

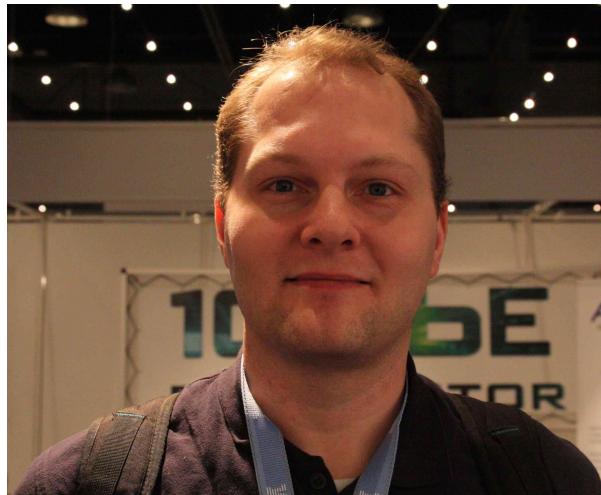
Welcome to

# Netværkssikkerhed i firmanetværk

Henrik Lund Kramshøj, internet samurai  
[hlk@solido.net](mailto:hlk@solido.net)

<http://www.solidonetworks.com>

# Kontaktinformation og profil



- Henrik Lund Kramshøj, IT-sikkerhed og internet samurai
- Email: [hlk@solido.net](mailto:hlk@solido.net)      Mobil: +45 2026 6000
- Cand.scient fra Datalogisk Institut ved Københavns Universitet, DIKU
- CISSP certificeret
- 2003 - 2010 Selvstændig sikkerhedskonsulent
- 2010 Stifter og partner i Solido Networks ApS

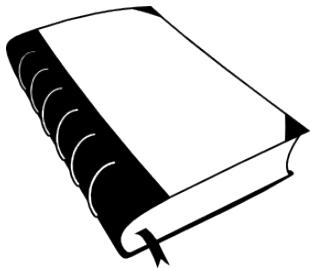
Vi skal have glæde af hinanden i følgende kursusforløb

- 2 dage med workshop ca. 6 timer per dag

I skal uover at lære en masse om protokoller og netværk

Forhåbentlig lærer i nogle gode vaner!

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



Free graphics by Lumen Design Studio

Dette materiale består af flere dele:

