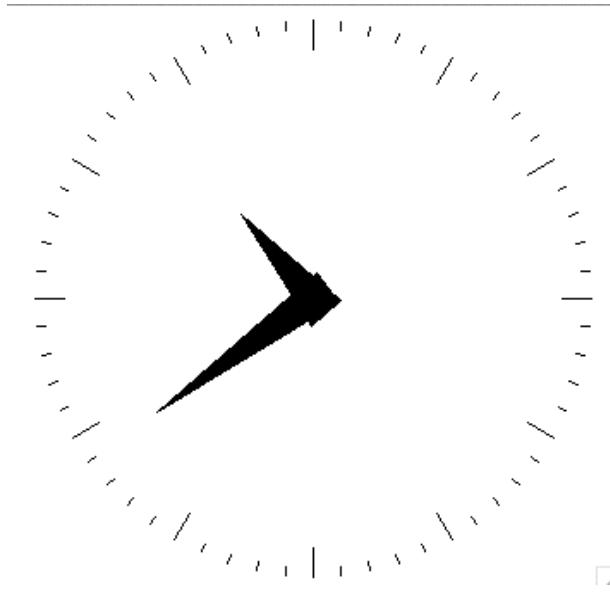
```
server ntp.cybercity.dk
```

```
server 0.dk.pool.ntp.org
```

```
server 0.europe.pool.ntp.org
```

```
server 3.europe.pool.ntp.org
```

# What time is it?



Hvad er klokken?

Hvad betydning har det for sikkerheden?

Brug NTP Network Time Protocol på produktionssystemer

# What time is it? - spørg ICMP

ICMP timestamp option - request/reply

hvad er klokken på en server

Slayer icmpush - er installeret på server

viser tidstempel

```
# icmpush -v -tstamp 10.0.0.12
```

```
ICMP Timestamp Request packet sent to 10.0.0.12 (10.0.0.12)
```

```
Receiving ICMP replies ...
```

```
fischer          -> 21:27:17
```

```
icmpush: Program finished OK
```

# Stop - NTP Konfigurationseksempler



Vi har en masse udstyr, de meste kan NTP, men hvordan

Vi gennemgår, eller I undersøger selv:

- Airport
- Switche (managed)
- Mac OS X
- OpenBSD - check `man rdate` og `man ntpd`

## BIND DNS server

Berkeley Internet Name Daemon server

Mange bruger BIND fra Internet Systems Consortium - altså Open Source

konfigureres gennem named.conf

det anbefales at bruge BIND version 9

- *DNS and BIND*, Paul Albitz & Cricket Liu, O'Reilly, 4th edition Maj 2001
- *DNS and BIND cookbook*, Cricket Liu, O'Reilly, 4th edition Oktober 2002

Kilde: <http://www.isc.org>

# BIND konfiguration - et udgangspunkt

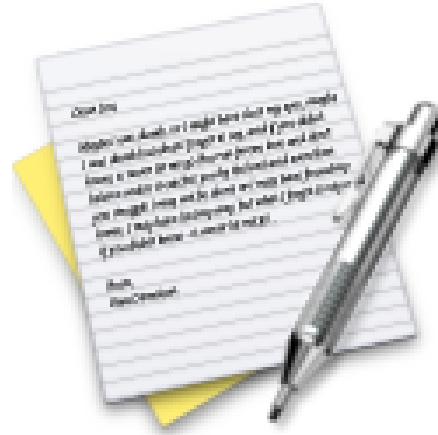
```
acl internals { 127.0.0.1; ::1; 10.0.0.0/24; };
options {
    // the random device depends on the OS !
    random-device "/dev/random"; directory "/namedb";
    port 53; version "Dont know"; allow-query { any; };
};
view "internal" {
    match-clients { internals; };
    recursion yes;
    zone "." {
        type hint; file "root.cache"; };
    // localhost forward lookup
    zone "localhost." {
        type master; file "internal/db.localhost";   };
    // localhost reverse lookup from IPv4 address
    zone "0.0.127.in-addr.arpa" {
        type master; file "internal/db.127.0.0"; notify no;   };
    ...
}
```



Vi laver nu øvelsen

## BIND version

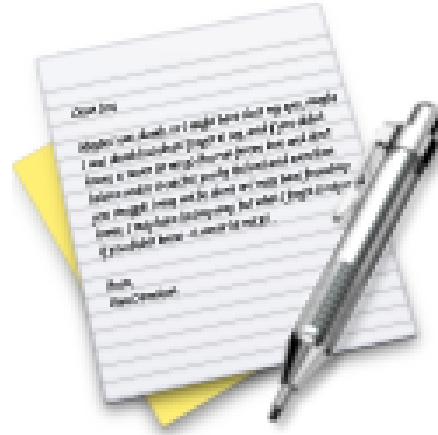
som er øvelse **26** fra øvelseshæftet.



Vi laver nu øvelsen

## Tilpasning af DNS server

som er øvelse **27** fra øvelseshæftet.



Vi laver nu øvelsen

## Vedligehold af DNS systemer

som er øvelse **28** fra øvelseshæftet.

# Små DNS tools bind-version - Shell script

```
#!/bin/sh
# Try to get version info from BIND server
PROGRAM=`basename $0`
. `dirname $0`/functions.sh
if [ $# -ne 1 ]; then
    echo "get name server version, need a target! "
    echo "Usage: $0 target"
    echo "example $0 10.1.2.3"
    exit 0
fi
TARGET=$1
# using dig
start_time
dig @$1 version.bind chaos txt
echo Authors BIND er i versionerne 9.1 og 9.2 - måske ...
dig @$1 authors.bind chaos txt
stop_time
http://www.kramse.dk/files/tools/dns/bind-version
```

# Små DNS tools dns-timecheck - Perl script



```
#!/usr/bin/perl
# modified from original by Henrik Kramshøj, hlk@kramse.dk
# 2004-08-19
#
# Original from: http://www.rfc.se/fpdns/timecheck.html
use Net::DNS;

my $resolver = Net::DNS::Resolver->new;
$resolver->nameservers ($ARGV[0] );

my $query = Net::DNS::Packet->new;
$query->sign_tsig("n","test");

my $response = $resolver->send($query);
foreach my $rr ($response->additional)
  print "localtime vs nameserver $ARGV[0] time difference: ";
  print $rr->time_signed - time() if $rr->type eq "TSIG";
```

<http://www.kramse.dk/files/tools/dns/dns-timecheck>

# DHCPD server

Dynamic Host Configuration Protocol Server

Mange bruger DHCPD fra Internet Systems Consortium

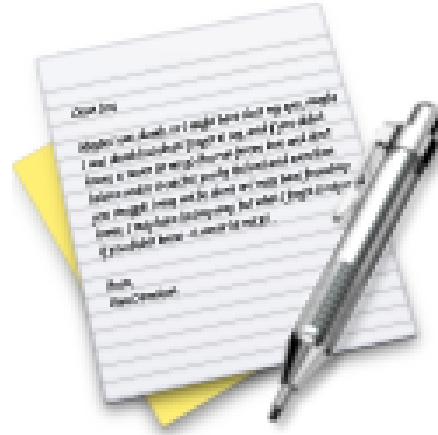
<http://www.isc.org> - altså Open Source

konfigureres gennem `dhcpd.conf` - næsten samme syntaks som BIND

DHCP er en efterfølger til BOOTP protokollen

```
ddns-update-style ad-hoc;
```

```
shared-network LOCAL-NET {
    option domain-name "security6.net";
    option domain-name-servers 212.242.40.3, 212.242.40.51;
    subnet 10.0.42.0 netmask 255.255.255.0 {
        option routers 10.0.42.1;
        range 10.0.42.32 10.0.42.127;
    }
}
```



Vi laver nu øvelsen

## Konfiguration af DHCP server

som er øvelse **29** fra øvelseshæftet.

# Logfiler

Logfiler er en nødvendighed for at have et transaktionsspor

Logfiler giver mulighed for statistik

Logfiler er desuden nødvendige for at fejlfinde

Det kan være relevant at sammenholde logfiler fra:

- routere
- firewalls
- webservere
- intrusion detection systemer
- adgangskontrolsystemer
- ...

Husk - tiden er vigtig! Network Time Protocol (NTP) anbefales

Husk at logfilerne typisk kan slettes af en angriber - hvis denne får kontrol med systemet

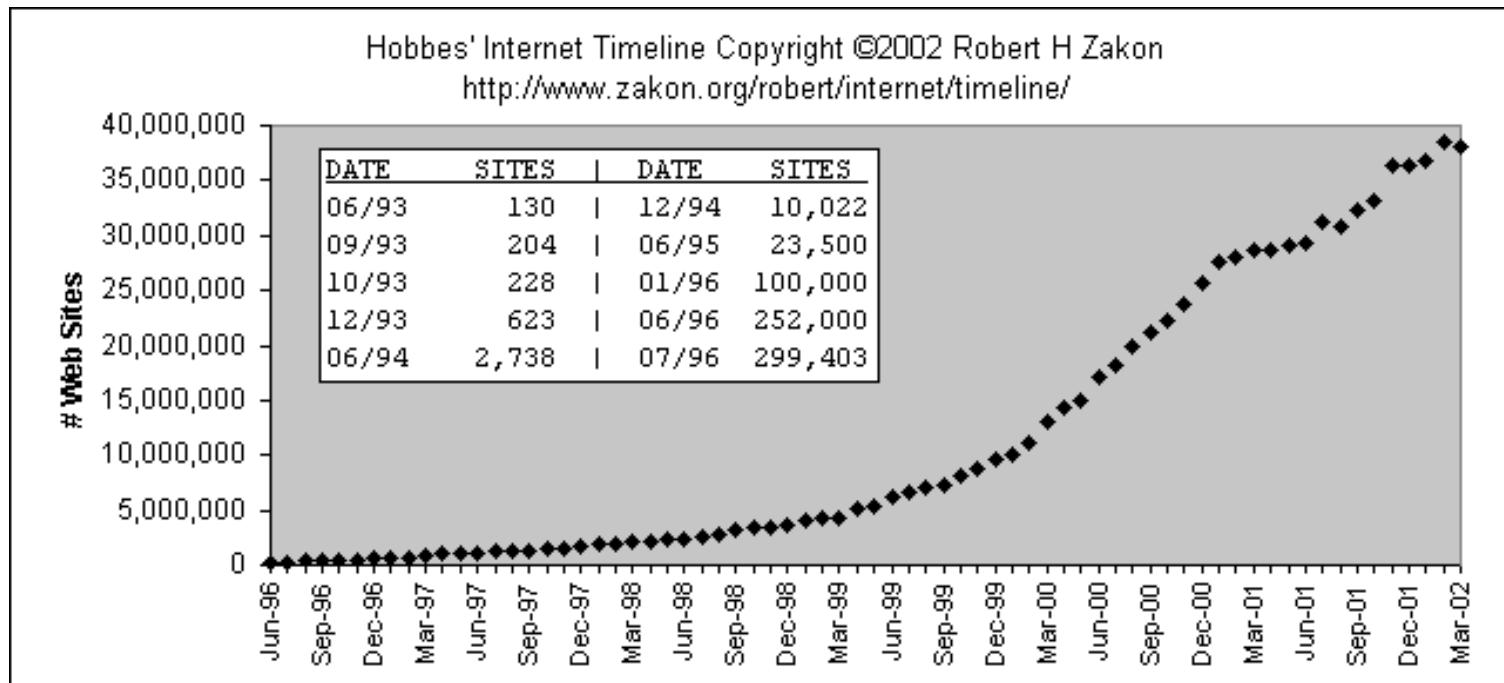
# World Wide Web fødes



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 af individuelle HTTP forespørgsler via TCP forbindelser

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)



## Hvorfor skrive Apache HTTP server?

Fordi Apache idag er en organisation med mange delprojekter - hvoraf mange relateres til web og webløsninger

# Er Apache HTTP server interessant?

Top Developers					
Developer	December 2004	Percent	January 2005	Percent	Change
Apache	38614673	67.84	39821368	68.43	0.59
Microsoft	12062761	21.19	12137446	20.86	-0.33
Sun	1812966	3.18	1830008	3.14	-0.04
Zeus	687508	1.21	690193	1.19	-0.02

Apache HTTP server er iflg. netcraft og andre den mest benyttede HTTP server på Internet!

Apache er grand old man i Internet sammenhæng - bygget udfra NCSA HTTP serveren

Mange løsninger bygges på Apache

Kilde: <http://www.netcraft.com>

# Hvad er Apache?



The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Kilde: Apache HTTPD FAQ <http://httpd.apache.org>

# Fordele ved Apache HTTPD

En HTTP webserver oprindeligt baseret på NCSA webserveren, (National Center for Supercomputing Applications)

- Apache is "A PAtCHy server"

- Konfigurerbar og fleksibel

- Understøtter moduler

- Open Source og kildeteksten er tilgængelig med en fri licens

- Understøtter HTTP/1.1

- allestedsnærværende - UNIX: Linux, IBM AIX, BSD, Sun Solaris...

Kilde: Apache HTTPD FAQ <http://httpd.apache.org>

## Men Apache er også ...

Apache Software Foundation med mange andre spændende projekter

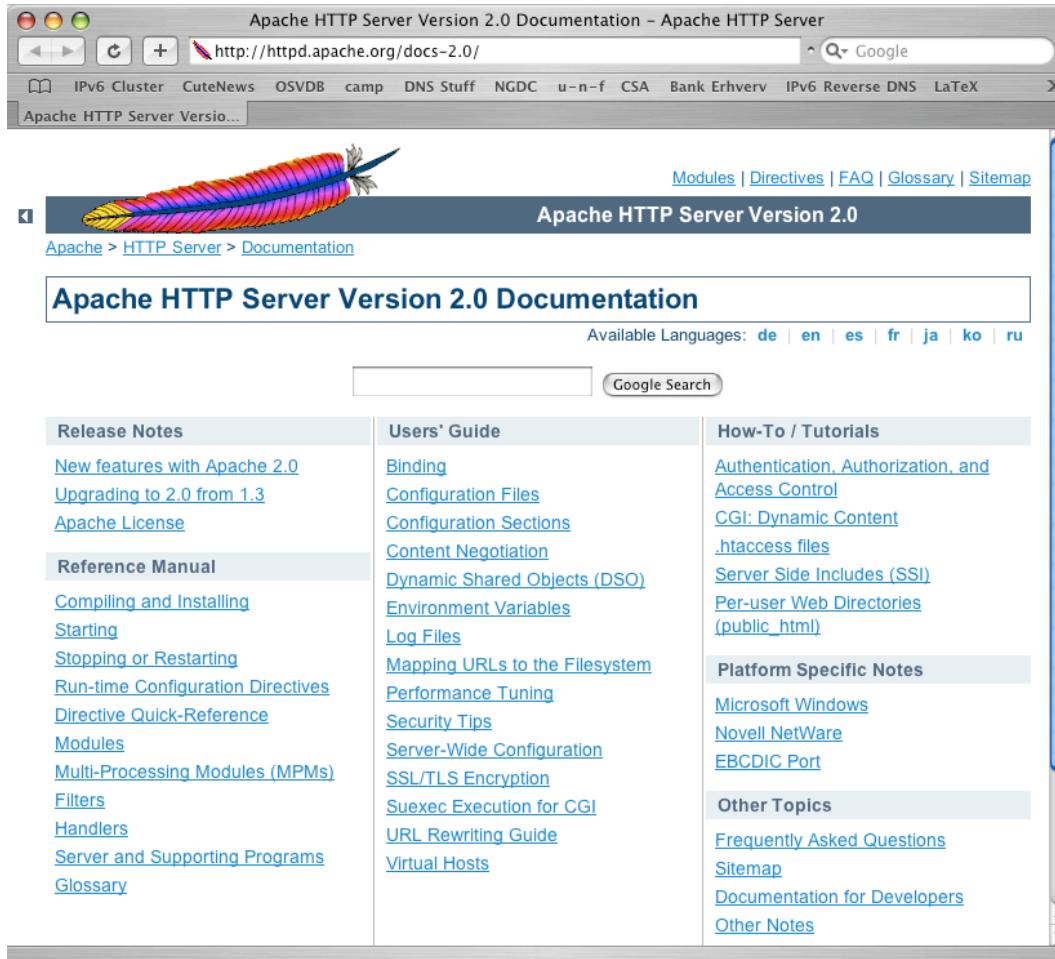
- Cocoon som er et komponentbaseret *web development framework*
- Apache Tomcat som er en servlet container der bruges som den officielle referenceimplementation for Java Servlet og JavaServer Pages teknologierne
- Apache-SSL SSL delen af webserveren
- FOP print formatter drevet af XSL formatting objects (XSL-FO)
- Xerces XML parser
- Xalan XSLT processor

XML og Web services er buzz-words idag

# Apache varianter

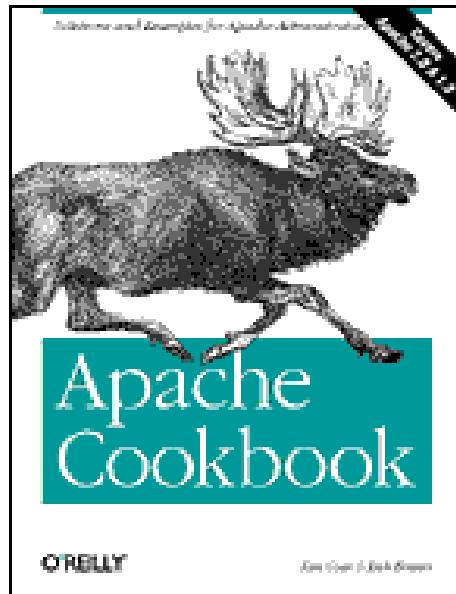
- Stronghold - sælges ikke mere  
<http://www.redhat.com/software/stronghold/>
- IBM HTTP server  
<http://www-306.ibm.com/software/webservers/httpservers/>
- Oracle HTTP Server
- HP Secure Web Server for OpenVMS Alpha (based on Apache)
- findes sikkert flere

# Brug dokumentationen

A screenshot of a web browser displaying the Apache HTTP Server Version 2.0 Documentation. The URL in the address bar is <http://httpd.apache.org/docs-2.0/>. The page title is "Apache HTTP Server Version 2.0 Documentation". The main content area is titled "Apache HTTP Server Version 2.0 Documentation". It includes a search bar and links to various documentation sections like Release Notes, Users' Guide, and How-To / Tutorials.

The screenshot shows the Apache HTTP Server Version 2.0 Documentation page. The URL in the address bar is <http://httpd.apache.org/docs-2.0/>. The page title is "Apache HTTP Server Version 2.0 Documentation". The main content area is titled "Apache HTTP Server Version 2.0 Documentation". It includes a search bar and links to various documentation sections like Release Notes, Users' Guide, and How-To / Tutorials.

<http://httpd.apache.org/docs-2.0/>

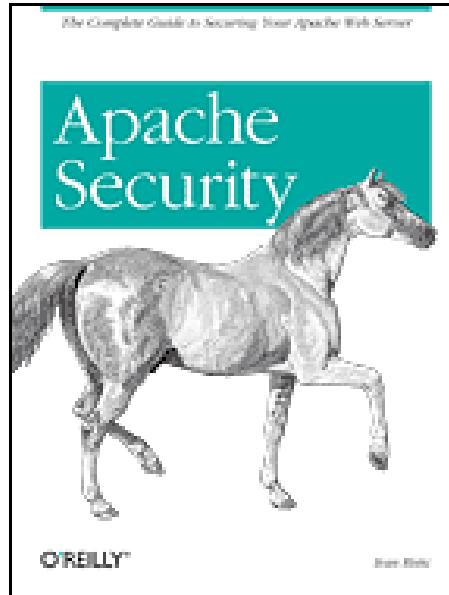


- Vi bruger bogen Apache cookbook på kurset
- Både som opgavehæfte og opslagsværk
- *Apache Cookbook* af Ken Coar, Rich Bowen, November 2003, ISBN: 0-596-00191-6

# Apache Security bogen

Security

.net



- Vi bruger bogen Apache Security bogen på kurset
- Primært som opslagsværk
- *Apache Security* af Ivan Ristic, February 2005, ISBN: 0-596-00724-8

# Start og stop af apache

Apache bruger programmet apachectl

Dette program kan bruges til flere formål:

- apachectl start - opstart af apache
- apachectl stop - stop af apache
- apachectl restart - genstart af apache
- apachectl configtest - test af apache konfigurationen - syntaks!

husk at der kan være flere versioner af apache på systemet!

Det kan være en fordel enten at lave et alias eller ændre PATH i jeres SHELL profil!

```
alias apachectl="/home/hlk/apache2/bin/apachectl"
```

# ServerAdmin, ServerRoot, DocumentRoot

```
ServerAdmin webmaster@security6.net
ServerRoot "/usr/local/apache2"
DocumentRoot "/userdata/sites"
User www
Group www
ServerName fluffy:80
```

- Indsættes i httpd.conf
- ServerAdmin - administratoren for denne webserver, bør sættes til eksempelvis webmaster@security6.net
- ServerRoot - rod'en af serveren, mange andre referencer sker relativt til denne
- DocumentRoot - den primære placering for filer der skal serveres fra denne server
- User og Group - hvilket brugerid skal serveren afvikles under, efter opstarten som root
- Servername - hvad hedder denne server

# Apache Logfiler, konfiguration af logfiler

```
LogLevel warn
ErrorLog /usr/local/apache2/logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\""
           \"%{User-Agent}i\" combined
CustomLog /usr/local/apache2/logs/access_log combined
```

Logning i Apache styres med log-direktiver til to slags - access og error

De vigtigste direktiver i httpd.conf er:

- LogLevel - hvad skal logges af beskeder i error log, fra emerg til debug
- ErrorLog - default fejlbeskeder fra Apache
- LogFormat - hvordan skal access log se ud
- CustomLog - hvor skal access log gemmes - default

# Tuning af Apache opstartsparametre

Det anbefales at indstille Apache opstarten som noget af det første

UNIX bruger som standard prefork modellen hvor Apache starter et antal processer der forventes at være nogenlunde OK til den forventede belastning

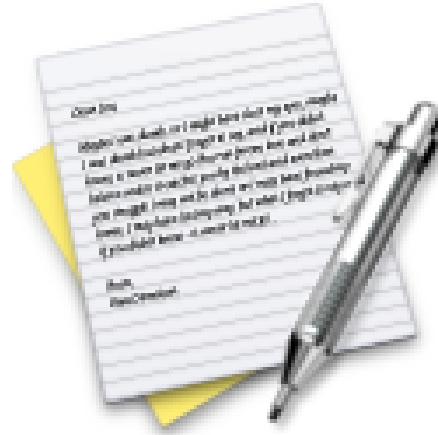
```
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule prefork.c>
StartServers          5
MinSpareServers      5
MaxSpareServers     10
MaxClients          150
MaxRequestsPerChild  0
</IfModule>
```

# Virtuelle hosts

```
<VirtualHost *:80>
    ServerAdmin webmaster@security6.net
    ServerName www.security6.net
    ServerAlias security6.net
    ServerAlias www.security6.dk
    DocumentRoot /userdata/sites/security6.net
    ErrorLog logs/security6.net-error_log
    CustomLog logs/security6.net-access_log combined
...
</VirtualHost>
```

Apache HTTPD tillader at man benytter virtuelle hosts

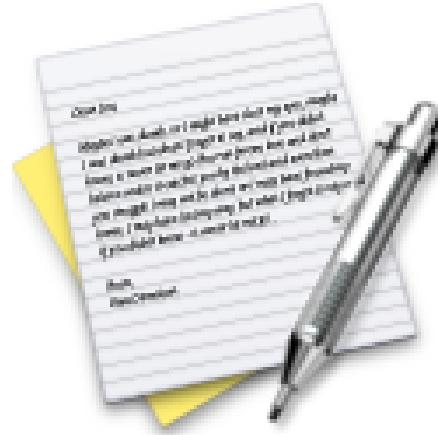
Bemærk at det er klienten der overfører hostnavnet i HTTP request



Vi laver nu øvelsen

## Opbygning af Apache httpd.conf

som er øvelse **30** fra øvelseshæftet.



Vi laver nu øvelsen

## Apache virtuelle hosts

som er øvelse **31** fra øvelseshæftet.

# Grundlæggende Apache CGI

Common Gateway Interface - standard metode til programmer

ScriptAlias er det direktiv der angiver at CGI må afvikles

Der følger to eksempler med Apache 2 i ServerRoot/cgi-bin:

- printenv - viser en del information om serveren
- test-cgi - viser hvordan man kan bruge parametre

**NB: husk at fjerne x-bit efter test!**

```
ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"  
<Directory "/usr/local/apache2/cgi-bin">  
    AllowOverride None  
    Options None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Problem:

Ønsker et simpelt CGI program, en web udgave af finger

Formål:

Vise oplysningerne om brugere på systemet

# review af nogle muligheder

## ASP

- server scripting, meget generelt - man kan alt

## SQL

- databasesprog - meget kraftfuldt
- mange databasesystemer giver mulighed for specifik tildeling af privilegier "grant"

## JAVA

- generelt programmeringssprog
- bytecode verifikation
- indbygget sandbox funktionalitet

## Perl og andre generelle programmeringssprog

Pas på shell escapes!!!

# Hello world of insecure web CGI

Demo af et sårbart system - badfinger

Løsning:

- Kalde finger kommandoen
- et Perl script
- afvikles som CGI
- standard Apache HTTPD 1.3 server

# De vitale - og usikre dele

```
print "Content-type: text/html\n\n<html>";
print "<body bgcolor=#666666 leftmargin=20 topmargin=20";
print "marginwidth=20 marginheight=20>";
print <<XX;
<h1>Bad finger command!</h1>
<HR COLOR=#000>
<form method="post" action="bad_finger.cgi">
Enter userid: <input type="text" size="40" name="command">
</form>
<HR COLOR=#000>
XX
if (&ReadForm(*input)) {
    print "<pre>\n";
    print "will execute:\n/usr/bin/finger $input{'command'}\n";
    print "<HR COLOR=#000>\n";
    print '/usr/bin/finger $input{'command'}';
    print "<pre>\n";
}
}
```

# Almindelige problemer

Security

.net

validering af forms

validering på klient er godt

- godt for brugervenligheden, hurtigt feedback

validering på clientside gør intet for sikkerheden

serverside validering er nødvendigt

generelt er input validering det største problem!

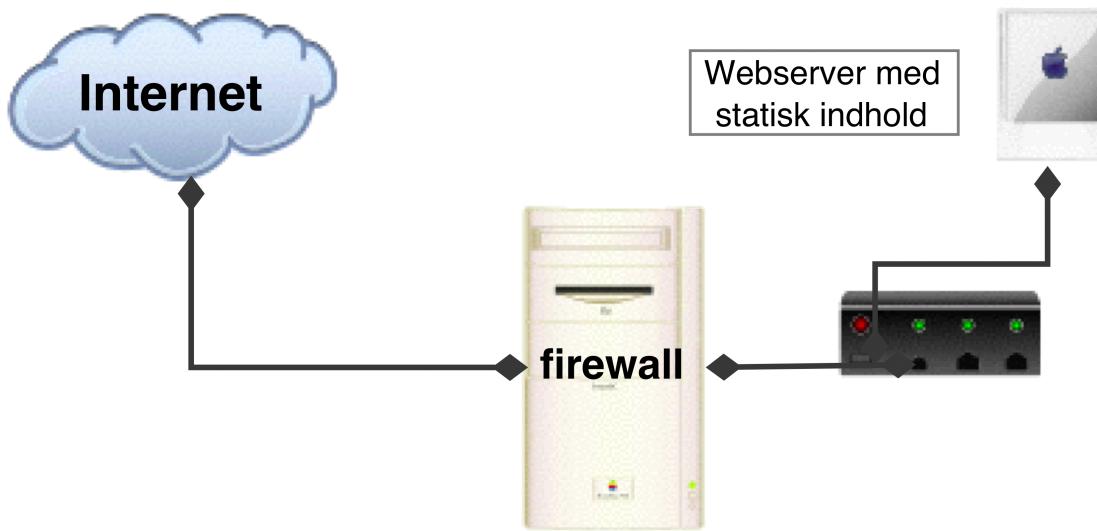
Brug *Open Web Application Security Project* <http://www.owasp.org>

# Apache HTTPD sikkerhedshuller

En Apache installation er ikke bare en HTTPD server  
ofte inkluderes:

- OpenSSL til SSL, dvs HTTPS forbindelser
- PHP - et web programmeringssprog
- Perl - et programmeringssprog som ofte benyttes til web

Hver eneste komponent kan have sikkerhedsproblemer!



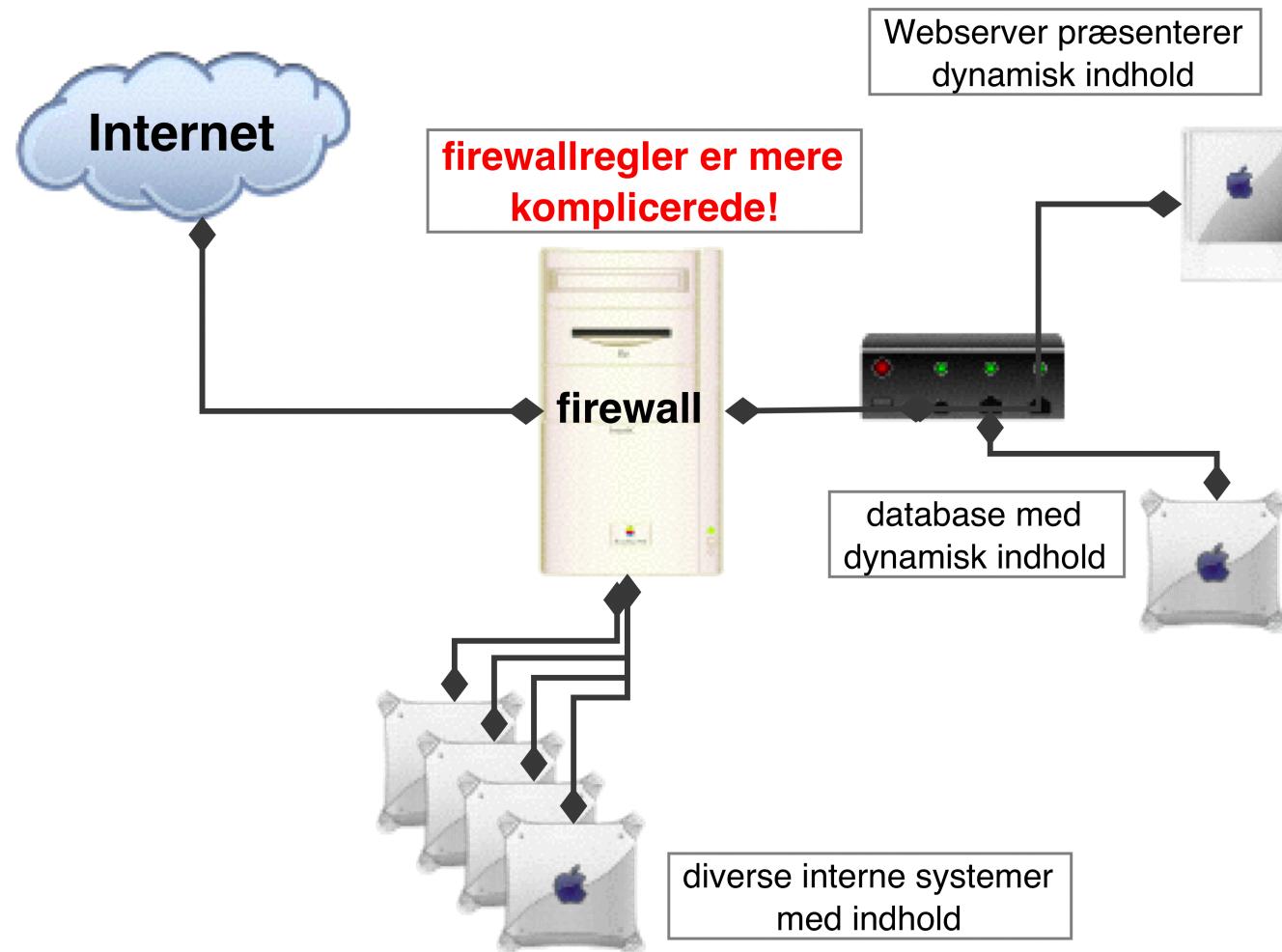
Statiske hjemmesider i HTML

Overskueligt

Få regler i firewall

Ikke behov for adgang til data indenfor firewall

# Web løsninger idag



## Web løsninger idag

Dynamiske hjemmesider - ASP, PHP m.fl.

Høj kompleksitet - flere muligheder for fejl

Mange regler i firewall - flere DMZ områder/net

Behov for adgang til ordredata m.v. indenfor firewall

# Gode Råd til dynamiske webmiljøer



Brug databaser - der er gode muligheder for finmasket adgangskontrol  
Brug versionsstyring - hvem lavede hvilket program, hvornår  
Brug ressourcer på opdatering af medarbejdere  
Lav retningslinier for webudvikling  
Overvåg alle systemerne!

# Typisk fejl på webservere

De mest alvorlige:

- Ingen hærdning
- Ingen opdatering efter idriftsættelse

Medium eller kritiske

- Adgang til eksempel-programmer (eng: sample programs) - kan til tider være meget kritisk!

De mindre alvorlige

- Informationsindsamling
- Netmaske - *icmpush -mask*
- Navne på udviklere, firmaer, datoer

## chroot og jails

Security6.net - [www.security6.net](http://www.security6.net)

Chroot står for change root, og betyder at processen som kalder chroot systemkaldet udskifter sin *filsystemsrod*-/ med et andet katalog på systemet

Oprindeligt blev denne funktion lavet til at teste nye UNIX releases uden at overskrive det oprindelige miljø man havde på systemet

men det kan bruges til at give mere sikkerhed

En daemon eller service der kører chroot'ed er sværere at udnytte - simpelthen fordi den kun har adgang til en lille del af systemet

FreeBSD har en endnu mere avanceret version af chroot som giver endnu mere kontrol over det miljø som programmerne ser

## brug af chroot

Security

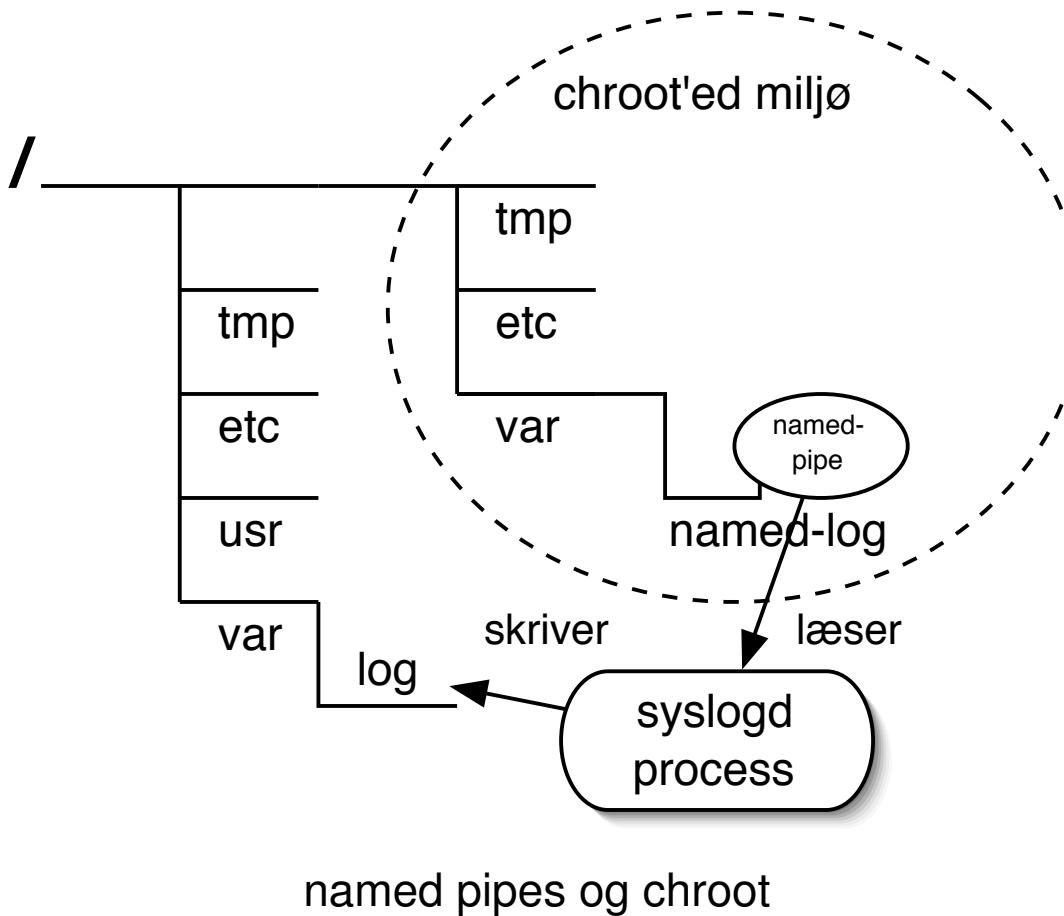
.net

De services man typisk vil chroot'e er BIND, Apache og andre utsatte services

Der findes heldigvis udførlige beskrivelser af hvordan man chroote de mest almindelige services

NB: husk at løsninger med Apache ofte kræver PHP, Perl, databaser osv.

# Gennemgang af chroot konceptet



- Husk: Apachedelen kan være i chroot, mens databasesystem er udenfor - forbindelser via TCP-sockets til localhost

# Produktionsmodning af miljøer

Tænk på det miljø som servere og services skal udsættes for  
Sørg for hærdning

Nedenstående kan bruges mod andre typer servere!

## Sikringsforanstaltninger:

- Opdateret software - ingen kendte sikkerhedshuller eller sårbarheder
- fjern **single points of failure** - er man afhængig af en ressource skal man ofte have en backup mulighed, redundant strøm eller lignende
- adskilte servere - interne og eksterne til forskellige formål  
Eksempelvis den interne postserver hvor alle e-mail opbevares og en DMZ-postserver hvor ekstern post opbevares kortvarigt
- lav filtre på netværket, eller på data - firewalls og proxy funktioner
- begræns adgangen til at læse information
- begræns adgangen til at skrive information - dynamic updates på BIND, men samme princip til webløsninger og opdatering af databaser
- **least privileges** - sorg for at programmer og brugere kun har de nødvendige rettigheder til at kunne udføre opgaver
- følg med på områderne der har relevans for virksomheden og *jeres* installation - Windows, UNIX, BIND, Oracle, ...

# Change management

Er der tilstrækkeligt med fokus på software i produktion

Kan en vilkårlig server nemt reetableres

Foretages rettelser direkte på produktionssystemer

Er der fall-back plan

Burde være god systemadministrator praksis

# Fundamentet skal være iorden

Sørg for at den infrastruktur som I bygger på er sikker:

- redundans
- opdateret
- dokumenteret
- nem at vedligeholde

Husk tilgængelighed er også en sikkerhedsparameter

# CVS til konfigurationsfiler

Det anbefales at bruge versionsstyring som eksempelvis CVS til konfigurationsfiler

Det kan eksempelvis gøres på følgende måde:

- mkdirhier /security6.net/cvshome /security6.net/etc
- export CVSROOT=/security6.net/cvshome
- cvs init - for at initialisere CVS repository
- derefter kan man tilføje filer og lave CVS checkin og checkout
- CVS bruger standard filrettigheder - så opret eventuelt en speciel gruppe til CVS brugere

Læs mere om CVS eksempelvis på:

<http://cvsbook.red-bean.com/cvsbook.html>

# CVS eksempel med /etc/fstab

```
# cd /security6.net/etc
# cvs import -m "initial CVS af etc" etc hlk start
# cd ..;rm -rf etc
# cvs co etc
cvs checkout: Updating etc
# cp /etc/fstab etc
# cvs add etc/fstab
cvs add: scheduling file 'fstab' for addition
cvs add: use 'cvs commit' to add this file permanently
# cvs commit -m "fstab initial"
cvs commit: Examining .
RCS file: /security6.net/cvshome/etc/fstab,v
done
Checking in fstab;
/security6.net/cvshome/etc/fstab,v  <--  fstab
initial revision: 1.1
done
```

filer i /etc bør ikke flyttes, andre kan flyttes og sym-linkes

ssh root@server1



Mange UNIX systemer administreres fejlagtigt ved brug af root-login

Undgå direkte root-login

Insister på sudo eller su

Hvorfor?

- Sporbarheden mistes hvis brugere logger direkte ind som root
- Hvis et kodeord til root gættes er der direkte adgang til alt!

# SMTP Simple Mail Transfer Protocol



```
hlk@bigfoot:hlk$ telnet mail.kramse.dk 25
Connected to sunny.
220 sunny.kramse.dk ESMTP Postfix
HELO bigfoot
250 sunny.kramse.dk
MAIL FROM: Henrik
250 Ok
RCPT TO: hlk@kramse.dk
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
hejsa
.
250 Ok: queued as 749193BD2
QUIT
221 Bye
```

RFC-821 SMTP Simple Mail Transfer Protocol fra 1982

RFC-2821 fra 2001 og flere andre er idag gældende

[http://en.wikipedia.org/wiki/Simple\\_Mail\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)

Vedhæftede filer kodes i MIME Multipurpose Internet Mail Extensions

Bemærk at MIME encoding forøger størrelsen med ca. 30%!

# e-mail servere

Sendmail, qmail og postfix

Tre meget brugte e-mail systemer

- Sendmail - den ældste og mest benyttede
- Postfix en modulært og sikkerhedsmæssigt god e-mail server  
er ligeledes nem at konfigurere
- Qmail - en underlig mailserver lavet af Dan J Bernstein, med en speciel licens - ligesom programmøren

Dertil kommer diverse andre mailservere:

Microsoft Exchange på Windows servere

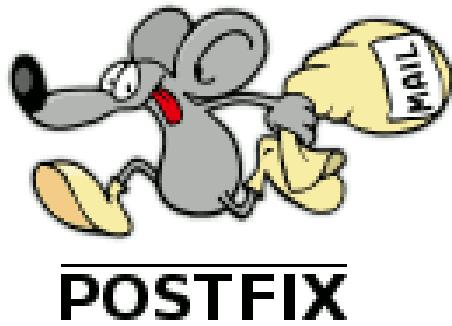
Jeg anbefaler at man har en postserver mod internet, der kun sender og modtager ekstern post, og en intern postserver der opbevarer al posten

# Sendmail postserveren

```
# "Smart" relay host (may be null)
DS
...
# strip group: syntax (not inside angle brackets!) and trailing semicolon
R$*                                $: $1 <@>                         mark addresses
R$* < $* > $* <@>                 $: $1 < $2 > $3                   unmark <addr>
R@ $* <@>                           $: @ $1                         unmark @host:...
R$* [ IPv6 : $+ ] <@>             $: $1 [ IPv6 : $2 ]                unmark IPv6 addr
R$* :: $* <@>                     $: $1 :: $2                      unmark node::addr
R:include: $* <@>                  $: :include: $1                  unmark :include:...
R$* : $* [ $* ]                      $: $1 : $2 [ $3 ] <@>            remark if leading colon
R$* : $* <@>                       $: $2                          strip colon if marked
```

mange konfigurerer Sendmail med `sendmail.cf` - det er **forkert**  
man bør bruge M4 konfigurationen  
- desværre følger M4 filerne sjældent med :-(  
Sendmail er oprindeligt lavet af Eric Allman

# Postfix postserveren



Lavet af Wietse Venema for IBM

Nem at konfigurere og sikker

`main.cf` findes typisk i kataloget `/etc/postfix`

# Audit af postservere

Security

.net

Typisk findes konfigurationsfilerne til postservere under /etc

- /etc/mail
- /etc/postfix

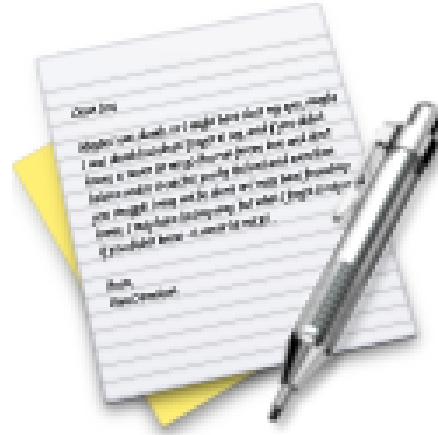
Det vigtigste er at den er opdateret og IKKE tillader relaying

Der findes diverse test-scripts til relaycheck på internet

Husk også at checke domæne records, MX og A

# Test af e-mail server