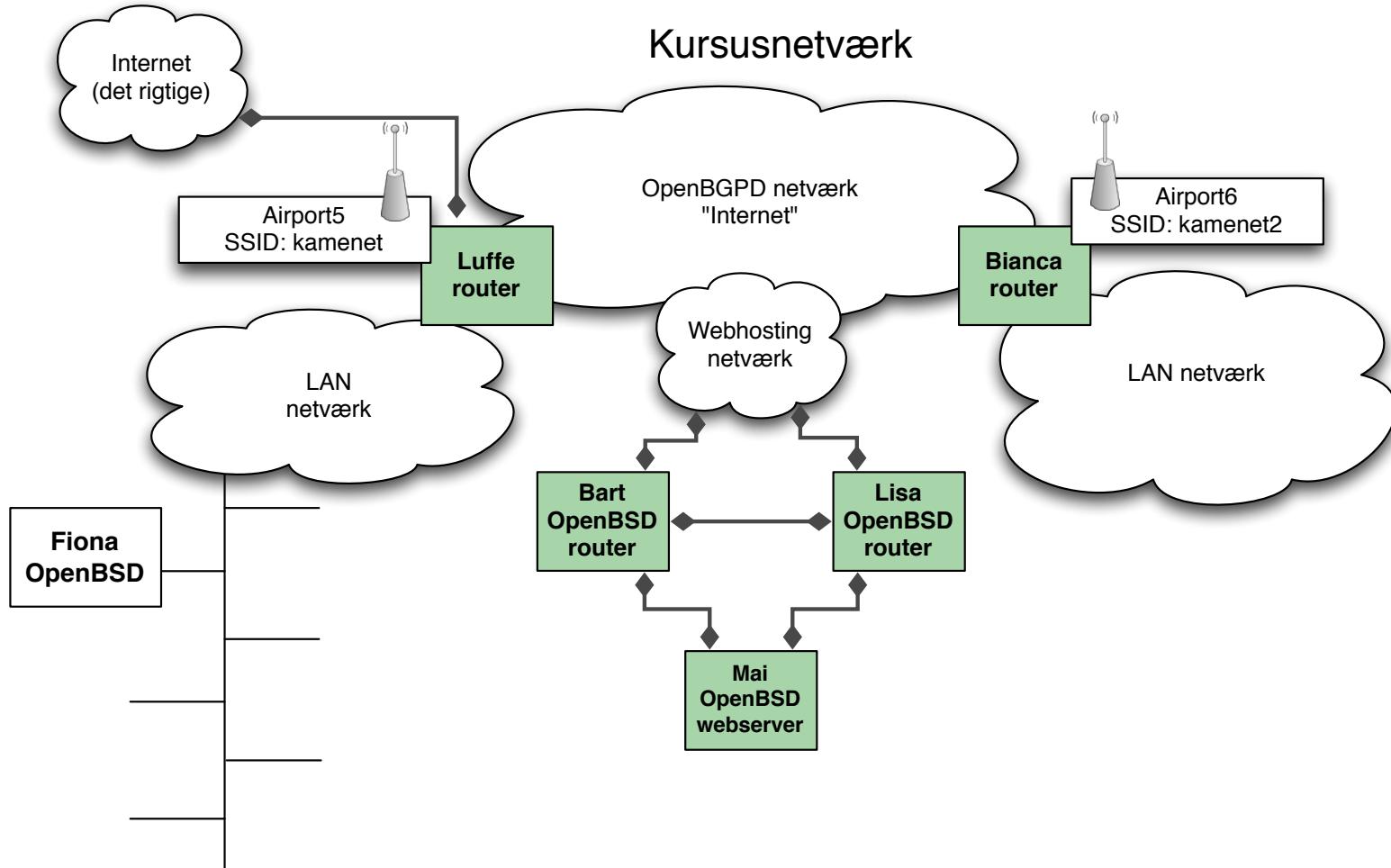
- Kursusmaterialet - præsentationen til undervisning - dette sæt
- Øvelseshæfte med øvelser

Hertil kommer diverse ressourcer fra internet

Der er USB sticks med VMware Player og ISO images for BackTrack

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

# Formål: netværkssikkerhed for TCP/IP netværk



TCP/IP-baserede netværk - internet er overalt

# Formål: mere specifikt

At introducere god sikkerhed i TCP/IP netværk for firmaer

Kendskab til almindeligt brugte protokoller

- VLAN, WLAN, DNS, RADIUS, LDAP m.v.

Kendskab til almindelige værktøjer i disse miljøer

- ping, traceroute, iperf, Smokeping, Nagios, Apache Benchmark m.v.

Gennemgang af netværksdesign ved hjælp af almindeligt brugte setups

- en skalamodel af internet

NB: ja, jeg bruger en masse Unix og webapplikationer på Unix

**men de fleste af programmerne KAN installeres på Windows,  
eller der kan findes alternativer der benytter samme protokoller!**

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

Kurset anvender Linux til øvelser, men Unix kendskab er ikke nødvendigt

De fleste øvelser kan udføres fra en Windows PC

Øvelserne foregår via

- Login til Unix maskinen
- Direkte fra jeres systemer Windows eller Linux boot CD
- Via administrationsprogrammer, ofte webinterfaces

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

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**

Login: **kursus1**

Password: **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 el kursus**



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

BackTrack Linux <http://www.backtrack-linux.org/> BackTrack er baseret på Linux og må kopieres frit :-)

Til begyndere indenfor Linux anbefales Ubuntu eller Kubuntu til arbejdsstationer

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**

**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!

Opstart - hvad er IP og TCP/IP repetition, TCP, UDP, Subnets og CIDR

Basale værktøjer traceroute, ping, dig, host

Wireshark sniffer

SSL Secure Sockets Layer

VLAN 802.1q

Målinger iperf, apache benchmark (ab)

Tuning og perfomancemålinger

Plus diverse webinterfaces og administrationsværktøjer

Management, SNMP, RRDTool og Smokeping

Overvågning og diagnosticering Nagios, syslog m.v.