```
[hlk]$ telnet localhost 25
Connected.
Escape character is '^]'.
220 server ESMTP Postfix
helo test
250 server
mail from: postmaster@pentest.dk
250 Ok
rcpt to: root@pentest.dk
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
skriv en kort besked
.
250 Ok: queued as 91AA34D18
quit
```



Vi laver nu øvelsen

## Konfiguration af e-mail server

som er øvelse **32** fra øvelseshæftet.

# Postservere til klienter

SMTP som vi har gennemgået er til at sende mail mellem servere

Når vi skal hente post sker det typisk med POP3 eller IMAP

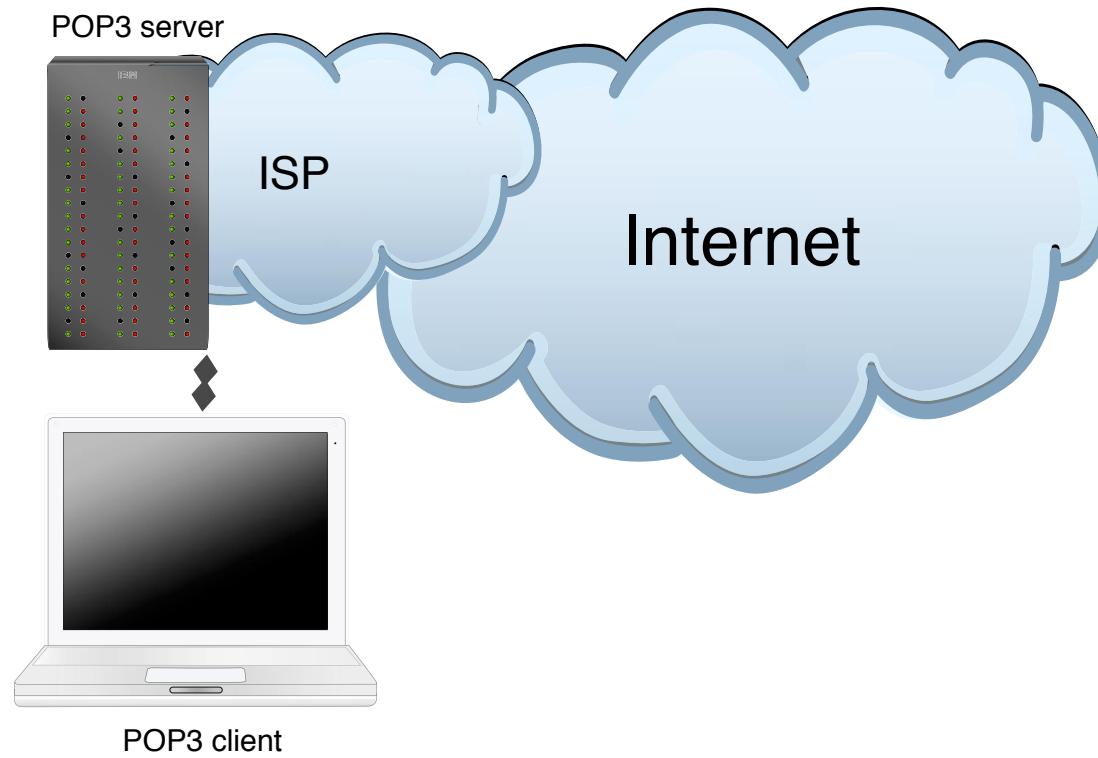
- POP3 Post Office Protocol version 3 RFC-1939
- Internet Message Access Protocol (typisk IMAPv4) RFC-3501

Forskellen mellem de to er at man typisk med POP3 henter posten, hvor man med IMAP lader den ligge på serveren

POP3 er bedst hvis kun en klient skal hente

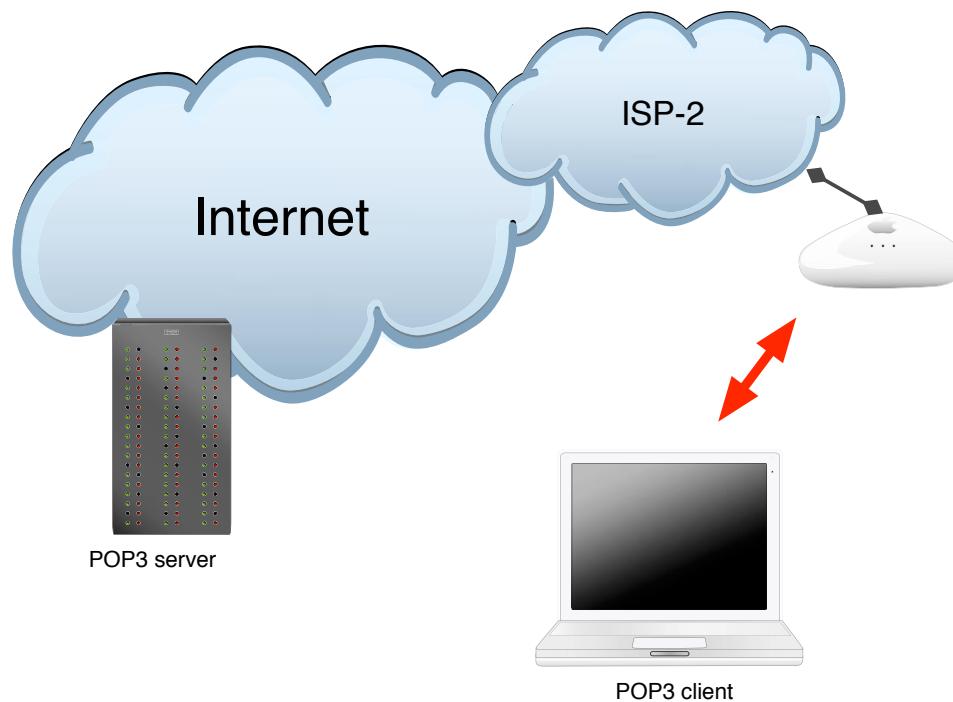
IMAP er bedst hvis du vil tilgå din post fra flere systemer

Jeg bruger selv IMAPS, IMAP over SSL kryptering - idet kodeord ellers sendes i klartekst



Man har tillid til sin ISP - der administrerer såvel net som server

# POP3 i Danmark - trådløst

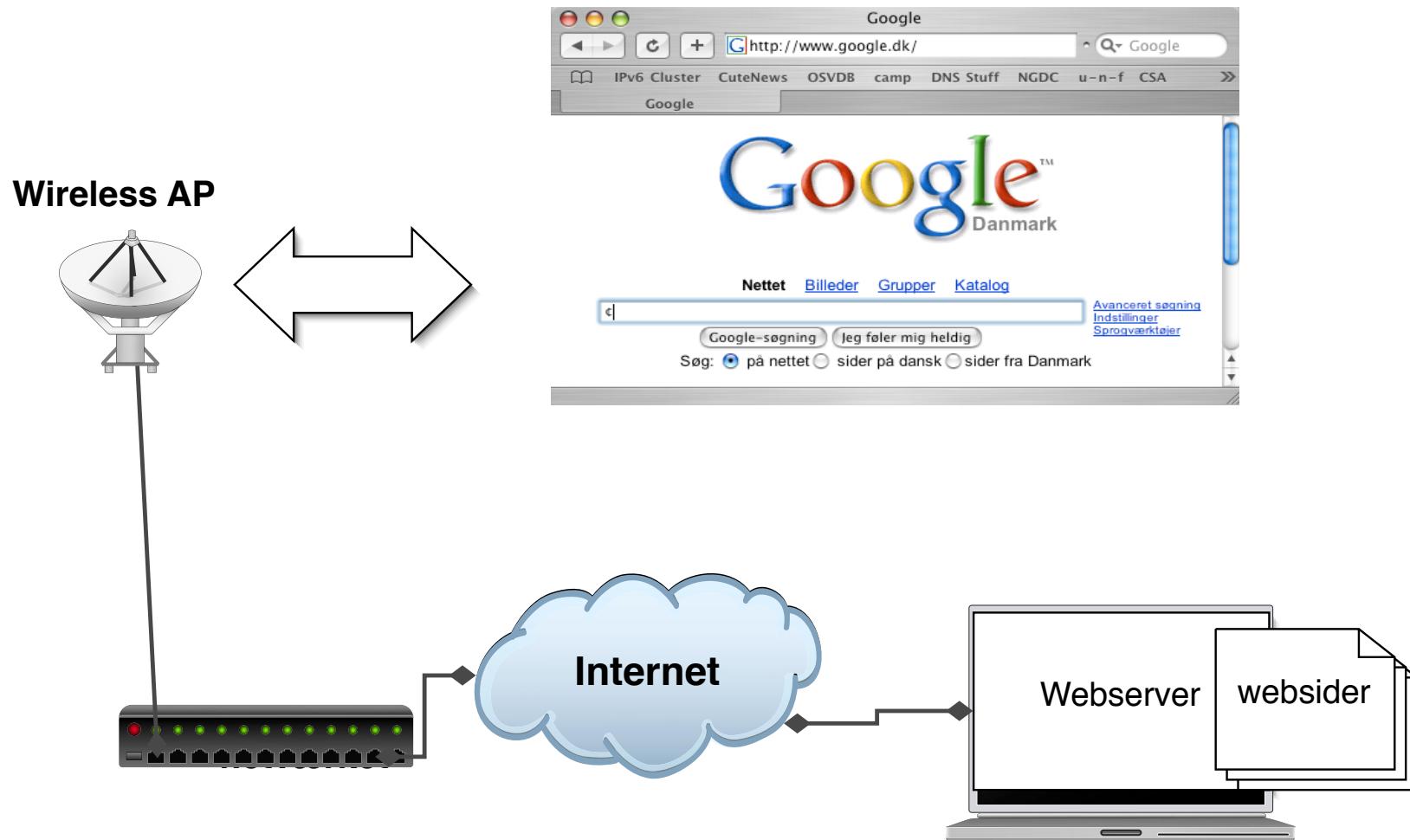


Har man tillid til andre ISP'er? Alle ISP'er?

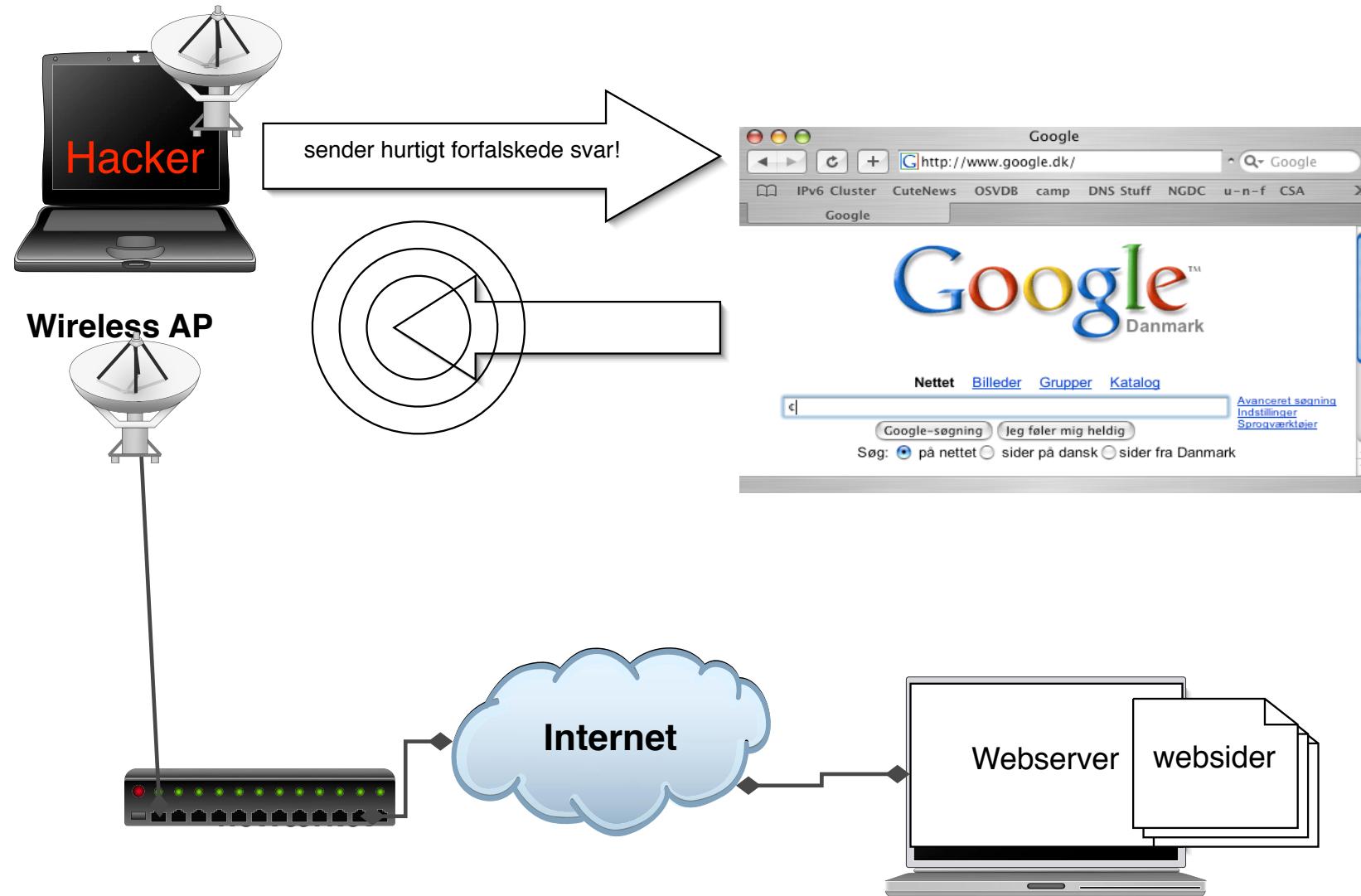
Deler man et netværksmedium med andre?

**Brug de rigtige protokoller!**

# Normal WLAN brug



# Packet injection - airpwn



## Airpwn teknikker

Klienten sender forespørgsel

Hackerens program airpwn lytter og sender så falske pakker

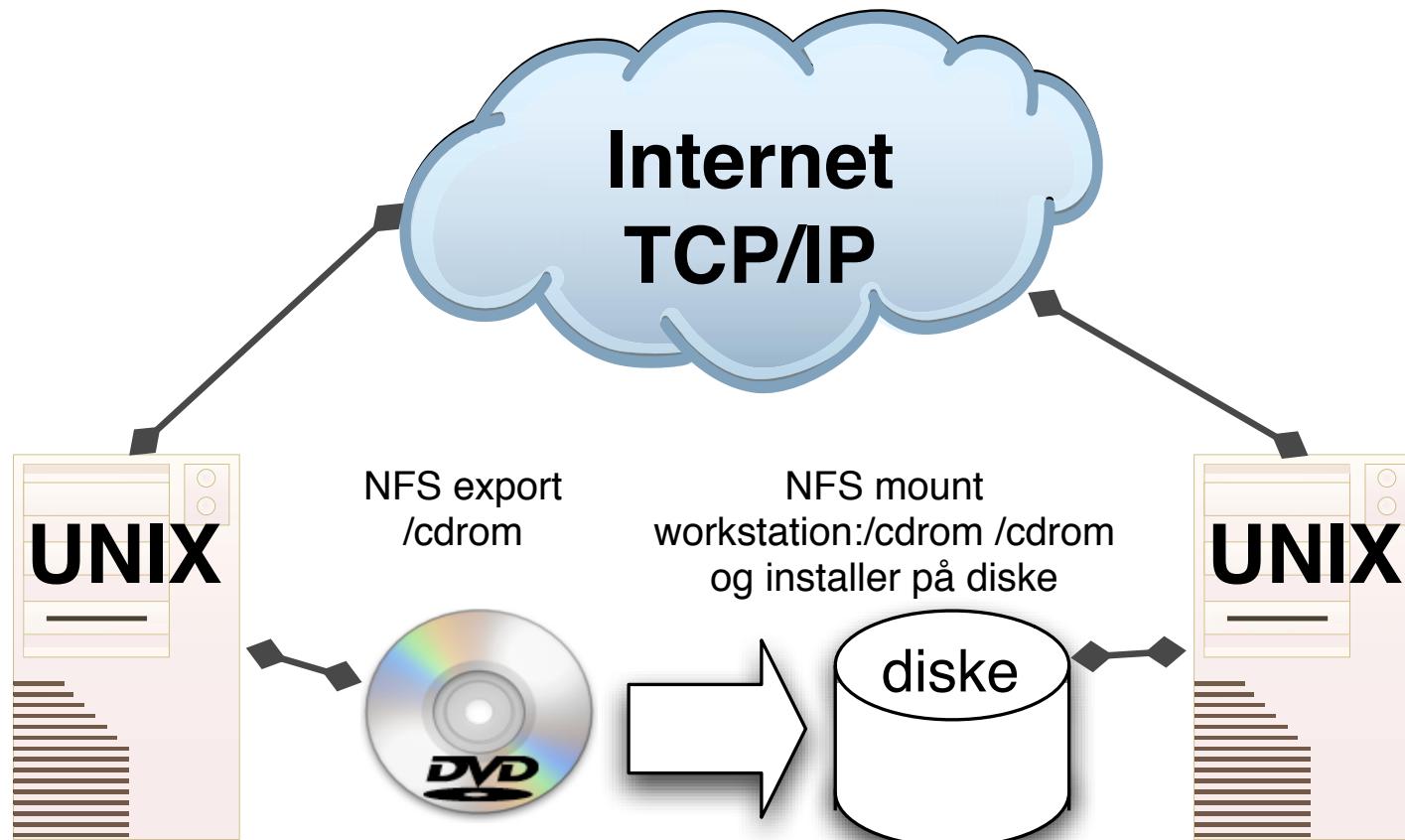
Hvordan kan det lade sig gøre?

- Normal forespørgsel og svar på Internet tager 50ms
- Airpwn kan svare på omkring 1ms angives det
- Airpwn har alle informationer til rådighed

Airpwn på Defcon 2004 - findes på Sourceforge

<http://airpwn.sourceforge.net/>

NB: Airpwn som demonstreret er begrænset til TCP og ukrypterede forbindelser



Hvorfor skal man stå ved serveren for at installere?

# Distribuerede filsystemer

Til lokalnetværk:

- Windows filesharing - tidligere et stort sikkerhedshul
- UNIX NFS - ikke beregnet til nutidens usikre netværk

Over Internet:

AFS - Andrew filesystem

<http://www-2.cs.cmu.edu/afs/andrew.cmu.edu/usr/shadow/www/afs.html>

CODA <http://www.coda.cs.cmu.edu/>

Tænk på de forudsætninger som et program har og forventer er til stede!

# NFS - netværksfilsystem

```
# sample /etc/exports file
/
/master (rw) trusty(rw,no_root_squash)
/projects proj*.local.domain(rw)
/usr *.local.domain(ro) @trusted(rw)
/home/joe pc001(rw,all_squash,anonuid=150,anongid=100)
/pub (ro,insecure,all_squash)
```

- UNIX NFS er netværksfilsystemet som alle UNIX varianter understøtter
- Adgangen styres ved brug af /etc/exports, eksempel fra manualen på Red Hat
- De fleste bruger version 3 over UDP eller TCP selvom version 4 burde have bedre sikkerhed
- Adgangen gives pr IP-adresse! IP adressebaseret autentifikation er pr definition dårlig!
- Pas på - det er nemt at give root-adgang til andre maskiner!



Vi laver nu øvelsen

## RPC info

som er øvelse **33** fra øvelseshæftet.



Vi laver nu øvelsen

## NFS info

som er øvelse **34** fra øvelseshæftet.

# Samba og SMB/CIFS

Microsoft Server Message Block bruges til netværksdrev og netværksprint i Windows miljøer

Samba er en open source implementation, som altid halter bagefter MS

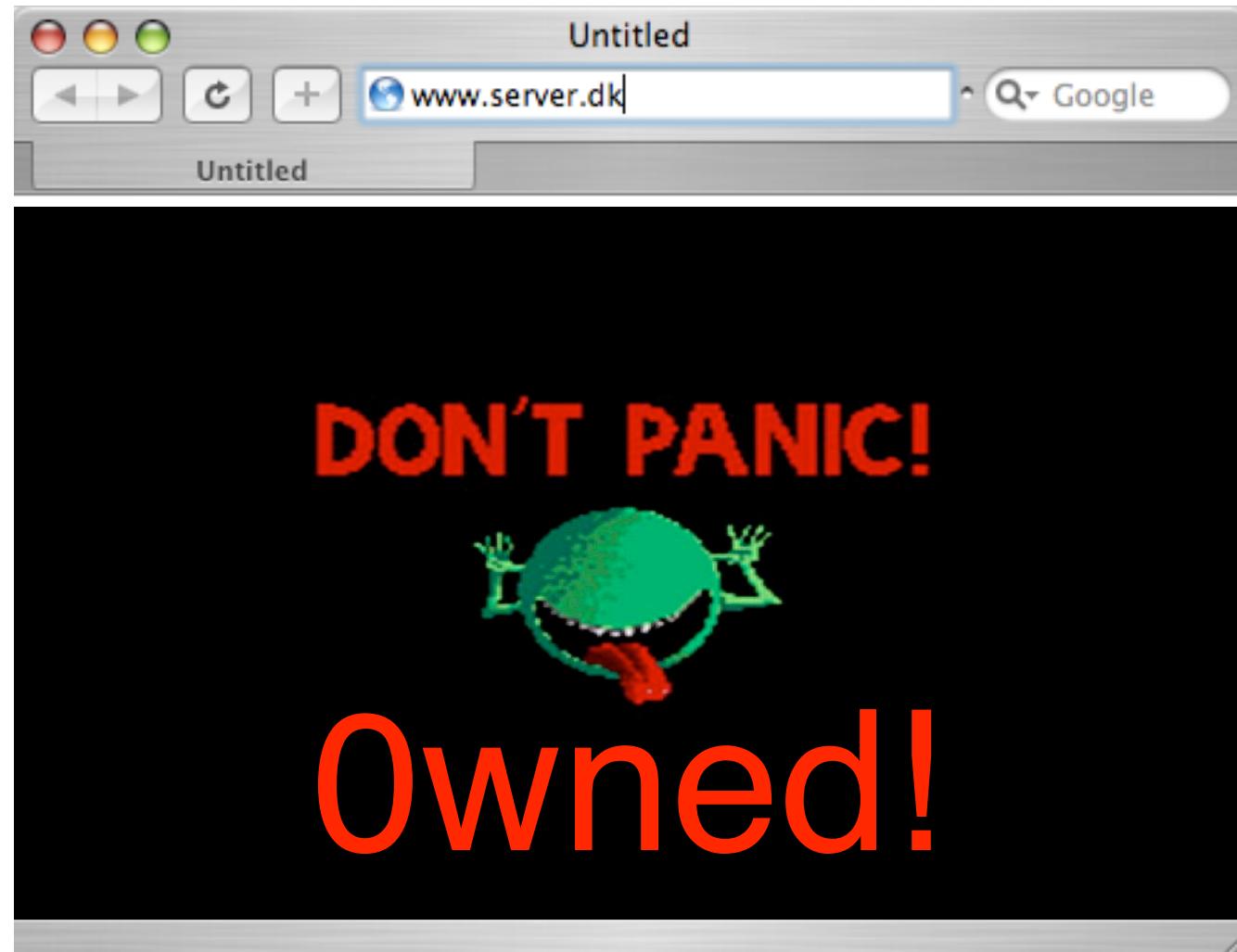
De gamle implementationer overfører password i en uheldig version, som kan knækkes  
7 tegn ad gangen - hurtigt

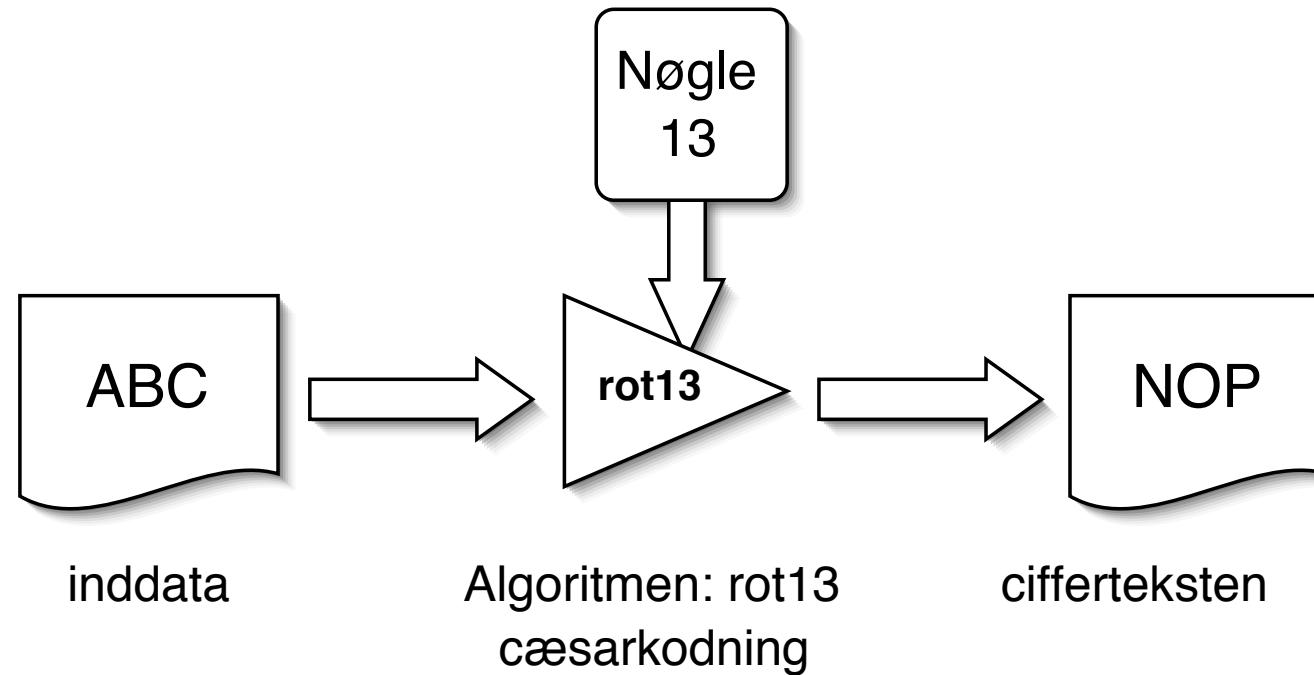
Microsoft har gennem tiden opdateret protokollen

Idag forsøger man at gøre det til en standard som Common Internet File System Protocol CIFS

Man kan læse om Samba på hjemmesiden <http://www.samba.org/>

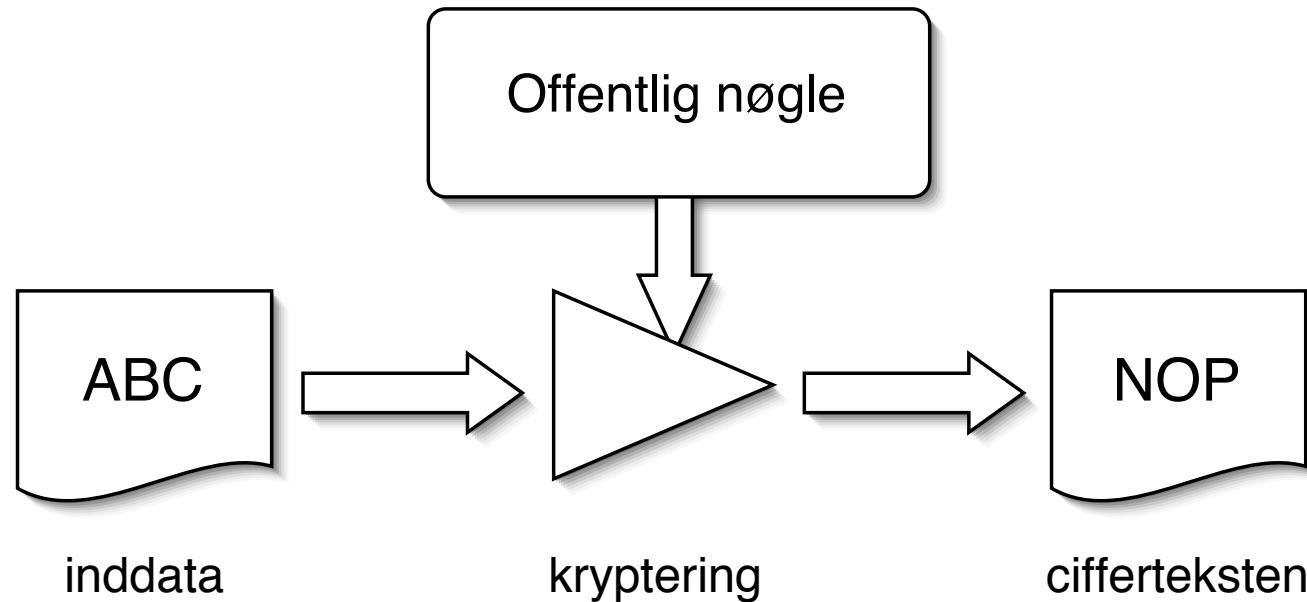
# Dag 4 Netværkssikkerhed og firewalls





Kryptografi er læren om, hvordan man kan kryptere data

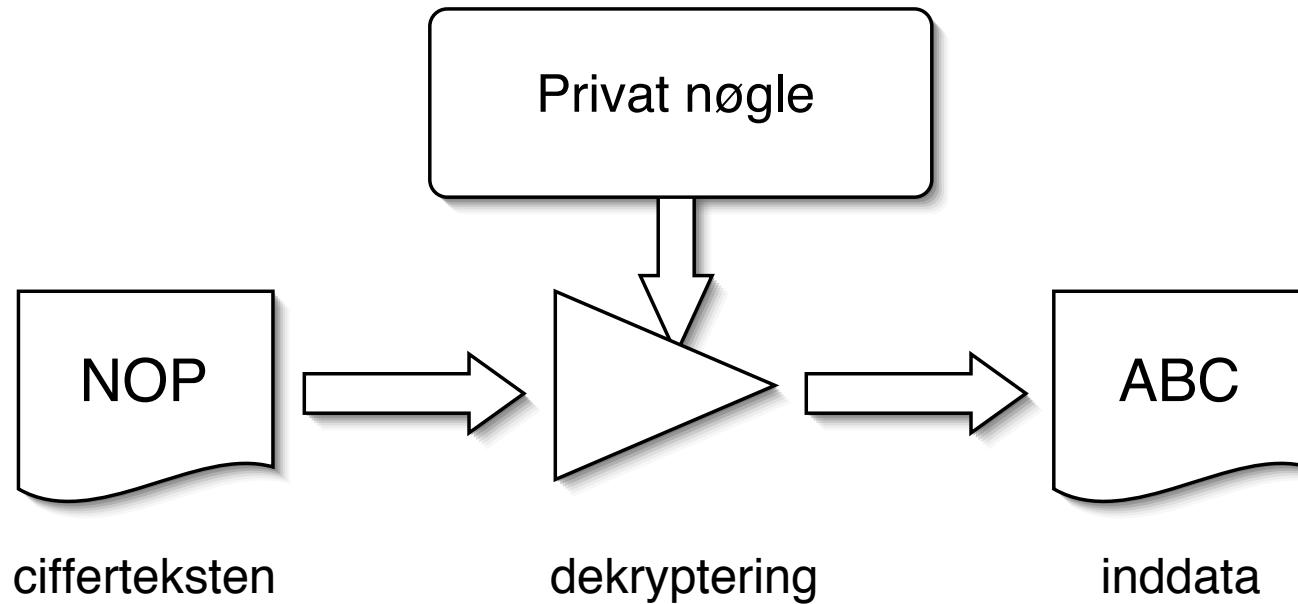
Kryptografi benytter algoritmer som sammen med nøgler giver en ciffertekst - der kun kan læses ved hjælp af den tilhørende nøgle



privat-nøgle kryptografi (eksempelvis AES) benyttes den samme nøgle til kryptering og dekryptering

offentlig-nøgle kryptografi (eksempelvis RSA) benytter to separate nøgler til kryptering og dekryptering

# Public key kryptografi - 2



offentlig-nøgle kryptografi (eksempelvis RSA) bruger den private nøgle til at dekryptere  
man kan ligeledes bruge offentlig-nøgle kryptografi til at signere dokumenter - som så  
verificeres med den offentlige nøgle

# Kryptografiske principper

Algoritmerne er kendte

Nøglerne er hemmelige

Nøgler har en vis levetid - de skal skiftes ofte

Et successfult angreb på en krypto-algoritme er enhver genvej som kræver mindre arbejde end en gennemgang af alle nøglerne

Nye algoritmer, programmer, protokoller m.v. skal gennemgås nøje!

Se evt. Snake Oil Warning Signs: Encryption Software to Avoid <http://www.interhack.net/people/cmcurtin/snake-oil-faq.html>

# DES, Triple DES og AES

## AES

Advanced Encryption Standard

DES kryptering baseret på den IBM udviklede Lucifer algoritme har været benyttet gennem mange år.

Der er vedtaget en ny standard algoritme Advanced Encryption Standard (AES) som afløser Data Encryption Standard (DES)

Algoritmen hedder Rijndael og er udviklet af Joan Daemen og Vincent Rijmen.

Kilde: <http://csrc.nist.gov/encryption/aes/>  
<http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>

kryptering er den eneste måde at sikre:

fortrolighed

autenticitet / integritet

# e-mail og forbindelser

## Kryptering af e-mail

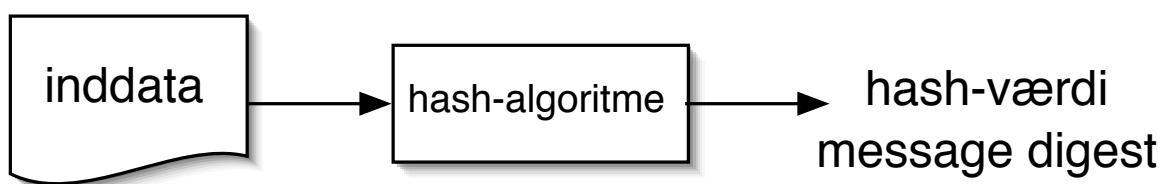
- Pretty Good Privacy - Phil Zimmermann
- PGP = mail sikkerhed

## Kryptering af sessioner SSL/TLS

- Secure Sockets Layer SSL / Transport Layer Services TLS
- krypterer data der sendes mellem webservere og klienter
- SSL kan bruges generelt til mange typer sessioner, eksempelvis POP3S, IMAPS, SSH m.fl.

Sender I kreditkortnummeret til en webserver der kører uden https?

# MD5 message digest funktion



HASH algoritmer giver en unik værdi baseret på input  
værdien ændres radikalt selv ved små ændringer i input

MD5 er blandt andet beskrevet i RFC-1321: The MD5 Message-Digest Algorithm  
Både MD5 og SHA-1 undersøges nøje og der er fundet kollisioner som kan påvirke  
vores brug i fremtiden - *stay tuned*

# 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

```
h1k@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>"
```

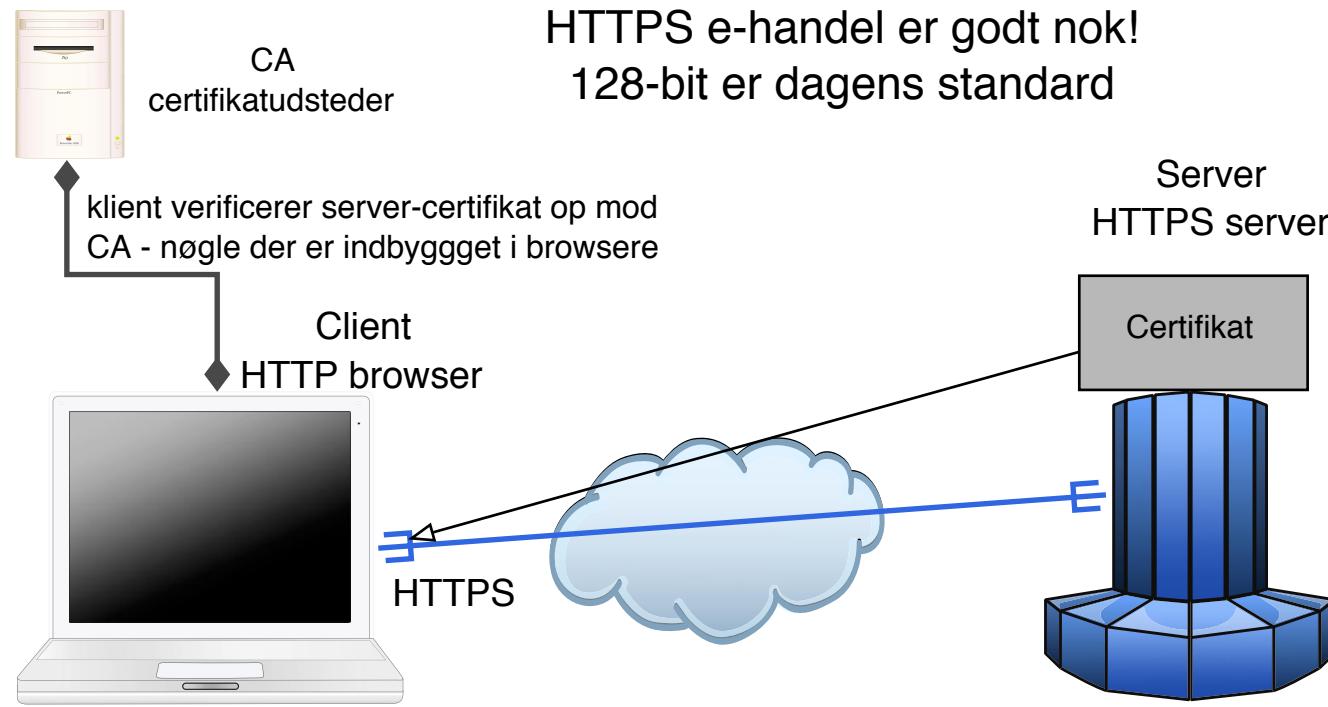
# Make and install programs from source

Mange open source programmer kommer som en tar-fil

De fleste C programmer benytter sig så af følgende kommando

- konfigurer softwaren - undersøg hvilket operativsystem det er
- byg software ved hjælp af en Makefile - kompilerer og linker
- installer software - ofte i /usr/local/bin

```
./configure;make;make install
```



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



## 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 brugerid/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

Protocol 2

PermitRootLogin no

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication no

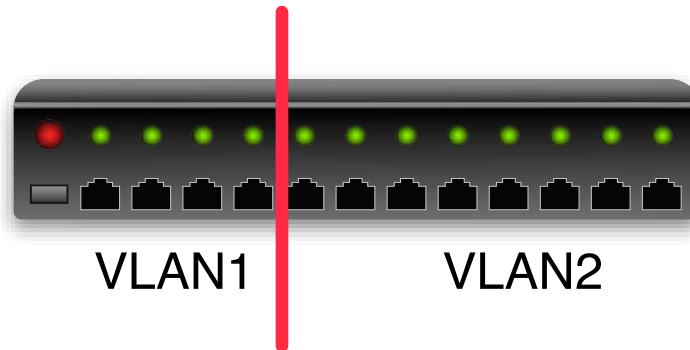
#X11Forwarding no

#X11DisplayOffset 10

#X11UseLocalhost yes

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

## Portbased VLAN



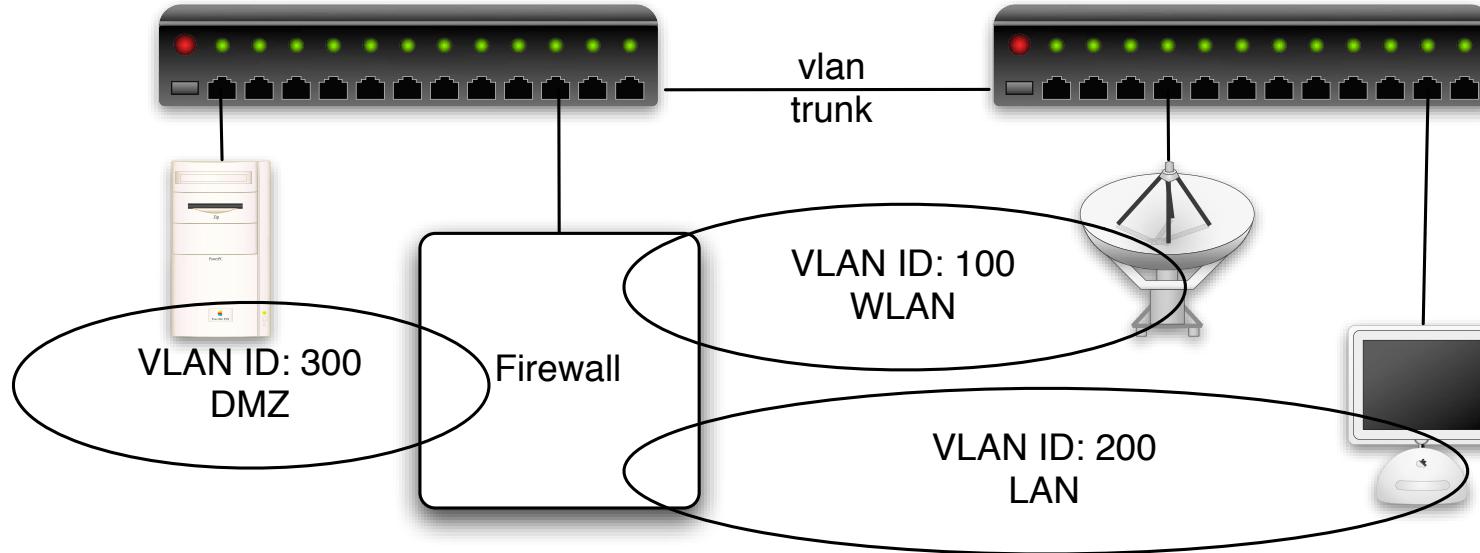
Nogle switcher tillader at man opdeler portene

Denne opdeling kaldes VLAN og portbaseret er det mest simple

Port 1-4 er et LAN

De resterende er et andet LAN

Data skal omkring en firewall eller en router for at krydse fra VLAN1 til VLAN2



Nogle switcher tillader at man opdeler portene, men tillige benytter 802.1q

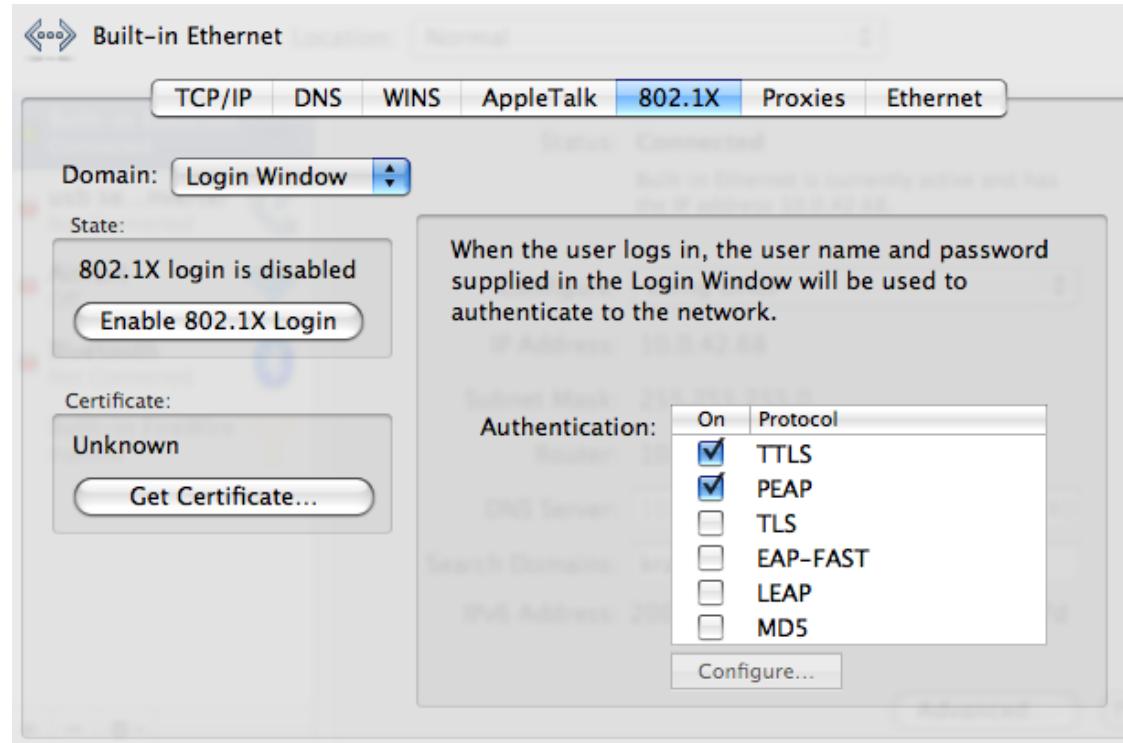
Med 802.1q tillades VLAN tagging på Ethernet niveau

Data skal omkring en firewall eller en router for at krydse fra VLAN1 til VLAN2

VLAN trunking giver mulighed for at dele VLANs ud på flere switches

Der findes administrationsværktøjer der letter dette arbejde: OpenNAC FreeNAC, Cisco VMPS

# IEEE 802.1x Port Based Network Access Control



Nogle switcher tillader at man benytter 802.1x

Denne protokol sikrer at man valideres før der gives adgang til porten

Når systemet skal have adgang til porten afleveres brugernavn og kodeord/certifikat

Denne protokol indgår også i WPA Enterprise

## 802.1x og andre teknologier

802.1x i forhold til MAC filtrering giver væsentlige fordele

MAC filtrering kan spoofes, hvor 802.1x kræver det rigtige kodeord

Typisk benyttes RADIUS og 802.1x integrerer således mod både LDAP og Active Directory

# Hvad er en firewall



En firewall er noget som **blokerer** traffik på Internet

En firewall er noget som **blokerer** traffik på Internet

En firewall er noget som **tillader** traffik på Internet

# Firewallrollen idag

Idag skal en firewall være med til at:

- Forhindre angribere i at komme ind
- Forhindre angribere i at sende traffik ud
- Forhindre virus og orme i at sprede sig i netværk
- Indgå i en samlet løsning med ISP, routere, firewalls, switchede strukturer, intrusion detectionsystemer samt andre dele af infrastrukturen

Det kræver overblik!

# firewalls

Basalt set et netværksfilter - det yderste fæstningsværk

Indholder typisk:

- Grafisk brugergrænseflade til konfiguration - er det en fordel?
- TCP/IP filtermuligheder - pakernes afsender, modtager, retning ind/ud, porte, protokol, ...
- Kun IPv4 for de fleste kommercielle firewalls
- Både IPv4 og IPv6 for Open Source firewalls: IPF, OpenBSD PF, Linux firewalls, ...
- Foruddefinerede regler/eksempler - er det godt hvis det er nemt at tilføje/åbne en usikker protokol?
- Typisk NAT funktionalitet indbygget
- Typisk mulighed for nogle serverfunktioner: kan agere DHCP-server, DNS caching server og lignende

En router med Access Control Lists - ACL kaldes ofte netværksfilter, mens en dedikeret maskine kaldes firewall - funktionen er reelt den samme - der filtreres trafik

# Packet filtering

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
+-----+																							
Version   IHL   Type of Service														Total Length									
+-----+																							
Identification   Flags								Fragment Offset															
+-----+																							
Time to Live				Protocol				Header Checksum															
+-----+																							
Source Address																							
+-----+																							
Destination Address																							
+-----+																							
Options																Padding							
+-----+																							

Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste statefull inspection

# Kommercielle firewalls

- Checkpoint Firewall-1 <http://www.checkpoint.com>
- Nokia appliances - Nokia IPSO <http://www.nokia.com>
- Cisco PIX <http://www.cisco.com>
- Clavister firewalls <http://www.clavister.com>
- Netscreen - nu ejet af Juniper <http://www.juniper.net>

Ovenstående er dem som jeg oftest ser ude hos mine kunder

# Open source baserede firewalls

## Linux firewalls

- Linux firewalls - fra begyndelsen til det nuværende netfilter til kerner version 2.4 og 2.6  
<http://www.netfilter.org>
- Firewall GUIs ovenpå Linux - mange! IPcop, Guarddog, Watchguard nogle Linux firewalls er kommercielle produkter
- IP Filter (IPF) <http://coombs.anu.edu.au/~avalon/>
- OpenBSD PF - findes idag på andre operativsystemer <http://www.openbsd.org>
- FreeBSD IPFW og IPFW2 <http://www.freebsd.org>
- Mac OS X benytter IPFW
- FreeBSD inkluderer også OpenBSD PF
- NetBSD - bruger IPF og er ved at inkludere OpenBSD PF

NB: kun eksempler og dem jeg selv har brugt

# Hardware eller software

Man hører indimellem begrebet *hardware firewall*

Det er dog et faktum at en firewall består af:

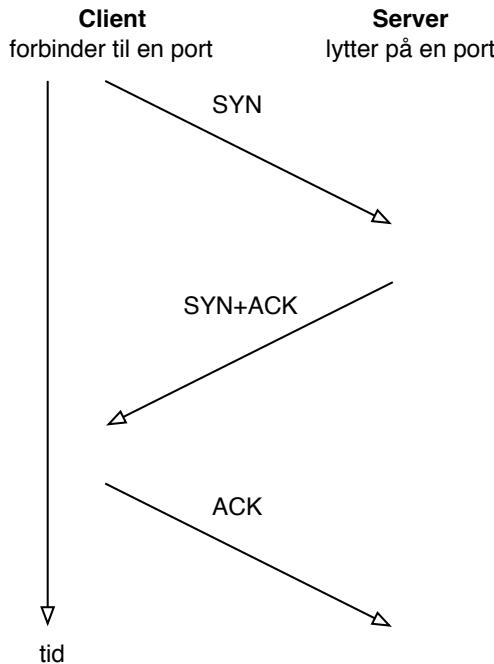
- Netværkskort - som er hardware
- Filtreringssoftware - som er *software!*

Det giver ikke mening at kalde en Zyxel 10 en hardware firewall og en Soekris med OpenBSD for en software firewall!

Det er efter min mening et marketingtrick

Man kan til gengæld godt argumentere for at en dedikeret firewall som en separat enhed kan give bedre sikkerhed

# TCP three way handshake



- **TCP SYN half-open** scans
- Tidligere loggede systemer kun når der var etableret en fuld TCP forbindelse - dette kan/kunne udnyttes til *stealth*-scans
- Hvis en maskine modtager mange SYN pakker kan dette fylde tabellen over connections op - og derved afholde nye forbindelser fra at blive oprette - **SYN-flooding**

# firewall regelsæt eksempel

```
# hosts
router="217.157.20.129"
webserver="217.157.20.131"
# Networks
homenet=" 192.168.1.0/24, 1.2.3.4/24 "
wlan="10.0.42.0/24"
wireless=wi0

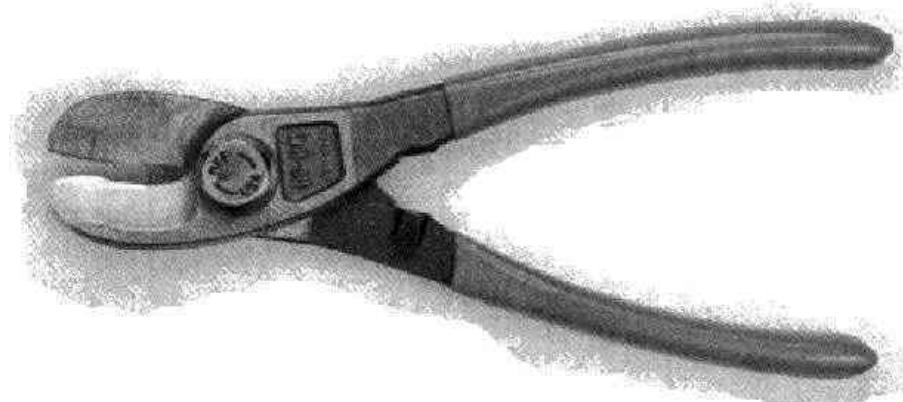
# things not used
spoofed=" 127.0.0.0/8, 172.16.0.0/12, 10.0.0.0/16, 255.255.255.255/32 "

block in all # default block anything
# loopback and other interface rules
pass out quick on lo0 all
pass in quick on lo0 all

# egress and ingress filtering - disallow spoofing, and drop spoofed
block in quick from $spoofed to any
block out quick from any to $spoofed

pass in on $wireless proto tcp from $wlan to any port = 22
pass in on $wireless proto tcp from $homenet to any port = 22
pass in on $wireless proto tcp from any to $webserver port = 80

pass out quick proto tcp from $homenet to any flags S/S keep state
pass out quick proto udp from $homenet to any keep state
pass out quick proto icmp from $homenet to any keep state
```



Hvor skal en firewall placeres for at gøre størst nytte?

Hvad er forudsætningen for at en firewall virker?

At der er konfigureret et sæt fornuftige regler!

Hvor kommer reglerne fra? Sikkerhedspolitikken!

Kilde: Billedet er fra Marcus Ranum The ULTIMATELY Secure Firewall

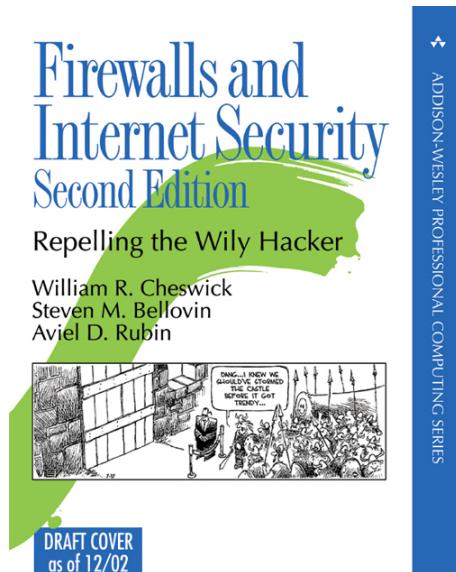
# Firewall er ikke alene

Firewalls er ikke alene

- anti-virus på klienter og postsystemer
- IDS systemer
- Backupsystemer
- Adgangskontrol
- ... mange andre ting er mindst ligeså vigtige

Forsvaret er som altid - flere lag af sikkerhed!

# Firewall historik



Firewalls har været kendt siden starten af 90'erne

Den første bog *Firewalls and Internet Security* udkom i 1994 men der findes mange akademiske artikler om firewalls

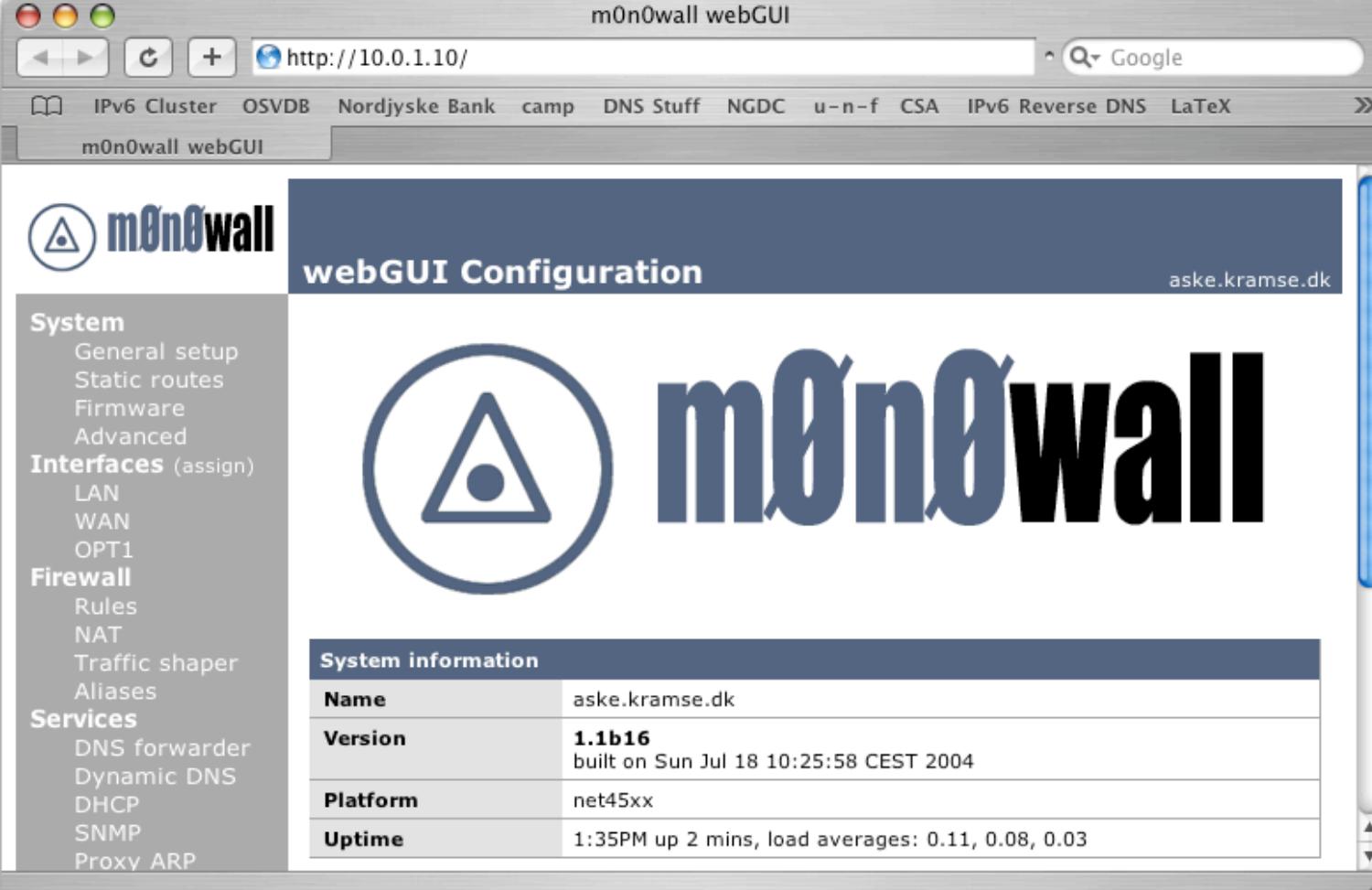
Bogen *Firewalls and Internet Security* anbefales, William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, Addison-Wesley, 2nd edition, 2003

Artikel om en hacker der lokkes, vurderes, overvåges

Et tidligt eksempel på en honeypot

Idag anbefales The Honeynet Project hvis man vil vide mere

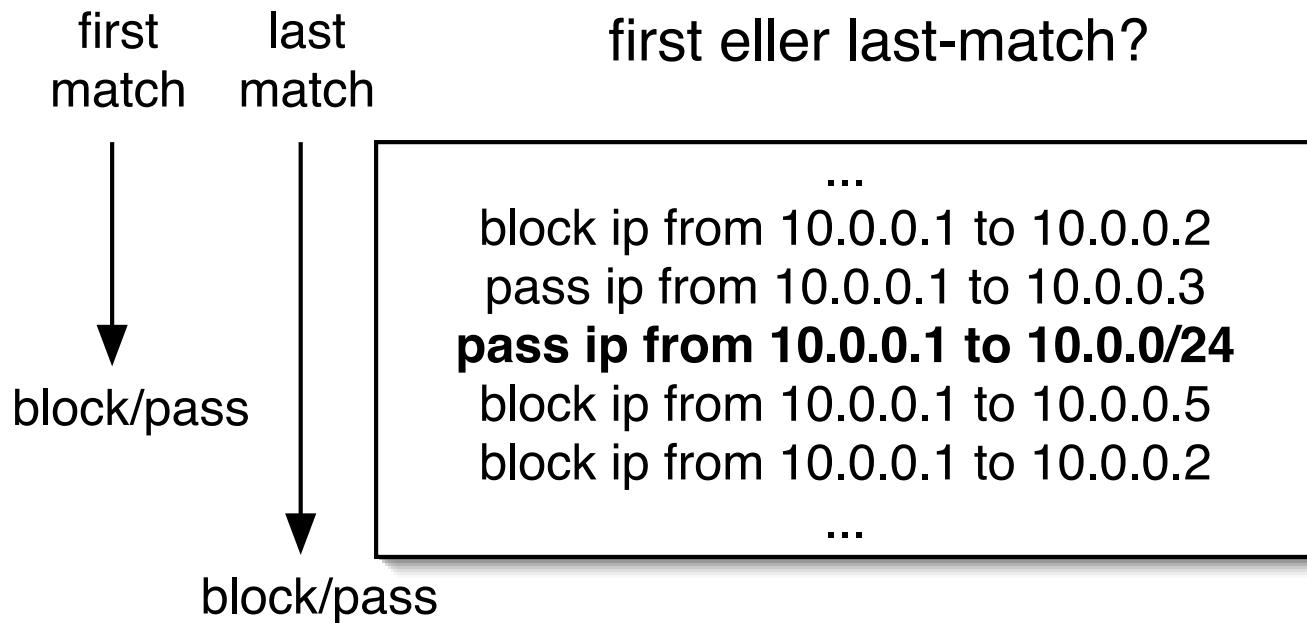
<http://www.honeynet.org>



The screenshot shows a web browser window titled "m0n0wall webGUI" with the URL "http://10.0.1.10/" in the address bar. The browser interface includes standard navigation buttons (back, forward, search, etc.) and a tab bar with multiple items like "IPv6 Cluster", "OSVDB", "Nordjyske Bank", "camp", "DNS Stuff", "NGDC", "u-n-f", "CSA", "IPv6 Reverse DNS", and "LaTeX". The main content area is titled "webGUI Configuration" and features the "m0n0wall" logo. On the left, a sidebar menu lists "System" (General setup, Static routes, Firmware, Advanced), "Interfaces (assign)" (LAN, WAN, OPT1), "Firewall" (Rules, NAT, Traffic shaper, Aliases), and "Services" (DNS forwarder, Dynamic DNS, DHCP, SNMP, Proxy ARP). The right side displays "System information" in a table:

System information	
Name	aske.kramse.dk
Version	<b>1.1b16</b> built on Sun Jul 18 10:25:58 CEST 2004
Platform	net45xx
Uptime	1:35PM up 2 mins, load averages: 0.11, 0.08, 0.03

# First or Last match firewall?



Med dette regelsæt vil en first-match firewall blokere pakker fra 10.0.0.1 til 10.0.0.2 - men tillade alt andet fra 10.0.0.1 til 10.0.0/24

Med dette regelsæt vil en last-match firewall blokere pakker fra 10.0.0.1 til 10.0.0.2, **10.0.0.1 til 10.0.0.5**, **10.0.0.1 til 10.0.0.2** - men ellers tillade alt andet fra 10.0.0.1 til 10.0.0/24

Rækkefølgen af regler betyder noget!

- To typer af firewalls: First match - når en regel matcher, gør det som angives block/pass Last match - marker pakken hvis den matcher, til sidst afgøres block/pass

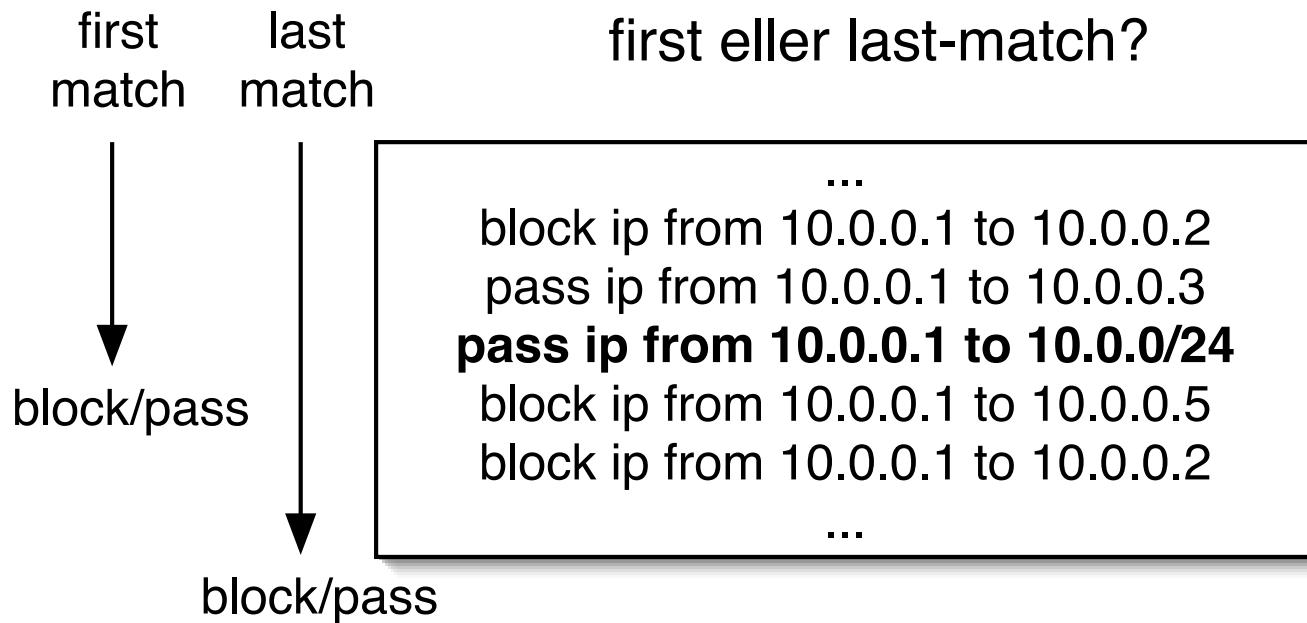
**Det er ekstremt vigtigt at vide hvilken type firewall man bruger!**

OpenBSD PF er last match

FreeBSD IPFW er first match

Linux iptables/netfilter er last match

# First or Last match firewall?



Med dette regelsæt vil en first-match firewall blokere pakker fra 10.0.0.1 til 10.0.0.2 - men tillade alt andet fra 10.0.0.1 til 10.0.0/24

Med dette regelsæt vil en last-match firewall blokere pakker fra 10.0.0.1 til 10.0.0.2, **10.0.0.1 til 10.0.0.5**, **10.0.0.1 til 10.0.0.2** - men ellers tillade alt andet fra 10.0.0.1 til 10.0.0/24

- To typer af firewalls: First match - eksempelvis IPFW, Last match - eksempelvis PF

## First match - IPFW

```
00100 16389 1551541 allow ip from any to any via lo0
00200      0      0 deny log ip from any to 127.0.0.0/8
00300      0      0 check-state
...
65435    36    5697 deny log ip from any to any
65535    865    54964 allow ip from any to any
```

Den sidste regel nås aldrig!

## Last match - OpenBSD PF

```
ext_if="ext0"
int_if="int0"

block in
pass out keep state

pass quick on { lo $int_if }

# Tillad forbindelser ind på port 80=http og port 53=domain
# på IP-adressen for eksterne netkort ($ext_if) syntaksen
pass in on $ext_if proto tcp to ($ext_if) port http keep state
pass in on $ext_if proto { tcp, udp } to ($ext_if) port domain keep state
```

Pakkerne markeres med block eller pass indtil sidste regel  
nøgleordet *quick* afslutter match - god til store regelsæt

# Linux iptables/netfilter eksempel

```
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

NB: husk at aktivere IP forwarding

# Firewall GUI



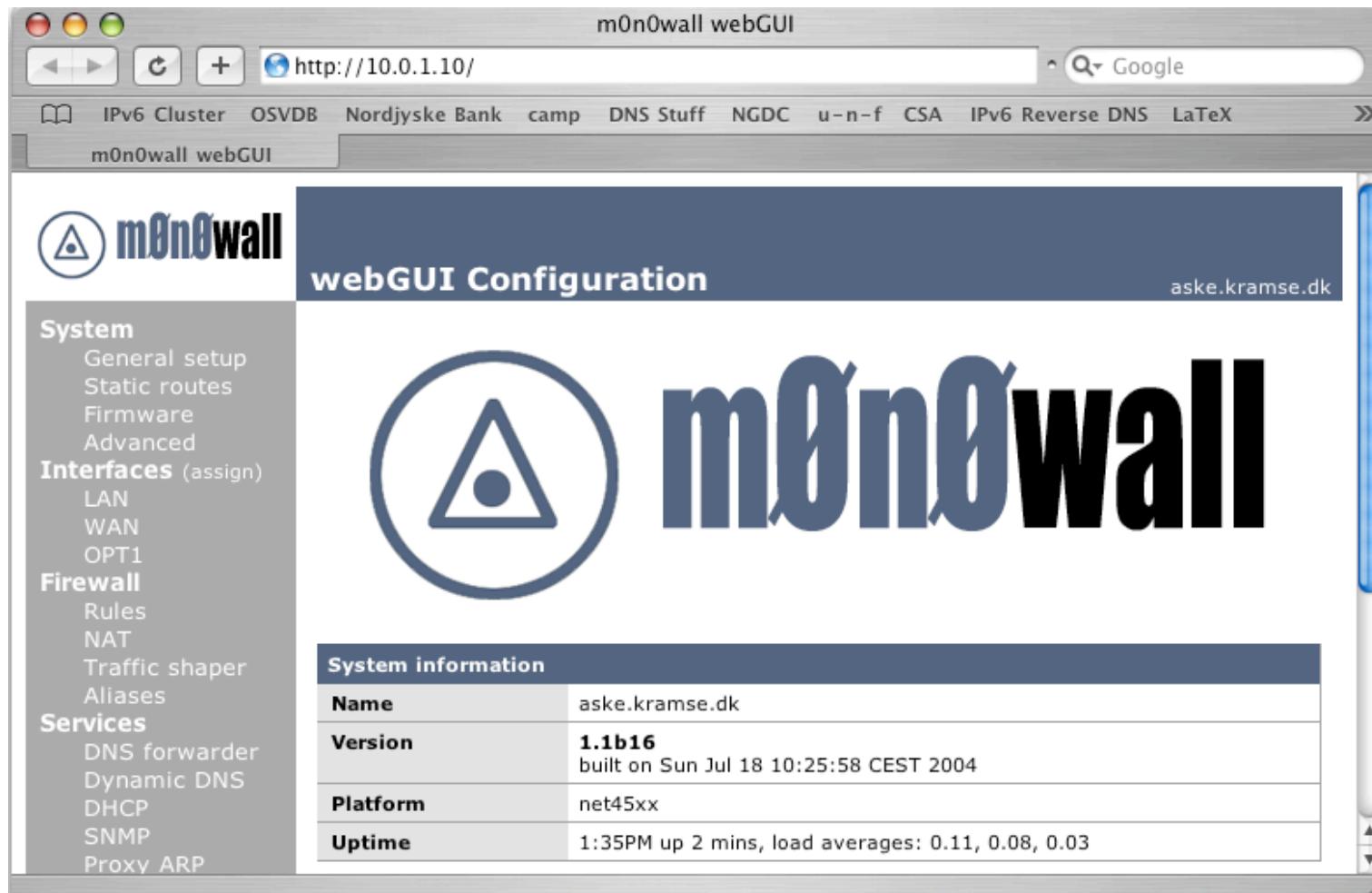
The screenshot shows a graphical user interface for managing firewall policies. On the left, a tree view titled 'User Standard' displays the structure of the firewall configuration, including 'Objects', 'Services', 'Firewalls', and specific interfaces like 'fw', 'eth1', 'eth0', and 'lo'. A 'Policy' node is selected. Below the tree is a 'NAT' section. On the right, a large table lists 8 firewall rules:

Num	Source	Destination	Service	Action	Time	Options	Comment
00		Any			Any		firewall uses DNS server on Inet
01	Any				Any		firewall serves as DNS server for LAN
02	Any				Any		firewall serves as DHCP server for LAN
03		Any			Any		firewall serves as DHCP server for LAN
04	Any		 		Any		mail and ftp server behind the firewall
05		Any			Any		
06					Any		ssh access to firewall from internal LAN
07	Any	Any	Any		Any		'catch all' rule

At the bottom right are 'Apply' and 'Undo' buttons.

Der findes mange GUI programmer til Open Source firewalls

Kilde: billede fra <http://www.fwbuilder.org>



Kilde: billede fra <http://m0n0.ch/wall/>

# Firewalls og ICMP

```
ipfw add allow icmp from any to any icmptypes 3,4,11,12
```

Ovenstående er IPFW syntaks for at tillade de interessant ICMP beskeder igennem  
Tillad ICMP types:

- 3 Destination Unreachable
- 4 Source Quench Message
- 11 Time Exceeded
- 12 Parameter Problem Message

# Firewall konfiguration

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

- En firewall skal passes
- En firewall skal opdateres
- Systemerne bagved skal hærdes!

# Blokér indefra og ud

Der er porte og services som altid bør blokeres

Det kan være kendte sårbare services

- Windows SMB filesharing - ikke til brug på Internet!
- UNIX NFS - ikke til brug på Internet!

Kendte problemer:

- KaZaA og andre P2P programmer - hvis muligt!
- Portmapper - port 111

# Firewall konfiguration

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

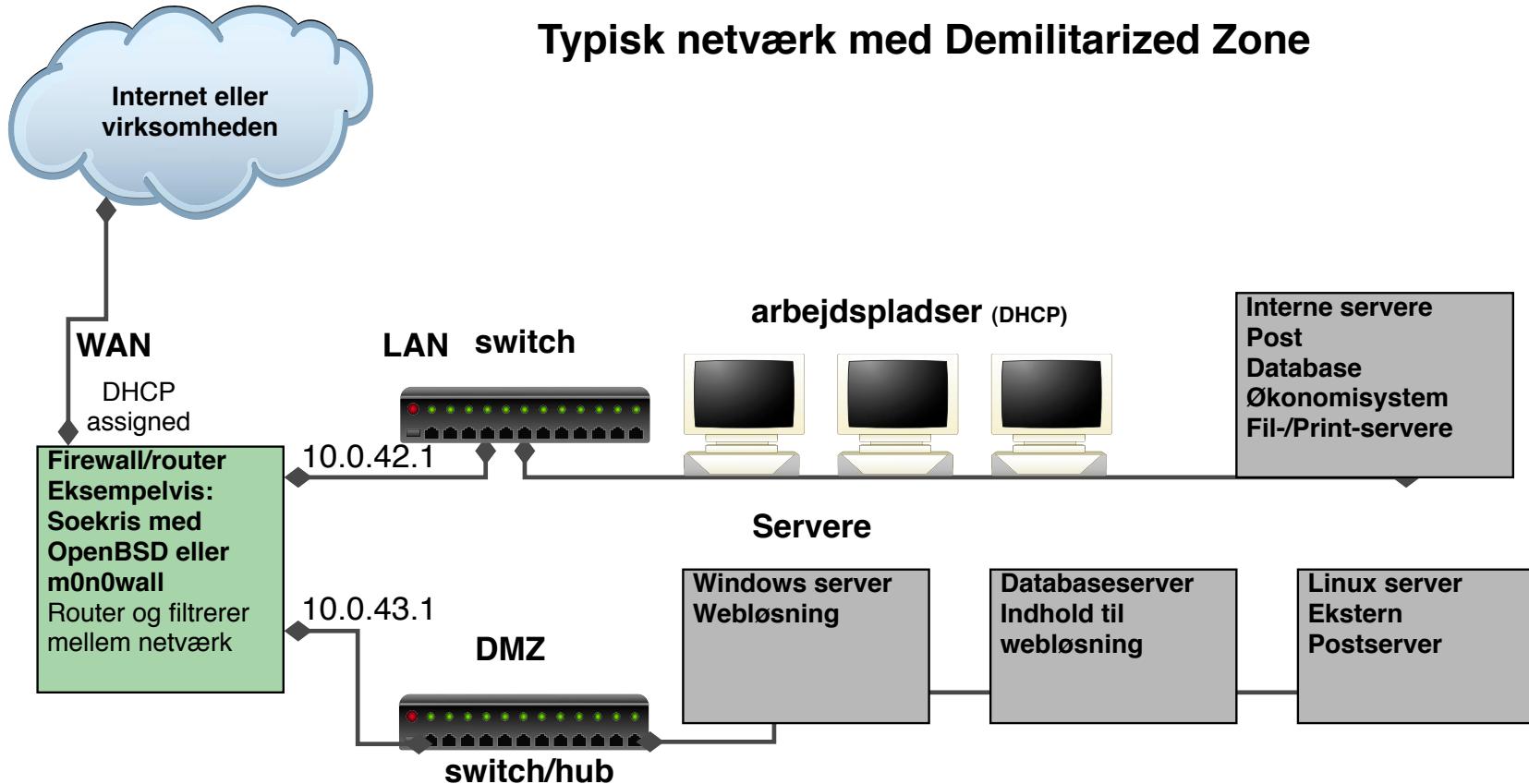
Brug dernæst en firewall med GUI første gang!

Husk dernæst:

- En firewall skal passes
- En firewall skal opdateres
- Systemerne bagved skal hærdes!

# En typisk firewall konfiguration

Typisk netværk med Demilitarized Zone



Opdeling i separate netværkssegmenter!

# personlige firewalls

Security

Personlige firewalls:

- Microsoft Windows XP
- ZoneAlarm <http://www.zonelabs.com>

Personlige firewalls til Microsoft Windows inkluderer ofte blokering af hvilket programmer der må tilgå netværk

Det anbefales at bruge en personlig firewall

Note: Lad være med at stille spørgsmål om logfilen i diverse fora!

**Hvis du ikke forstår loggen så lad den ligge!**

# Firewallværktøjer

Der benyttes på kurset en del værktøjer:

- nmap - <http://www.insecure.org> portscanner
- Nessus - <http://www.nessus.org> automatiseret testværktøj
- Ethereal - <http://www.ethereal.com> avanceret netværkssniffer
- OpenBSD - <http://www.openbsd.org> operativsystem med fokus på sikkerhed
- m0n0wall - <http://www.m0n0.ch> gratis firewall baseret på FreeBSD

# Specielle features

- Network Address Translation - NAT
- IPv6 funktionalitet
- Båndbredde håndtering
- VLAN funktionalitet - mere udbredt i forbindelse med VoIP
- Redundante firewalls - pfsync og CARP
- IPsec og Andre VPN features

# Proxy servers

Filtrering på højere niveauer i OSI modellen er muligt

Idag findes proxy applikationer til de mest almindelige funktioner

Den typiske proxy er en caching webproxy der kan foretage HTTP request på vegne af arbejdsstationer og gemme resultatet

NB: nogle protokoller egner sig ikke til proxy servere

SSL forbindelser til *secure websites* kan per design ikke proxies

# IPsec og Andre VPN features

De fleste firewalls giver mulighed for at lave krypterede tunneler

Nyttigt til fjernkontorer der skal have usikker trafik henover usikre netværk som Internet  
Konceptet kaldes Virtual Private Network VPN

IPsec er de facto standarden for VPN og beskrevet i RFC'er

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

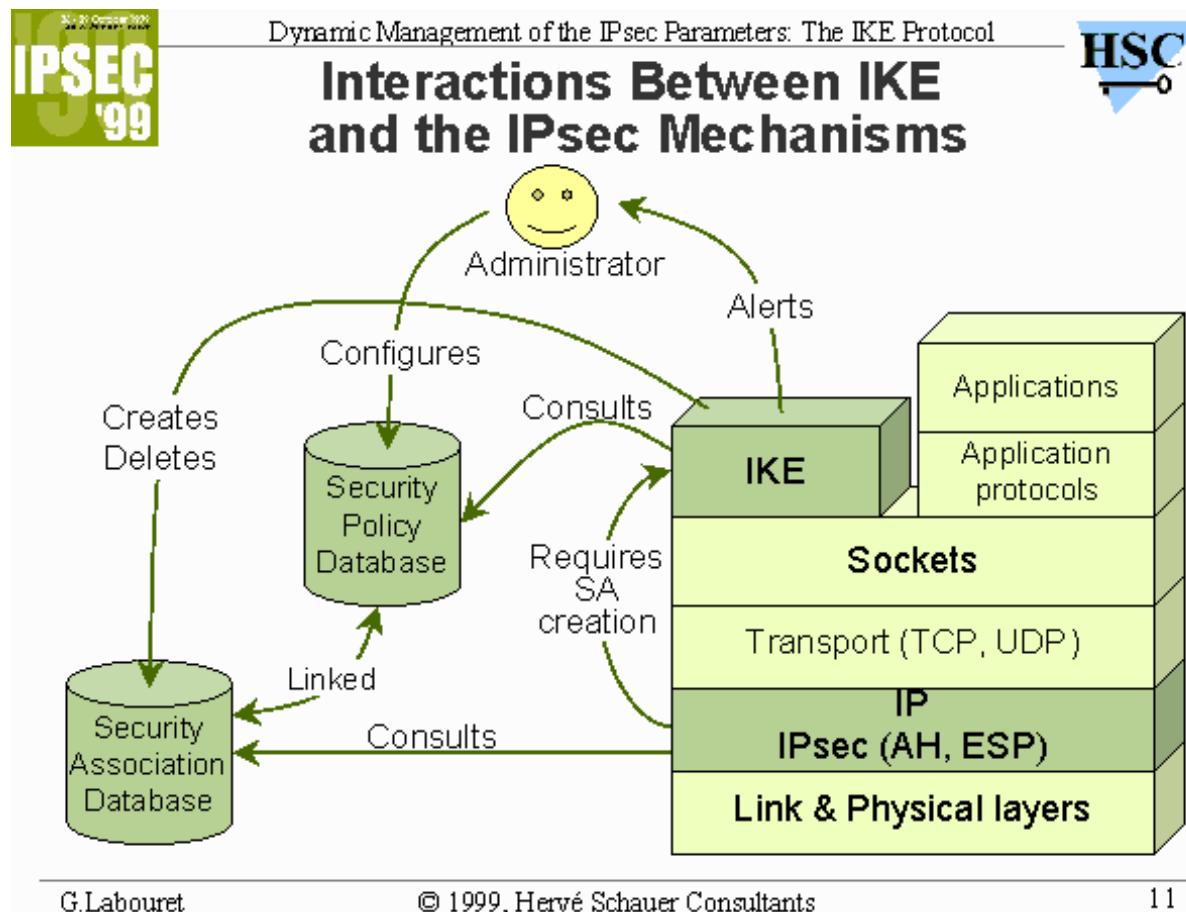
**MANDATORY** i IPv6! - et krav hvis man implementerer fuld IPv6 support

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!



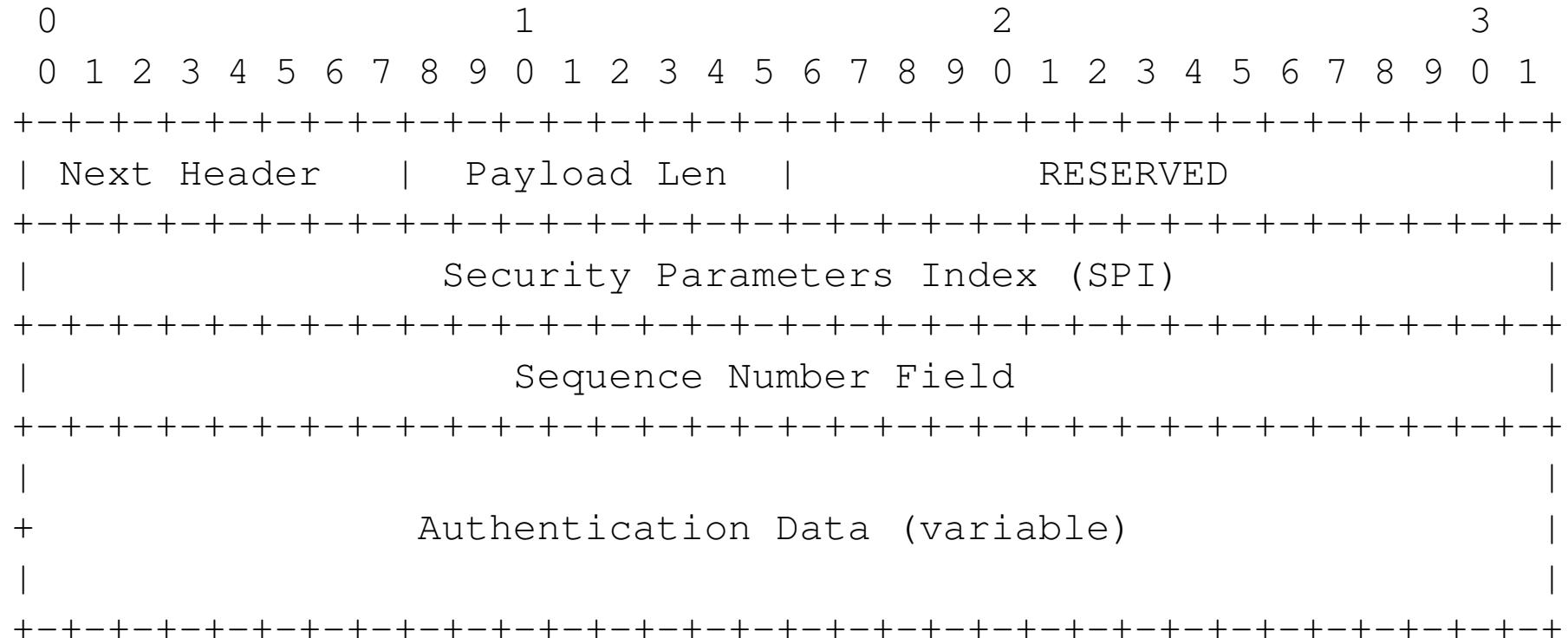
G.Labouret

© 1999, Hervé Schauer Consultants

11

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	hdtrs			
	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

Security  
.net

Der er følgende filer tilgængelige

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

# IPsec setup

Client: Mac OS X/NetBSD/FreeBSD - samme syntaks

rc.ipsec.client

Server: OpenBSD - bruger ipsecadm kommando

rc.ipsec.server

Øvelse til læseren: lav samme i Cisco IOS

Det vil ofte være relevant at se på IOS og IPsec i laboratoriet

Dette setup når vi ikke at demonstrere

# rc.ipsec.client - client setup - adresser

```
#!/bin/sh
# /etc/rc.ipsec.client - IPsec client configuration
# built from http://rt.fm/~jcs/ipsec_wep.phtml
# FreeBSD/NetBSD syntaks! - used on Mac OS X
# IPv4
SEC SERVER=10.0.42.1
SEC CLIENT=10.0.42.53
# IPv6
#SEC SERVER=2001:618:433:101::1
#SEC CLIENT=2001:618:433:101::153
ESPKEY='cat ipsec.esp.key'
AHKEY='cat ipsec.ah.key'

# Flush IPsec SAs in case we get called more than once
setkey -F
setkey -F -P
```

# rc.ipsec.client - client setup - SAs

```
# Establish Security Associations
# 1000 is from the server to the client
# 1001 is from the client to the server
setkey -c <<EOF

add $SEC SERVER $SEC CLIENT esp 0x1000 \
-m tunnel -E blowfish-cbc 0x$ESPKEY -A hmac-sha1 0x$AHKEY;

add $SEC CLIENT $SEC SERVER esp 0x1001 \
-m tunnel -E blowfish-cbc 0x$ESPKEY -A hmac-sha1 0x$AHKEY;

spdadd $SEC CLIENT $SEC SERVER any -P out \
ipsec esp/tunnel/$SEC CLIENT-$SEC SERVER/default;

spdadd $SEC SERVER $SEC CLIENT any -P in \
ipsec esp/tunnel/$SEC SERVER-$SEC CLIENT/default;
EOF
```

# rc.ipsec.server - server setup - adresser

```
#!/bin/sh
#
# Henrik Lund Kramshøj
# /etc/rc.ipsec - IPsec server configuration
# built from http://rt.fm/~jcs/ipsec_wep.phtml
# OpenBSD syntaks!
SEC SERVER=10.0.42.1
SEC CLIENT=10.0.42.53
#SEC SERVER6=2001:618:433:101::1
#SEC CLIENT6=2001:618:433:101::153

ESPKEY='cat ipsec.esp.key'
AHKEY='cat ipsec.ah.key'

# Flush IPsec SAs in case we get called more than once
ipsecadm flush
```

## rc.ipsec.server - server setup - SAs

```
# Establish Security Associations
#
# 1000 is from the server to the client
ipsecadm new esp -spi 1000 -src $SEC SERVER -dst $SEC CLIENT \
-forcetunnel -enc blf -key $ESPKEY \
-auth sha1 -authkey $AHKEY

# 1001 is from the client to the server
ipsecadm new esp -spi 1001 -src $SEC CLIENT -dst $SEC SERVER \
-forcetunnel -enc blf -key $ESPKEY \
-auth sha1 -authkey $AHKEY
```

## rc.ipsec.server - server setup - flows

```
# Create flows
#
# Data going from the outside to the client
ipsecadm flow -out -src $SEC SERVER -dst $SEC CLIENT -proto esp \
-addr 0.0.0.0 0.0.0.0 $SEC CLIENT 255.255.255.255 -dontacq
# IPv6
#ipsecadm flow -out -src $SEC SERVER -dst $SEC CLIENT -proto esp \
#-addr :: :: $SEC CLIENT ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff -dontacq

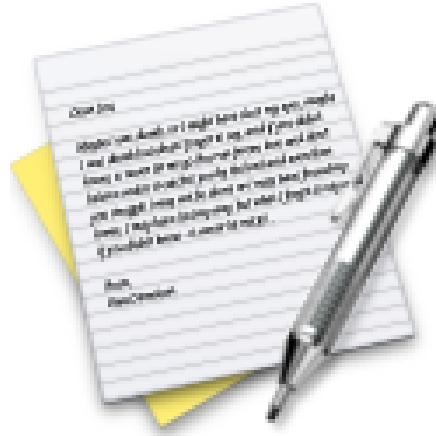
# Data going from the client to the outside
ipsecadm flow -in -src $SEC SERVER -dst $SEC CLIENT -proto esp \
-addr $SEC CLIENT 255.255.255.255 0.0.0.0 0.0.0.0 -dontacq
# IPv6
#ipsecadm flow -in -src $SEC SERVER -dst $SEC CLIENT -proto esp \
#-addr :: :: $SEC CLIENT ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff -dontacq
```

OpenVPN is a full-featured SSL VPN solution which can accommodate a wide range of configurations, including remote access, site-to-site VPNs, WiFi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls (articles) (examples) (security overview) (non-english languages).

Et andet populært VPN produkt er OpenVPN

Bemærk dog at hvis der benyttes TCP i TCP risikerer man at støde ind i et problem som kaldes *TCP in TCP meltdown*

Kilde: <http://openvpn.net/>



Vi laver nu øvelsen

## Firewallkonfiguration

som er øvelse **35** fra øvelseshæftet.

# Portscan, TCP, UDP og ICMP

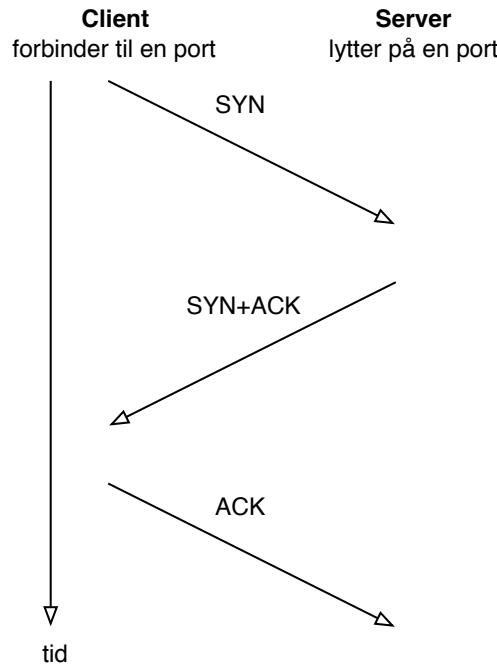
Forskellen mellem TCP og UDP i forbindelse med portscan, og effekten af en firewall der dropper pakker

# Basal Portscanning

Hvad er portscanning  
afprøvning af alle porte fra 0/1 og op til 65535  
målet er at identificere åbne porte - sårbare services  
typisk TCP og UDP scanning  
TCP scanning er ofte mere pålidelig end UDP scanning

TCP handshake er nemmere at identificere  
UDP applikationer svarer forskelligt - hvis overhovedet

# TCP three way handshake



- **TCP SYN half-open** scans
- Tidligere loggede systemer kun når der var etableret en fuld TCP forbindelse - dette kan/kunne udnyttes til *stealth*-scans
- Hvis en maskine modtager mange SYN pakker kan dette fylde tabellen over connections op - og derved afholde nye forbindelser fra at blive oprette - **SYN-flooding**

# Ping og port sweep

scanninger på tværs af netværk kaldes for sweeps

Scan et netværk efter aktive systemer med PING

Scan et netværk efter systemer med en bestemt port åben

Er som regel nemt at opdage:

- konfigurerer en maskine med to IP-adresser som ikke er i brug
- hvis der kommer trafik til den ene eller anden er det portscan
- hvis der kommer trafik til begge IP-adresser er der nok foretaget et sweep - bedre hvis de to adresser ligger et stykke fra hinanden

# nmap port sweep after port 80/TCP

Port 80 TCP er webservere

```
# nmap -p 80 217.157.20.130/28
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on router.kramse.dk (217.157.20.129):
Port      State       Service
80/tcp    filtered   http
```

```
Interesting ports on www.kramse.dk (217.157.20.131):
Port      State       Service
80/tcp    open        http
```

```
Interesting ports on  (217.157.20.139):
Port      State       Service
80/tcp    open        http
```

# nmap port sweep after port 161/UDP

Port 161 UDP er SNMP

```
# nmap -sU -p 161 217.157.20.130/28
```

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on router.kramse.dk (217.157.20.129):
Port      State       Service
161/udp   open        snmp
```

The 1 scanned port on mail.kramse.dk (217.157.20.130) is: closed

```
Interesting ports on www.kramse.dk (217.157.20.131):
Port      State       Service
161/udp   open        snmp
```

The 1 scanned port on (217.157.20.132) is: closed

# OS detection

```
# nmap -O ip.adresse.slet.tet scan af en gateway
Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2003-12-03 11:31 CET
Interesting ports on gw-int.security6.net (ip.adresse.slet.tet):
(The 1653 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
1080/tcp  open  socks
5000/tcp  open  UPnP
Device type: general purpose
Running: FreeBSD 4.X
OS details: FreeBSD 4.8-STABLE
Uptime 21.178 days (since Wed Nov 12 07:14:49 2003)
Nmap run completed -- 1 IP address (1 host up) scanned in 7.540 seconds
```

- lavniveau måde at identificere operativsystemer på
- send pakker med *anderledes* indhold
- Reference: *ICMP Usage In Scanning* Version 3.0, Ofir Arkin  
<http://www.sys-security.com/html/projects/icmp.html>

# Top 75 Security Tools



listen over 75 top security tools - nogle værktøjer springes over, nogle har vi brugt

Den er samlet af Fyodor og findes på:

<http://www.insecure.org/tools.html>

# Hvad skal der ske?

Tænk som en hacker

Rekognoscering

- ping sweep, port scan
- OS detection - TCP/IP eller banner grab
- Servicescan - rpcinfo, netbios, ...
- telnet/netcat interaktion med services

Udnyttelse/afprøvning: Nessus, nikto, exploit programs

Oprydning vises ikke på kurset, men I bør i praksis:

- Lav en rapport
- Gennemgå rapporten, registrer ændringer
- Opdater programmer, konfigurationer, arkitektur, osv.

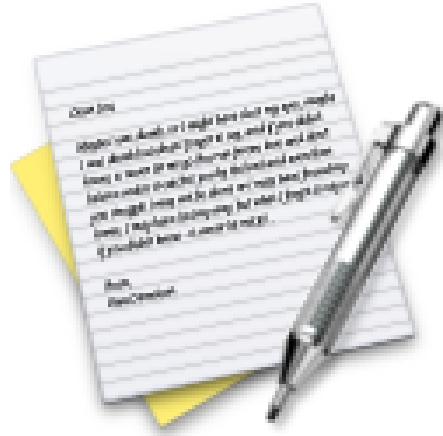
I skal jo også VISE andre at I gør noget ved sikkerheden.



Vi laver nu øvelsen

## Find maskiner

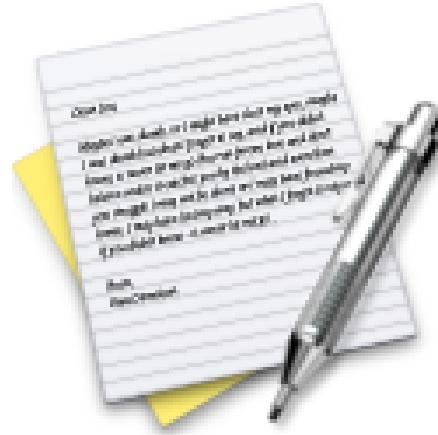
som er øvelse **36** fra øvelseshæftet.



## Vi laver nu øvelsen

# nmap portscanning

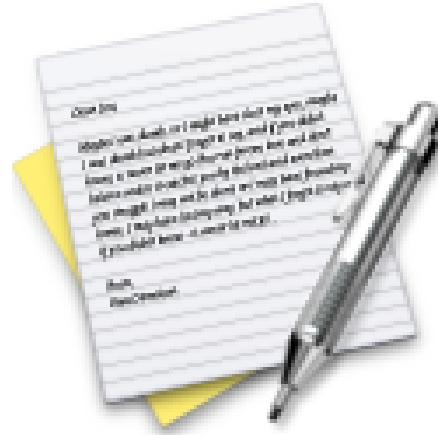
som er øvelse 37 fra øvelseshæftet.



Vi laver nu øvelsen

## nmap servicescanning

som er øvelse **38** fra øvelseshæftet.



Vi laver nu øvelsen

## nmap OS detection

som er øvelse **39** fra øvelseshæftet.

## Firewalls og IPv6

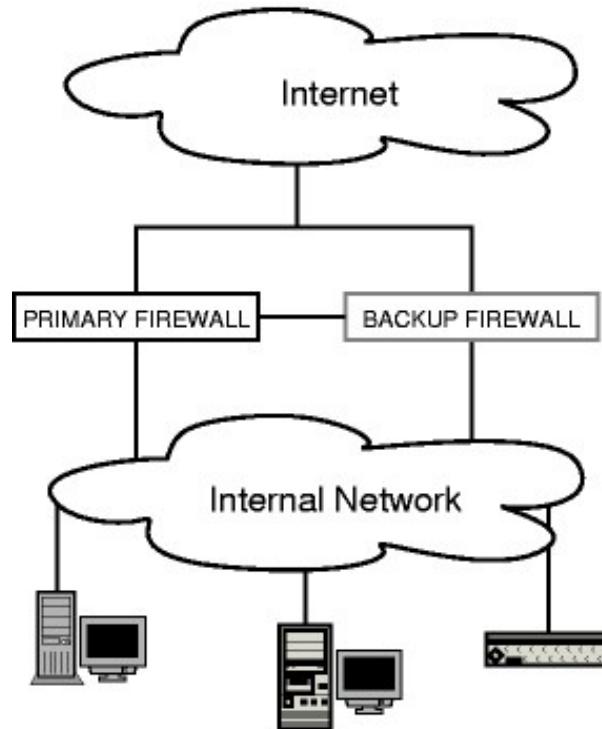
Læg mærke til forskellen mellem ARP og ICMPv6

Hvis det er muligt lav een regel der tillader adgang til services uanset protokol

NB: husk at aktivere IP forwarding når I skal lave en firewall

```
# Macros: define common values, so they can be referenced and changed easily.
int_if=vr0
ext_if=vr2
tunnel_if=gif0
table <homenet6> 2001:16d8:ffd2:cf0f::/64
set skip on lo0
scrub in all
# Filtering: the implicit first two rules are
block in all
block out all
# allow ICMPv6 for NDP
pass in inet6 proto ipv6-icmp all icmp6-type neighboradv keep state
# server with configured IP address and router advertisement daemon running
pass out inet6 proto ipv6-icmp all icmp6-type routersol keep state
# client which uses autoconfiguration would use this instead
#pass in inet6 proto ipv6-icmp all icmp6-type routeradv keep state
#pass out inet6 proto ipv6-icmp all icmp6-type neighborbsol keep state
table <sixxspop> 82.96.56.14 2001:16d8:ff00:155::1
pass in on $ext_if proto icmp from <sixxspop6> to ($ext_if)
pass in on $tunnel_if proto icmp6 from <sixxspop6> to any
pass in on $int_if all
pass out on $int_if all keep state
... probably not working AS IS
```

# Redundante firewalls



- OpenBSD Common Address Redundancy Protocol CARP - både IPv4 og IPv6 overtagelse af adresse både IPv4 og IPv6
- pfSync - sender opdateringer om firewall states mellem de to systemer
- Kilde: <http://www.countersiege.com/doc/pfSync-carp/>

# Redundante forbindelser hardware

```
root@azumi:# cat hostname.fxp0
up
root@azumi:# cat hostname.fxp1
up
root@azumi:# cat /etc/hostname.trunk0
trunkproto failover trunkport fxp0 trunkport fxp1
dhcpc
```

OpenBSD trunk interface

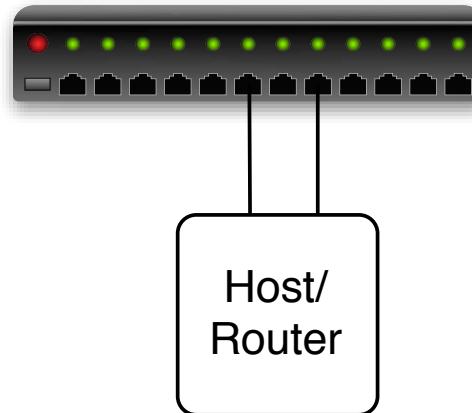
Linux bonding,

Etherchannel Cisco

Idag anbefales IEEE 802.3ad LACP som er en åben standard

<http://en.wikipedia.org/wiki/EtherChannel>

# LACP Link Aggregation Control Protocol



IEEE 802.3ad standardiseret bundling/failover

Målet er at give:

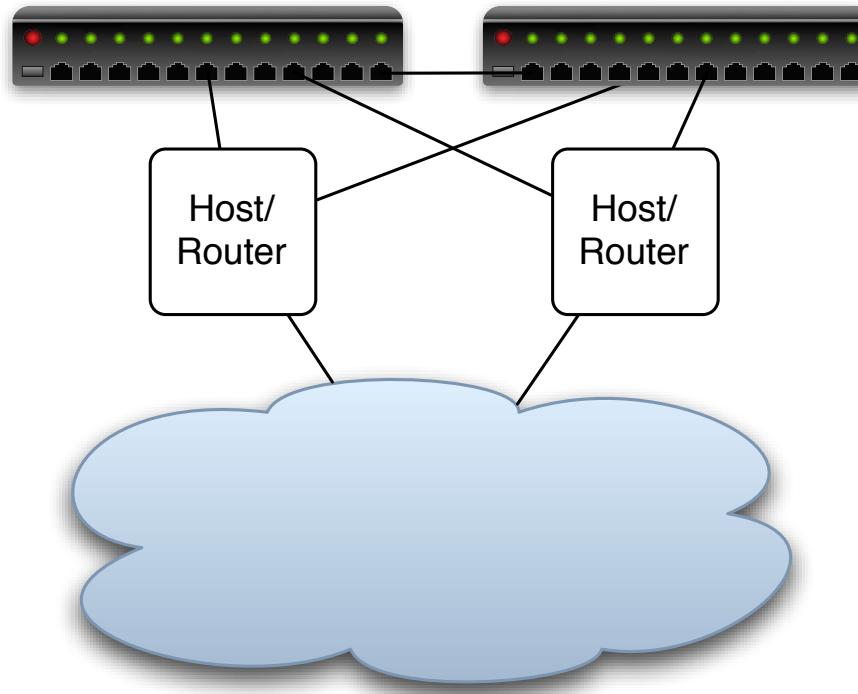
- mere båndbrede end en enkelt port
- failover - hvis et link falder ud

En server med to netinterfaces kan med fordel forbindes til to porte

Er ikke generelt understøttet i alle operativsystemer, men det kommer

[http://en.wikipedia.org/wiki/Link\\_Aggregation\\_Protocol](http://en.wikipedia.org/wiki/Link_Aggregation_Protocol)

# Redundante forbindelser IP-niveau



HSRP Hot Standby Router Protocol, Cisco protokol, RFC-2281

VRRP Virtual Router Redundancy Protocol, IETF RFC-3768, åben standard - ikke fri

CARP Common Address Redundancy Protocol, findes på OpenBSD og FreeBSD

[http://en.wikipedia.org/wiki/Common\\_Address\\_Redundancy\\_Protocol](http://en.wikipedia.org/wiki/Common_Address_Redundancy_Protocol)

## Mobile IP

Mobility er ved at blive et krav, idet enheder idag er mobile  
Specielt ønsker vi at håndholdte computere og laptops kan modtage data  
Tidligere skiftede man blot adresse undervejs

Idag ønsker man at enheden kan kontaktes nemmere, selv udenfor *huset*  
RFC-3344 IP Mobility Support for IPv4

RFC-4721 Mobile IPv4 Challenge/Response Extensions (Revised)

RFC-3775

[http://en.wikipedia.org/wiki/Mobile\\_IP](http://en.wikipedia.org/wiki/Mobile_IP)

Bemærk at Mobile IP ikke altid er nødvendig eller benyttes, mange protokoller som eksempelvis POP3/IMAP virker fint ved at enheden kalder tilbage til serveren

# Mobile IP begreber

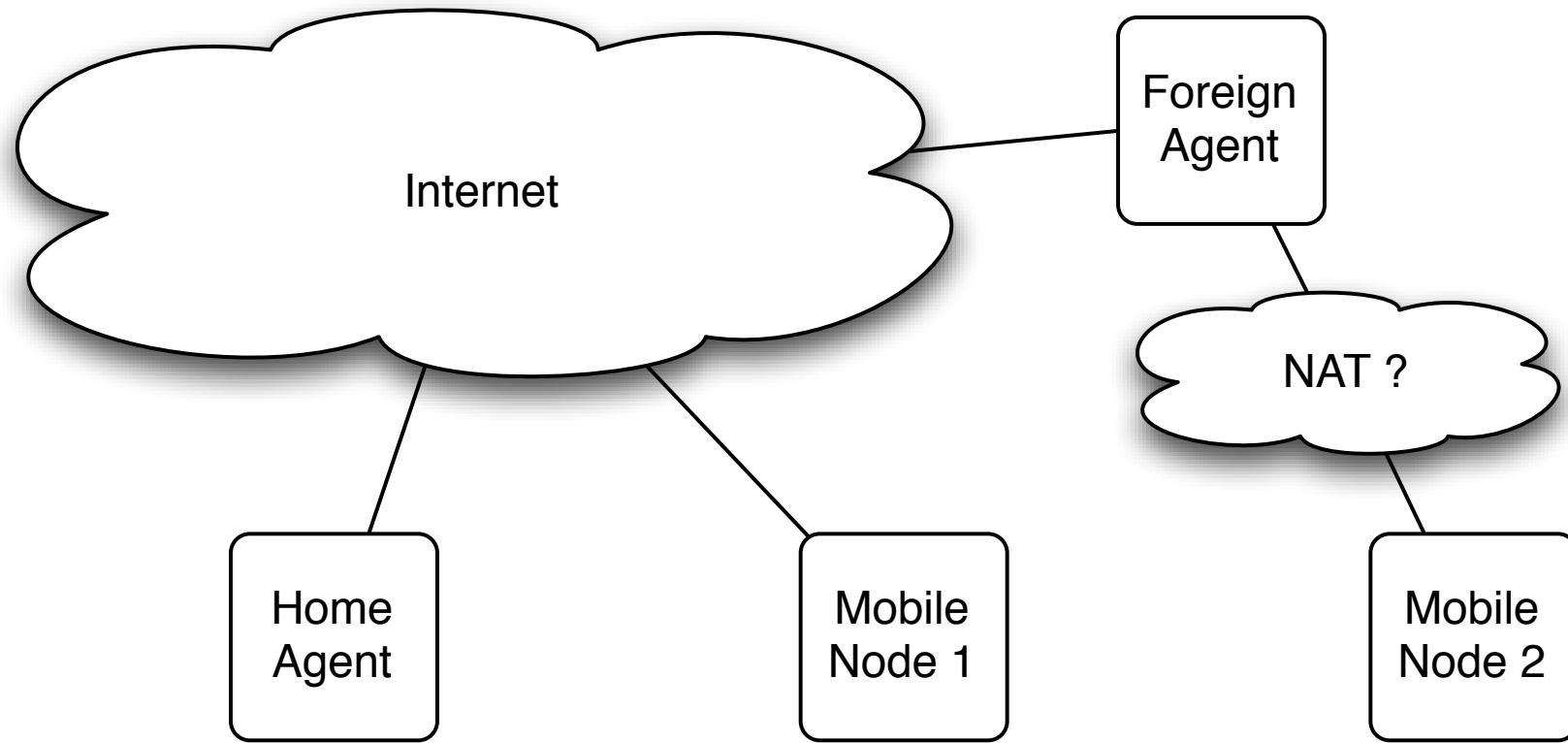
## Definitioner - fra RFC-3344:

- Mobile Node A host or router that changes its point of attachment from one network or subnetwork to another.
- Home Agent A router on a mobile node's home network which tunnels datagrams for delivery to the mobile node when it is away from home, and maintains current location information for the mobile node.
- Foreign Agent A router on a mobile node's visited network which provides routing services to the mobile node while registered. The foreign agent detunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent. For datagrams sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes.

## Selve funktionen:

A mobile node is given a long-term IP address on a home network. This home address is administered in the same way as a "permanent" IP address is provided to a stationary host. When away from its home network, a "care-of address" is associated with the mobile node and reflects the mobile node's current point of attachment.

# Oversigt Mobile IP



Se også Mobile IPv6 A short introduction <http://www.hznet.de/ipv6/m.ipv6-intro.pdf>

Tidligere havde vi adskilte netværk, nu samles de  
Idag er det meget normalt at både firmaer og private bruger IP-telefoni  
Fordele er primært billigere og mere fleksibelt

Eksempler på IP telefoni:

- Skype benytter IP, men egenudviklet protokol
- Cisco IP-telefoner benyttes ofte i firmaer
- Cybercity telefoni kører over IP, med analog adapter

Det anbefales at se på Asterisk telefoniserver, hvis man har mod på det :-)

<http://www.asterisk.org/>

# VoIP bekymringer

Der er generelt problemer med:

- Stabilitet - quality of service, netværket skal være bygget til det
- Sikkerhed - hvem lytter med, hvem kan afbryde forbindelsen  
Se evt. <http://www.voipsa.org/>
- Spam over VoIP, connect, send WAV fil med spam kaldes SPIT
- Kompatibilitet - hvilke protokoller, codecs, standarder, ...

Der er flere store spillere

# VoIP protokoller

SIP Session Initiation Protocol, IETF standard signaleringsprotokol

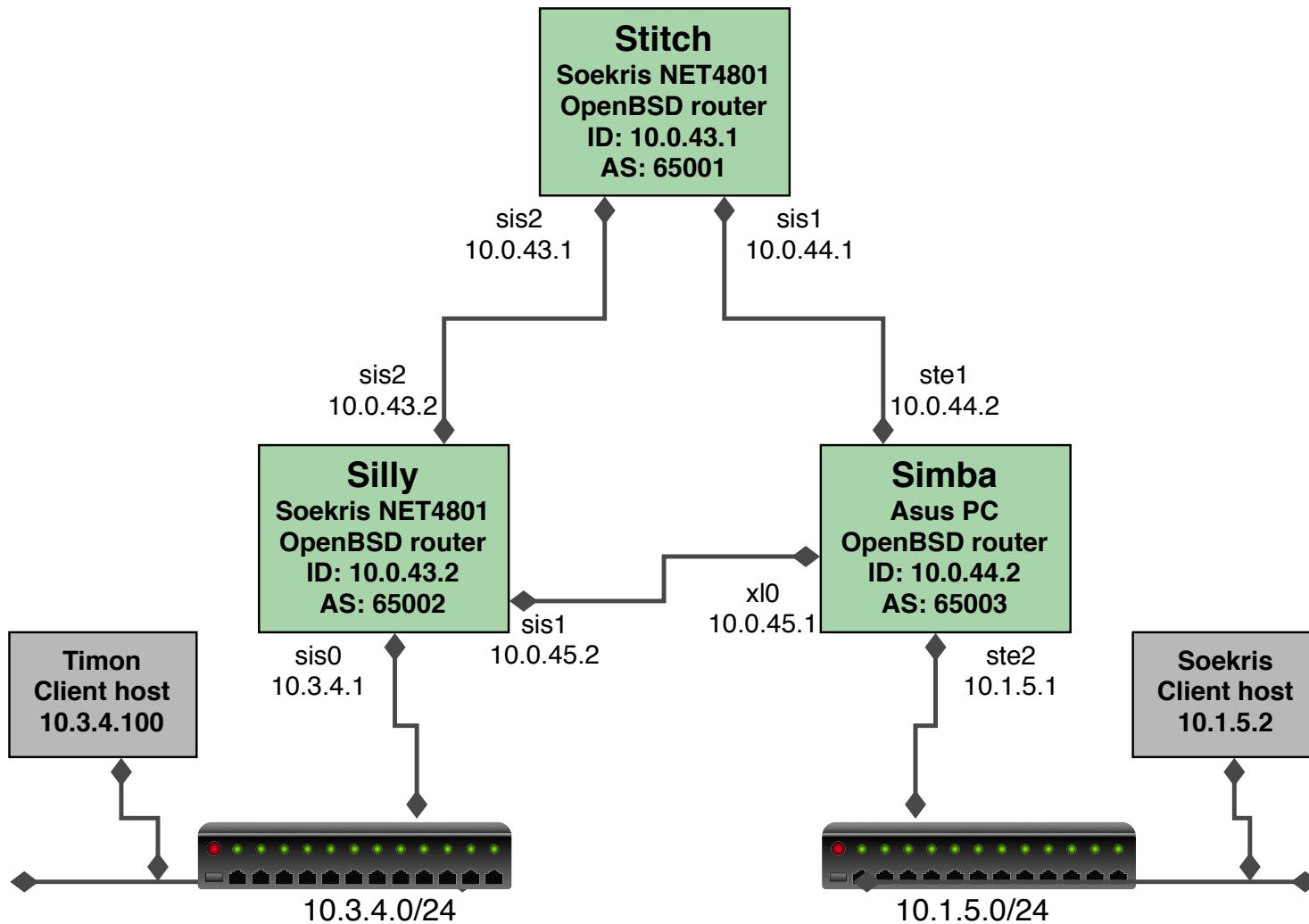
H.323 ITU-T standard signaleringsprotokol

IAX Inter-Asterisk Exchange Protocol, Asterisk protokol

SSCP Cisco protokol

ZRTP Phil Zimmermann, zfone - sikker kommunikation

<http://zfoneproject.com/>

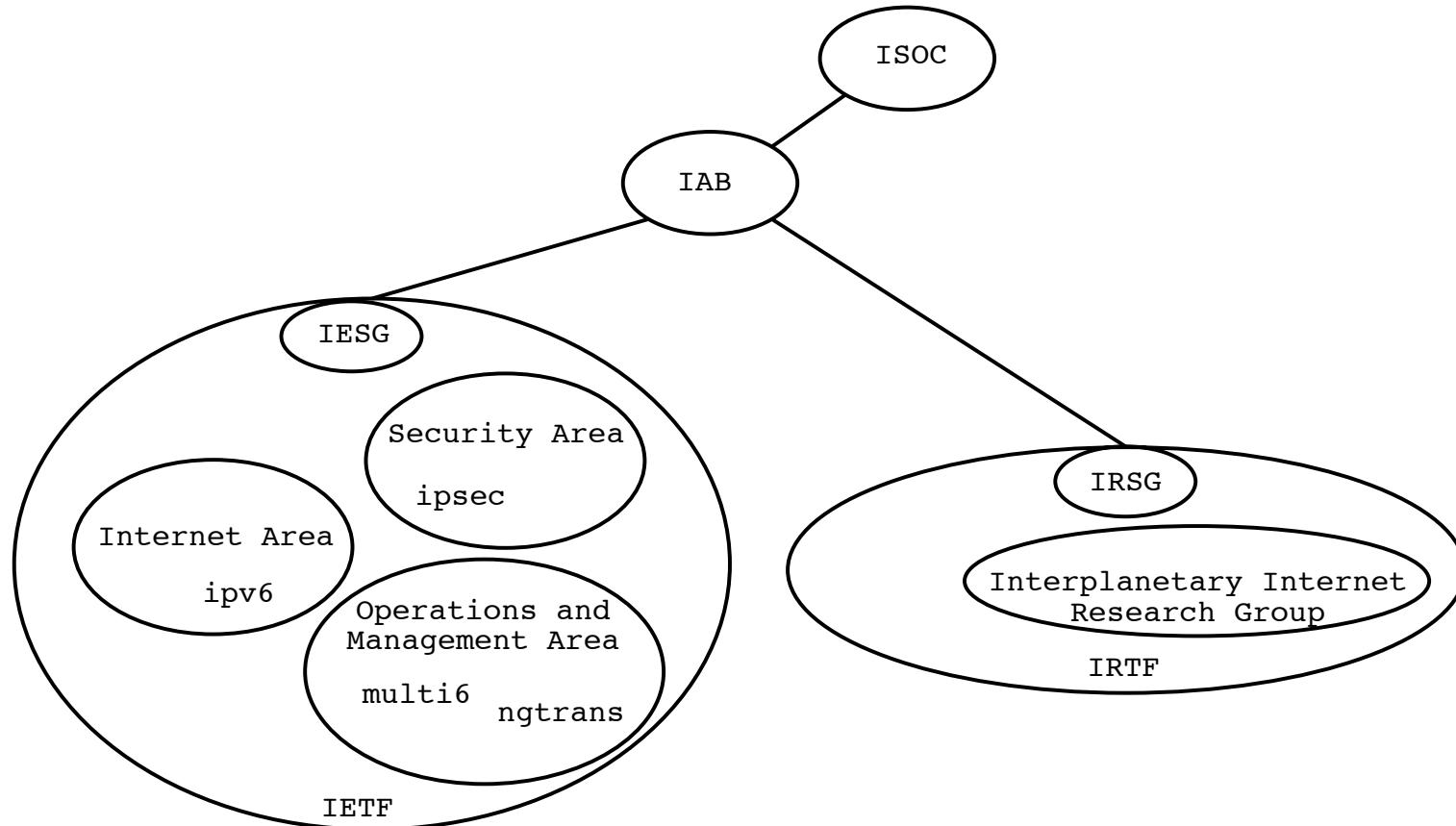


# Opsamling

Dagen idag er primært beregnet til opsamling

Detaljer som ikke har været gennemgået undervejs, fordi jeg mente det var bedre at skærme imod i den første gennemgang

# Internet-relaterede organisationer



Oftest er man interesseret i <http://www.ietf.org/>

# Proxy-arp

Routere understøtter ofte Proxy ARP

Med Proxy ARP svarer de for en adresse bagved routeren

Derved kan man få trafik nemt igennem fra internet til adresser

Det er smart i visse situationer hvor en subnetting vil spilde for mange adresser

Hvis man kun har få adresser er subnetting måske heller ikke muligt

[http://en.wikipedia.org/wiki/Proxy\\_ARP](http://en.wikipedia.org/wiki/Proxy_ARP)

## Reverse ARP

Tidligere brugte man en protokol kaldet Reverse ARP til at uddele IP-adresser  
Med Reverse ARP sender en enhed et request og får et Reverse ARP svar tilbage  
*Jeg har denne MAC adresse, hvad er min IP?*

*Hvis du er denne MAC adresse er din IP 10.2.3.1*

Det benyttes meget sjældent idag, men var tidligere brugt til netboot af arbejdsstationer m.v.

## ICMP redirect

Routere understøtter ofte ICMP Redirect

Med ICMP Redirect kan man til en afsender fortælle en anden vej til destination

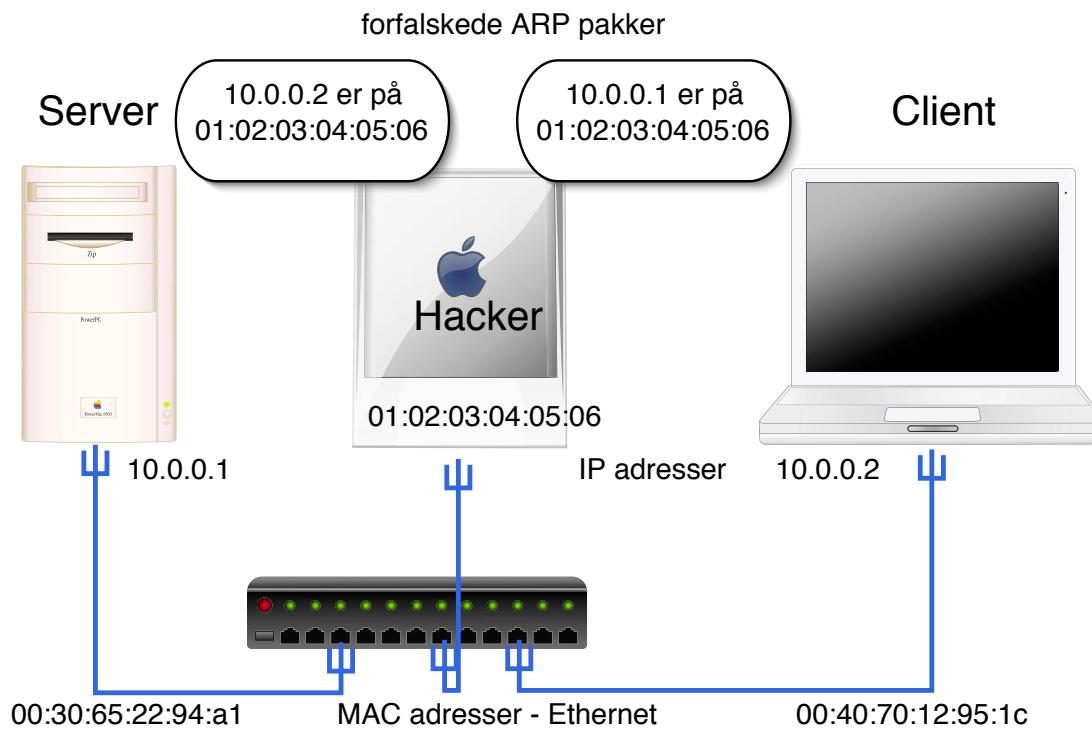
Den angivne vej kan være smartere eller mere effektiv

Det er desværre uheldigt, idet der ingen sikkerhed er

Idag bør man ikke lytte til ICMP redirects, ej heller generere dem

Det svarer til ARP spoofing, idet trafik omdiriges

# Hvordan virker ARP spoofing?



Hackeren sender forfalskede ARP pakker til de to parter

De sender derefter pakkerne ud på Ethernet med hackerens MAC adresse som modtager - han får alle pakkerne

# Forsvar mod ARP spoofing

Hvad kan man gøre?

Iåse MAC adresser til porte på switche

Iåse MAC adresser til bestemte IP adresser

Efterfølgende administration!

**arpwatch er et godt bud** - overvåger ARP

bruge protokoller som ikke er sårbare overfor opsamling

Der er defineret Multicast protokoller på internet

Med multicast kan man sende data til en nærmere angivet gruppe

Multicast er tiltænkt ting som radio og video broadcast

IPv6 benytter en del multicast adresser, all-nodes, all-routes, ...

Hvem der modtager data styres så ved hjælp af IGMP

IGMP bruges således til at styre hvem der på et givet tidspunkt er med i IP multicast grupper

RFC-3376 Internet Group Management Protocol, Version 3

[http://en.wikipedia.org/wiki/Internet\\_Group\\_Management\\_Protocol](http://en.wikipedia.org/wiki/Internet_Group_Management_Protocol)

# TCP sequence number prediction



tidligere baserede man ofte login og adgange på de IP adresser som folk kom fra  
det er ikke pålideligt at tro på address based authentication

TCP sequence number kan måske gættes

Mest kendt er nok Shimomura der blev hacket på den måde, måske af Kevin D Mitnick  
eller en kompagnon

I praksis vil det være svært at udføre på moderne operativsystemer

Se evt. <http://www.takedown.com/>

(filmen er ikke så god ;-)

# Hardware IPv4 checksum offloading

IPv4 checksum skal beregnes hvergang man modtager en pakke

IPv4 checksum skal beregnes hvergang man sender en pakke

Lad en ASIC gøre arbejdet!

De fleste servernetkort tilbyder at foretage denne beregning på IPv4

IPv6 benytter ikke header checksum, det er unødvendigt

NB: kan resultere i at tcpdump siger checksum er forkert!

## RFC-2142 Mailbox Names for Common Services, Roles and Functions

Du BØR konfigurere dit domæne til at modtage post for følgende adresser:

- postmaster@domæne.dk
- abuse@domæne.dk
- webmaster@domæne.dk, evt. www@domæne.dk

Du gør det nemmere at rapportere problemer med dit netværk og services

# E-mail best current practice

MAILBOX	AREA	USAGE
ABUSE	Customer Relations	Inappropriate public behaviour
NOC	Network Operations	Network infrastructure
SECURITY	Network Security	Security bulletins or queries
...		
MAILBOX	SERVICE	SPECIFICATIONS
POSTMASTER	SMTP	[RFC821], [RFC822]
HOSTMASTER	DNS	[RFC1033-RFC1035]
USENET	NNTP	[RFC977]
NEWS	NNTP	Synonym for USENET
WEBMASTER	HTTP	[RFC 2068]
WWW	HTTP	Synonym for WEBMASTER
UUCP	UUCP	[RFC976]
FTP	FTP	[RFC959]

Kilde: RFC-2142 Mailbox Names for Common Services, Roles and Functions. D. Crocker. May 1997

# Brug krypterede forbindelser

```
[root@hlk hlk]# dsniff
dsniff: listening on fxp0
-----
05/20/03 08:53:38 tcp client.49154 -> server.110 (pop)
USER hlk
PASS secr3t! Her er opsamlet et kodeord til e-mail

-----
05/20/03 08:54:11 tcp client.49155 -> server.23 (telnet)
[poppe]

hlk Her er opsamlet kodeord og  
kommandoer fra en session
secr3t!
ls
exit

-----
05/20/03 08:55:33 tcp client.49156 -> server.23 (telnet)
[poppe]

anja
anjnaanja
anja
```

Især på utroværdige netværk kan det give problemer at benytte sårbare protokoller

## Mission 1: Kommunikere sikkert

Du må ikke bruge ukrypterede forbindelser til at administrere UNIX

Du må ikke sende kodeord i ukrypterede e-mail beskeder

**Telnet daemonen - telnetd må og skal dø!**

## Mission 1: Kommunikere sikkert

Security

.net

Du må ikke bruge ukrypterede forbindelser til at administrere UNIX

Du må ikke sende kodeord i ukrypterede e-mail beskeder

**Telnet daemonen - telnetd må og skal dø!**

**FTP daemonen - ftpd må og skal dø!**

# Mission 1: Kommunikere sikkert

Du må ikke bruge ukrypterede forbindelser til at administrere UNIX

Du må ikke sende kodeord i ukrypterede e-mail beskeder

**Telnet daemonen - telnetd må og skal dø!**

**FTP daemonen - ftpd må og skal dø!**

**POP3 daemonen port 110 må og skal dø!**

## Mission 1: Kommunikere sikkert

Security

Du må ikke bruge ukrypterede forbindelser til at administrere UNIX

Du må ikke sende kodeord i ukrypterede e-mail beskeder

**Telnet daemonen - telnetd må og skal dø!**

**FTP daemonen - ftpd må og skal dø!**

**POP3 daemonen port 110 må og skal dø!**

**IMAPD daemonen port 143 må og skal dø!**

## Mission 1: Kommunikere sikkert

Du må ikke bruge ukrypterede forbindelser til at administrere UNIX

Du må ikke sende kodeord i ukrypterede e-mail beskeder

**Telnet daemonen - telnetd må og skal dø!**

**FTP daemonen - ftpd må og skal dø!**

**POP3 daemonen port 110 må og skal dø!**

**IMAPD daemonen port 143 må og skal dø!**

**væk med alle de ukrypterede forbindelser!**

Vi vil nu gennemgå netværksdesign med udgangspunkt i vores setup

Vores setup indeholder:

- Routere
- Firewall
- Wireless
- DMZ
- DHCPD, BIND, BGPD, OSPFD, ...

Den kunne udvides med flere andre teknologier vi har til rådighed:

- VLAN inkl VLAN trunking/distribution
- WPA Enterprise

Hvad taler for og imod - de næste slides gennemgår nogle standardsetups

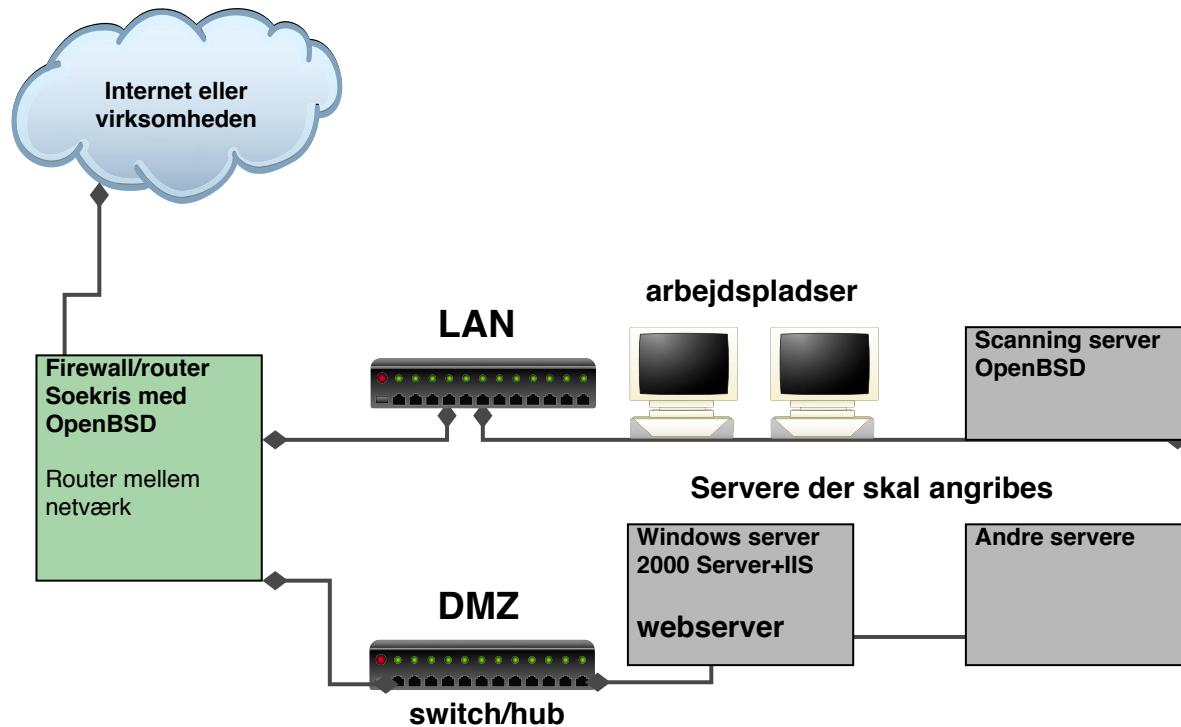
En slags Patterns for networking

# Netværksdesign og sikkerhed

Hvad kan man gøre for at få bedre netværkssikkerhed?

- Bruge switcher - der skal ARP spoofes og bedre performance
- Opdele med firewall til flere DMZ zoner for at holde utsatte servere adskilt fra hinanden, det interne netværk og Internet
- Overvåge, læse logs og reagere på hændelser

Husk du skal også kunne opdatere dine servere



Du bør opdele dit netværk i segmenter efter traffik

Du bør altid holde interne og eksterne systemer adskilt!

Du bør isolere farlige services i jails og chroots

# Intrusion Detection Systems - IDS

angrebsværktøjerne efterlader spor

hostbased IDS - kører lokalt på et system og forsøger at detektere om der er en angriber inde

network based IDS - NIDS - bruger netværket

Automatiserer netværksovervågning:

- bestemte pakker kan opfattes som en signatur
- analyse af netværkstrafik - FØR angreb
- analyse af netværk under angreb - sender en alarm

<http://www.snort.org> - det kan anbefales at se på Snort



Snort er Open Source og derfor godt til undervisning  
man kan se det som et antivirus system til netværket  
forsøger at detektere *angreb*, *skadelig* og *forkert* traffik  
pakker der minder om eksempelvis:

- nmap portscan
- nmap OS detection - med underlige pakker
- fragmenter der overlapper
- shellcode der sendes til systemer som BIND

# Snort regler

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"ICMP Address Mask Reply"; icode:0; itype:18; classtype:misc-activity; sid:386; rev:5;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Address Mask Reply undefined code"; icode:>0; itype:18; classtype:misc-activity; sid:387; rev:7;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Address Mask Request"; icode:0; itype:17; classtype:misc-activity; sid:388; rev:5;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Address Mask Request undefined code"; icode:>0; itype:17; classtype:misc-activity; sid:389; rev:7;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Alternate Host Address"; icode:0; itype:6; classtype:misc-activity; sid:390; rev:5;)
```

- sid - snort rules id - identificerer en signatur
- reference - hvor kommer reglen fra
- icode - ICMP code
- itype - ICMP type
- ... se mere i snort manualen



snort er baseret på signaturer

mange falske alarmer - tuning og vedligehold

hvordan sikrer man sig at man har opdaterede signaturer for angreb som går verden rundt på et døgn

# Planlægning af IDS miljøer

## Før installationen

- Hvad er formålet - reaktion eller "statistik"
- Hvor skal der måles - hele netværket eller specifikke dele
- Hvad skal måles og hvilke operativsystemer og servere/services

## Implementationen

- Er infrastrukturen iorden som den er
- Er der gode målepunkter - monitorporte
- Et målepunkt eller flere
- Hvormeget trafik skal måles

## Selve idriftsættelsen

- Ændringer af infrastrukturen
- Installation af udstyret
- Test af udstyret udenfor drift
- Installation i driftsmiljøet
- Test af udstyret i driftsmiljøet

# Opsætning og konfiguration af IDS miljøer

Vælg en simpel installation til at starte med!

Undgå for alt i verden for meget information

- Start med en enkelt sensor
- Byg en server med database og "brugerværktøjer"
- Start med at overvåge dele af nettet
- Brug et specifikt regelsæt i starten - eksempelvis kun Windows eller kun UNIX
- Lav nogle simple rapporter til at starte med

Gør netværket mere sikkert før du lytter på hele netværket

Brug tcpdump/Ethereal til at se på trafik, lær IP pakker at kende

Brug Snort til at evaluere

- husk at man kan starte med Snort og senere skifte til andre produkter
- Erfaring tæller, Snort tillader at man ser de fine detaljer - motoren

# Vedligehold og overvågning af IDS miljøer

Uden vedligehold er IDS værdiløst - lad hellere være!

- Vedligehold af software på operativsystemet
- Vedligehold af IDS softwaren
- Vedligehold af regelsæt

Overvågning - kører IDS systemet, databaser og sensorer

Statistik og brug af IDS systemet

- Vedligehold af rapporter - hvad er vi interesseret i
- Automatisk rapportgenerering - daglig rapport, rapport pr måned
- Specielle hændelser - hvad skete der onsdag mellem 11-12

Et IDS kan også blot være en ARPwatch

ARPwatch advarer hvis nogen tager adressen fra default gateway

# Honeypots

Man kan uddover IDS installere en honeypot

En honeypot består typisk af:

- Et eller flere sårbare systemer
- Et eller flere systemer der logger traffik til og fra honeypot systemerne

Meningen med en honeypot er at den bliver angrebet og brudt ind i

Hvad muligheder har man

- Ændre miljø
- forbedre systemerne
- undgå standardindstillinger
- vær opdateret på sikkerhedsområdet
- have retningslinier - ens sikkerhedsniveau
- drop kompatibilitet med usikre systemer
- en god infrastruktur
- brug kryptografi
- brug standartbiblioteker
- test af systemer

## Ændre miljø

### Ændre arkitektur sw/hw/netværkstopologi

- blokere porte således at en webserver IKKE kan connecte tilbage til hackeren!
- blokere de services der IKKE skal tilgås udefra
- skifte programmeringssprog

Husk altid at hackeren også kan gå ind ad hovedøren

eksempelvis SAP Internet gateway, hvor man kunne lægge det bagvedliggende system ned med loginrequests

# Forbedre systemerne

## Operativsystemet

- non-executable stack
- non-executable heap

## Applikationsservere

- filtrering af "dårlige" requests e-Eye sikret IIS
- mere "sikker" default opsætning

Jeg tror vi vil se flere implementere den slags løsninger

## Eksempelvis:

- Microsoft IIS web server version 6 er mere sikker i default opsætningen
- Apache HTTPD web server version 2 er mere modulær og nemmere at bygge sikkert

# Undgå standard indstillinger

Giv jer selv mere tid til at patche og opdatere

Tiden der går fra en sårbarhed annonceres på bugtraq til den bliver udnyttet er meget kort idag!

Ved at undgå standard indstillinger kan der måske opnås en lidt længere frist - inden ormene kommer

NB: ingen garanti

## Pattern: erstat Telnet med SSH

Telnet er død!

Brug altid Secure Shell fremfor Telnet

Opgrader firmware til en der kan SSH, eller køb bedre udstyr næste gang

Selv mine små billige Linksys switcher forstår SSH!

## Pattern: erstat FTP med HTTP

Hvis der kun skal distribueres filer kan man ofte benytte HTTP istedet for FTP

Hvis der skal overføres med password er SCP/SFTP fra Secure Shell at foretrække

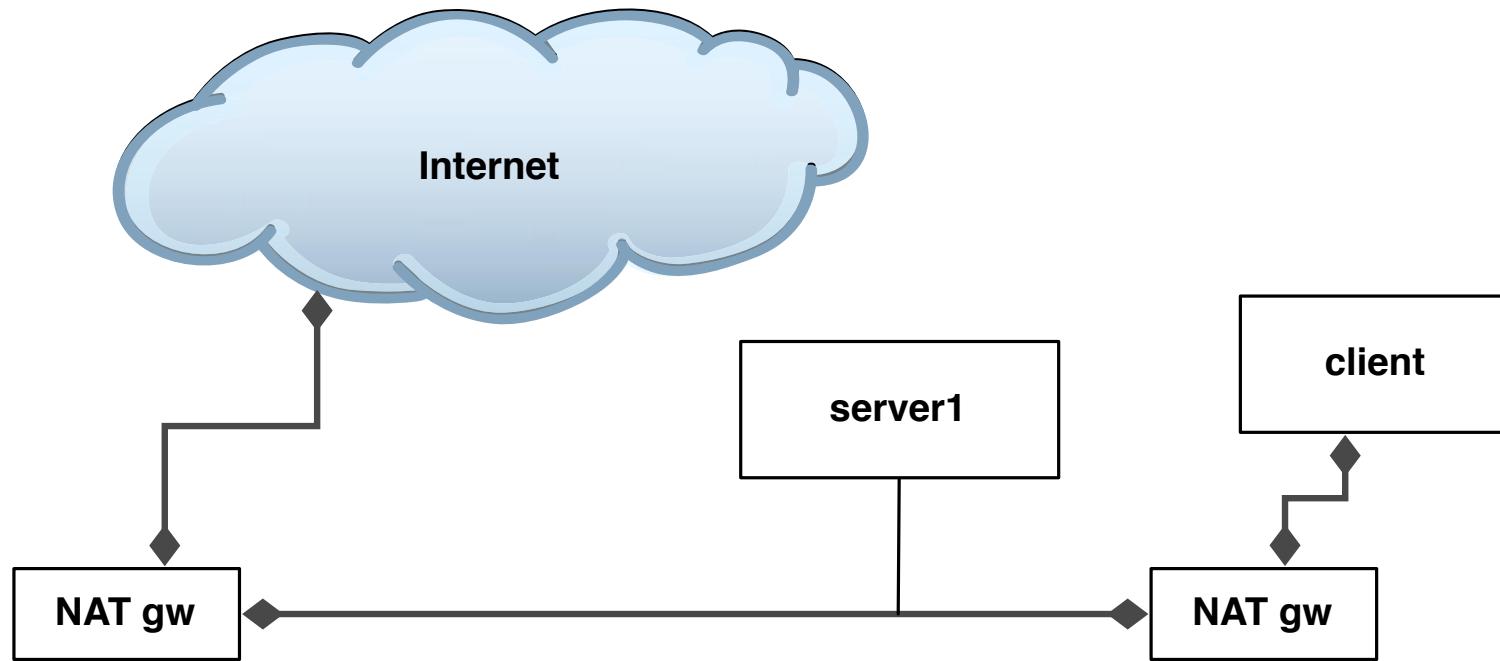
# Anti-patterns

Nu præsenteres et antal setups, som ikke anbefales

Faktisk vil jeg advare mod at bruge dem

Husk følgende slides er min mening

# Anti-pattern dobbelt NAT i eget netværk



Det er nødvendigt med NAT for at oversætte traffik der sendes videre ud på internet.

Der er ingen som helst grund til at benytte NAT indenfor eget netværk!

## Anti-pattern blokering af ALT ICMP

```
ipfw add allow icmp from any to any icmptypes 3,4,11,12
```

Lad være med at blokere for alt ICMP, så ødelægger du funktionaliteten i dit net

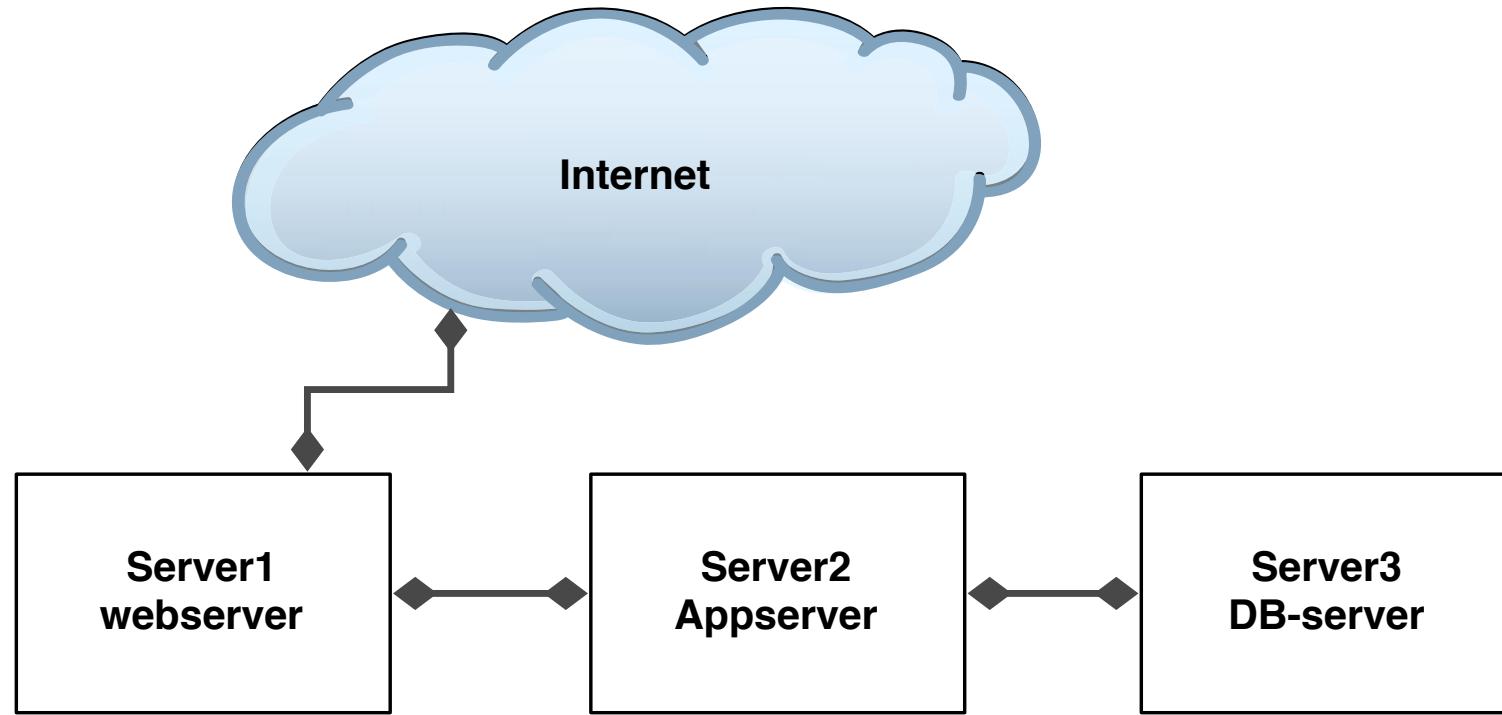
## Anti-pattern blokering af DNS opslag på TCP

Det bliver (er) nødvendigt med DNS opslag over TCP på grund af store svar. Det betyder at firewalls skal tillade DNS opslag via TCP

Guide:

Brug en caching nameserver, således at det kun er den som kan lave DNS opslag ud i verden

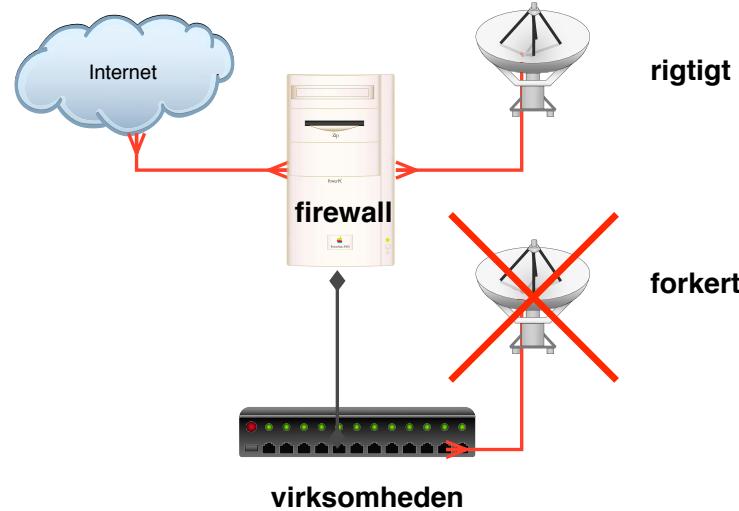
# Anti-pattern daisy-chain



Daisy-chain af servere, erstat med firewall, switch og VLAN

Det giver et væld af problemer med overvågning, administration, backup og opdatering

# Anti-pattern WLAN forbundet direkte til LAN



WLAN AP'er forbundet direkte til LAN giver risiko for at sikkerheden brydes, fordi AP falder tilbage på den usikre standardkonfiguration

Ved at sætte WLAN direkte på LAN risikerer man at eksterne får direkte adgang  
Kan selvfølgelig gå an i et privat hjem  
Det forværres jo flere AP'er man har, har du 100 skal du være sikker på allesammen er sikre!

# Hackerværktøjer

Dan Farmer og Wietse Venema skrev i 1993 artiklen  
*Improving the Security of Your Site by Breaking Into it*

Senere i 1995 udgav de så en softwarepakke med navnet SATAN *Security Administrator Tool for Analyzing Networks*. Pakken vakte en del furore, idet man jo gav alle på internet mulighed for at hænge.

We realize that SATAN is a two-edged sword - like many tools, it can be used for good and for evil purposes. We also realize that intruders (including wannabees) have much more capable (read intrusive) tools than offered with SATAN.

SATAN og ideerne med automatiseret scanning efter sårbarheder blev siden ført videre i programmer som Saint, SARA og i dag findes mange hackerværktøjer og automatiserede scannere:

- Nessus, ISS scanner, Fyodor Nmap, Typhoon, ORAScan

Kilde: <http://www.fish.com/security/admin-guide-to-cracking.html>

# Brug hackerværktøjer!

Hackerværktøjer - bruger I dem? - efter dette kursus gør I portscannere kan afsløre huller i forsvaret  
webtestværktøjer som crawler igennem et website og finder alle forms kan hjælpe  
I vil kunne finde mange potentielle problemer proaktivt ved regelmæssig brug af disse  
værktøjer - også potentielle driftsproblemer  
husk dog penetrationstest er ikke en sølvkugle  
honeypots kan måske være med til at afsløre angreb og kompromitterede systemer  
hurtigere

# "I only replaced index.html"

Hvad skal manøre når man bliver hacket ?

Hvad koster et indbrud?

- Tid - antal personer der ikke kan arbejde
- Penge - oprydning, eksterne konsulenter
- Bøvl - sker altid på det værst mulige tidspunkt
- Besvær - ALT skal gennemrodes
- Tab af image/goodwill

Forensic challenge: I gennemsnit brugte deltagerne 34 timer pr person på at efterforske i rigtige data fra et indbrud! angriberen brugte ca. 30 min

Kilder: <http://project.honeynet.org/challenge/results/>

<http://packetstorm.securify.com/docs/hack/i.only.replaced.index.html.txt>

# Recovering from break-ins

## **DU KAN IKKE HAVE TILLID TIL NOGET**

På CERT website kan man finde mange gode ressourcer omkring sikkerhed og hvad man skal gøre med kompromiterede servere

Eksempelvis listen over dokumenter fra adressen:

<http://www.cert.org/nav/recovering.html>

- The Intruder Detection Checklist
- Windows NT Intruder Detection Checklist
- The UNIX Configuration Guidelines
- Windows NT Configuration Guidelines
- The List of Security Tools
- Windows NT Security and Configuration Resources

# Opsummering

Husk følgende:

- UNIX og Linux er blot eksempler - navneservice eller HTTP server kører fint på Windows
- DNS er grundlaget for Internet
- Sikkerheden på internet er generelt dårlig, brug SSL!
- Procedurerne og vedligeholdelse er essentiel for alle operativsystemer!
- Man skal *hærde* operativsystemer *før* man sætter dem på Internet
- Husk: IT-sikkerhed er ikke kun netværkssikkerhed!
- God sikkerhed kommer fra langsigtede intiativer

Jeg håber I har lært en masse om netværk og kan bruge det i praksis :-)

# Spørgsmål?



Henrik Lund Kramshøj  
hlk@security6.net

<http://www.security6.net>

I er altid velkomne til at sende spørgsmål på e-mail

## Referencer: netværksbøger

- Stevens, Comer,
- Network Warrior
- TCP/IP bogen på dansk
- KAME bøgerne
- O'Reilly generelt IPv6 Essentials og IPv6 Network Administration
- O'Reilly cookbooks: Cisco, BIND og Apache HTTPD
- Cisco Press og website
- Firewall bøger, Radia Perlman: IPsec,

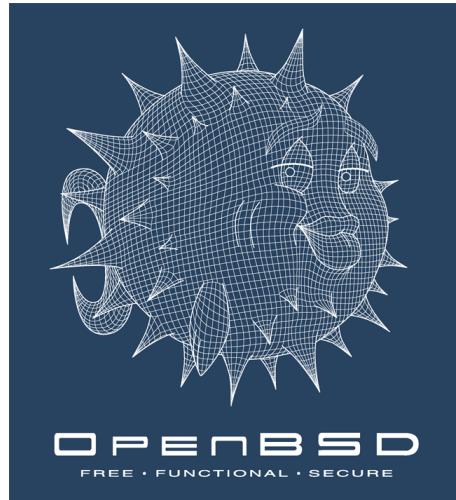
## Bøger om IPv6

*IPv6 Network Administration* af David Malone og Niall Richard Murphy - god til real-life admins, typisk O'Reilly bog

*IPv6 Essentials* af Silvia Hagen, O'Reilly 2nd edition (May 17, 2006) god reference om emnet

*IPv6 Core Protocols Implementation* af Qing Li, Tatuya Jinmei og Keiichi Shima

*IPv6 Advanced Protocols Implementation* af Qing Li, Jinmei Tatuya og Keiichi Shima  
- flere andre



Primære website: <http://www.openbsd.org>

Ved at støtte OpenBSD støtter du:

- OpenSSH - inkluderet i mere end 80-100 distributioner
- Udviklingen af OpenBSD PF - en super firewall, er med i FreeBSD, NetBSD
- Udvikling af stackprotection i Open Source operativsystemer
- OpenBGPD - en fri routing daemon, OpenNTPD - en fri NTP daemon, OpenCVS - en fri NTP daemon, CARP - redundancy must be free!

**(ISC)<sup>2<sub>SM</sub></sup>**

**(CISSP)<sup>®</sup>**

**(SSCP)<sup>CM</sup>**

Approved marks of the International Information Systems Security Certification Consortium, Inc.

Primære website: <http://www.isc2.org>

Vigtigt link <http://www.cccure.org/>

Den kræver mindst 3 års erfaring indenfor et relevant fagområde

Multiple choice 6 timer 250 spørgsmål - kan tages i Danmark



TM

Certified Ethical Hacker

Bredt kendskab til hackermetoder

Kursus+eksamen

Eksamten kan tages alene såfremt man kan demonstrere kendskab til emnet

150 spørgsmål - 4 timer



Security Essentials - basal sikkerhed

Krav om en *Practical assignment* - mindst 8 sider, 15 sider i gennemsnit  
multiple choice eksamen

Primære website: <http://www.giac.org>

Reading room: <http://www.sans.org/rr/>

Der findes en god oversigt i filen *GIAC Certification: Objectives and Curriculum*  
[http://www.giac.org/GIAC\\_Cert\\_Brief.pdf](http://www.giac.org/GIAC_Cert_Brief.pdf)

# Referencer

## Papers - der findes MANGE dokumenter på Internet

- CERT/CC <http://www.cert.org>
- AusCERT Computer Emergency Response Team for Australia  
<http://www.auscert.org.au/>
- <http://www.securityfocus.com>
- CERIAS hotlist [http://www.cerias.purdue.edu/tools\\_and\\_resources/hotlist/](http://www.cerias.purdue.edu/tools_and_resources/hotlist/)

## Honeypots og sårbare systemer

- <http://www.project.honeynet.org> - diverse honeynet projekter information om pakker og IP netværk

Husk også at mange forlag tillader at man henter et kapitel som PDF!

# Referencer: bøger

- *Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, Ed Skoudis, Prentice Hall PTR, 1st edition July 2001
- *CISSP All-in-One Exam Guide*, Shon Harris, McGraw-Hill Osborne Media, 2nd edition, June 17 2003
- *Practical UNIX and Internet Security*, Simson Garfinkel, Gene Spafford, Alan Schwartz, 3rd Edition, O'Reilly February 2003
- *Network Security Assessment: Know Your Network*, Chris McNab, O'Reilly March 2004
- *Secure Coding: Principles & Practices*, Mark G. Graff, Kenneth R. van Wyk, O'Reilly June 2003
- *Firewalls and Internet Security*, William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, Addison-Wesley, 2nd edition, 2003
- *Building Firewalls with OpenBSD and PF*, Jacek Artymiak, 2nd edition 2003
- bøger om TCP/IP - Alle bøger af Richard W Steven kan anbefales!
- Reference books for the CISSP CBK domains - en liste der vedligeholdes af Rob Slade  
<http://victoria.tc.ca/int-grps/books/techrev/mnbksccd.htm>

# Hackerværktøjer

- nmap - <http://www.insecure.org> portscanner
- Nessus - <http://www.nessus.org> automatiseret testværktøj
- l0phtcrack - <http://www.atstake.com/research/lc/> - The Password Auditing and Recovery Application, kig også på Cain og Abel fra <http://oxid.it> hvis det skal være gratis
- Ethereal - <http://www.ethereal.com> avanceret netværkssniffer
- OpenBSD - <http://www.openbsd.org> operativsystem med fokus på sikkerhed
- <http://www.isecom.org/> - Open Source Security Testing Methodology Manual - gennemgang af elementer der bør indgå i en struktureret test
- Putty - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> terminal emulator med indbygget SSH
- <http://www.remote-exploit.org/> - BackTrack security collection - en boot CD med omkring 300 hackerværktøjer

# Hvordan bruges hackerværktøjerne

Tænk som en hacker

Rekognoscering

- ping sweep
- portscan
- OS detection - TCP/IP eller banner grab
- Servicescan - rpcinfo, netbios, ...
- telnet/netcat interaktion med services

Udnyttelse/afprøvning: Nessus, whisker, exploit programs

Oprydning

- Lav en rapport
- Gennemgå rapporten, registrer ændringer
- Opdater programmer, konfigurationer, arkitektur, osv.

I skal jo også VISE andre at I gør noget ved sikkerheden.

# Reklamer: kursusafholdelse

Security6.net afholder følgende kurser med mig som underviser

- IPv6 workshop - 1 dag  
Introduktion til Internetprotokollerne og forberedelse til implementering i egne netværk. Internetprotokollerne har eksisteret i omkring 20 år, og der er kommet en ny version kaldet version 6 af disse - IPv6.
- Wireless teknologier og sikkerhed workshop - 2 dage  
En dag med fokus på netværksdesign og fornuftig implementation af trådløse netværk og integration med eksempelvis hjemmepc og virksomhedens netværk
- Hacker workshop 2 dage  
Workshop med detaljeret gennemgang af hackermetoderne angreb over netværk, exploitprogrammer, portscanning, Nessus m.fl.
- Forensics workshop 2 dage  
Med fokus på tilgængelige open source værktøjer gennemgås metoder og praksis af undersøgelse af diskimages og spor på computer systemer
- Moderne Firewalls og Internetsikkerhed 2 dage  
Informere om trusler og aktivitet på Internet, samt give et bud på hvorledes en avanceret moderne firewall idag kunne konfigureres.

BSD-DK - dansk forening for BSD'erne,

<http://www.bsd-dk.dk>

SSLUG, Skåne Sjælland Linux User Group

<http://www.sslug.dk>

DKUUG, Dansk UNIX User Group

<http://www.dkuug.dk>

medlemsskab giver god rabat på bøger gennem

<http://www.polyteknisk.dk>, typisk 15-20%



# Soekris bestilling

Et lille embedded system

- Soekris 5501-30 + case ..... 2250,-
- Soekris 4801-50 + case ..... 1400,-
- Strømforsyning 1.5A (lille) ..... 130,-
- Strømforsyning 3A (stor) ..... 170,-
- vpn1411 miniPCI ..... 400,-
- 4801 Harddisk mount kit 2.5" ..... 70,-
- Alle priser er cirkapriser og ekskl. moms. kontakt leverandører for nøjagtige oplysninger!
- Anbefalet leverandør <http://www.kd85.com>
- Alternativ leverandør <http://www.cortexsystems.dk>, men de er ekstremt dårlige til kundepleje - så køb kun hvis det er på lager!