Trådløse netværk og sikkerhed Wi-Fi Protected Access (WPA)

Sikkerhedsværktøjer til trådløse netværk aircrack-ng suiten af værktøjer

Directory services RADIUS, LDAP m.v.

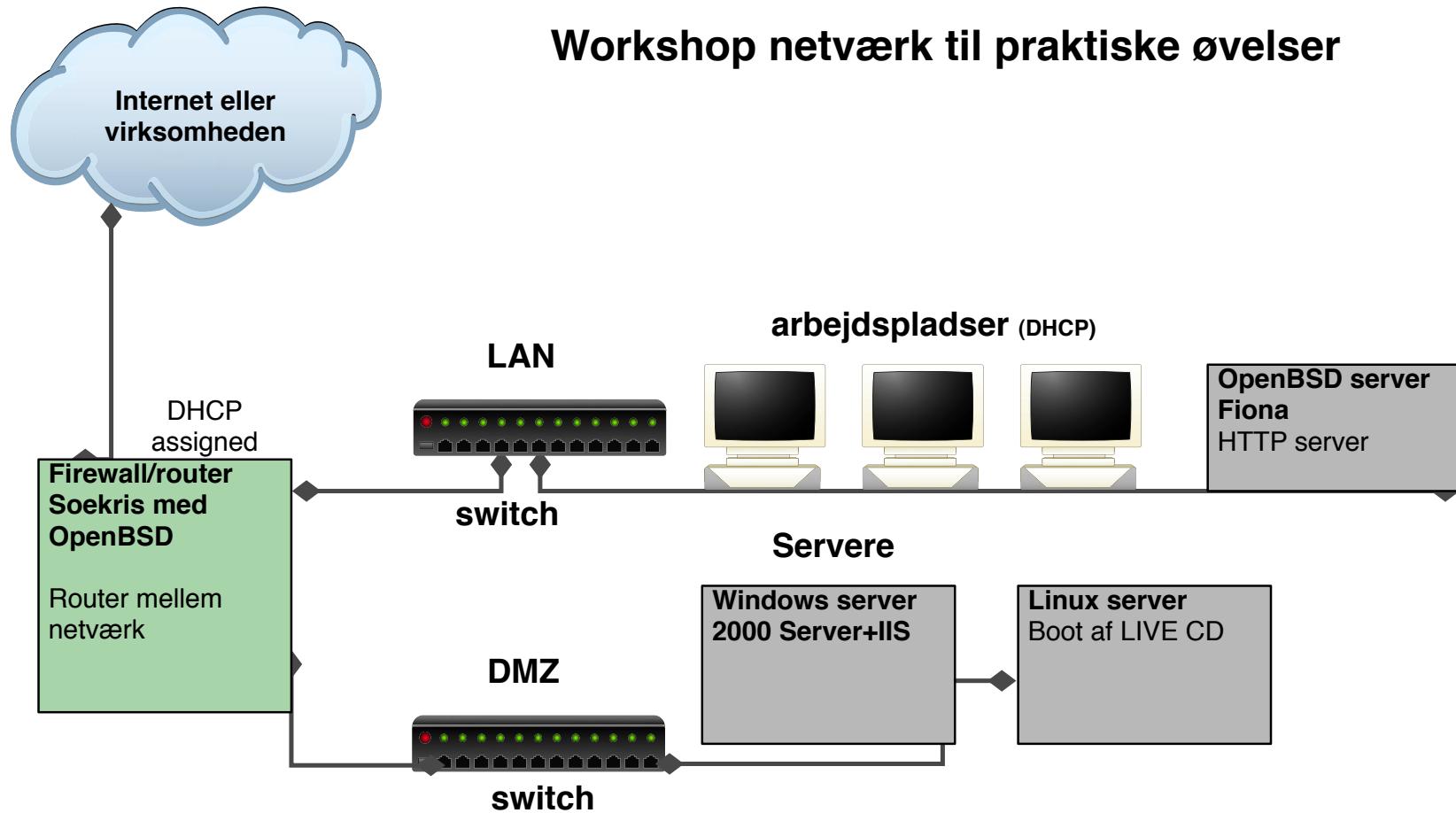
Avancerede teknologier som 802.1x portbaseret autentifikation

VoIP - kort introduktion

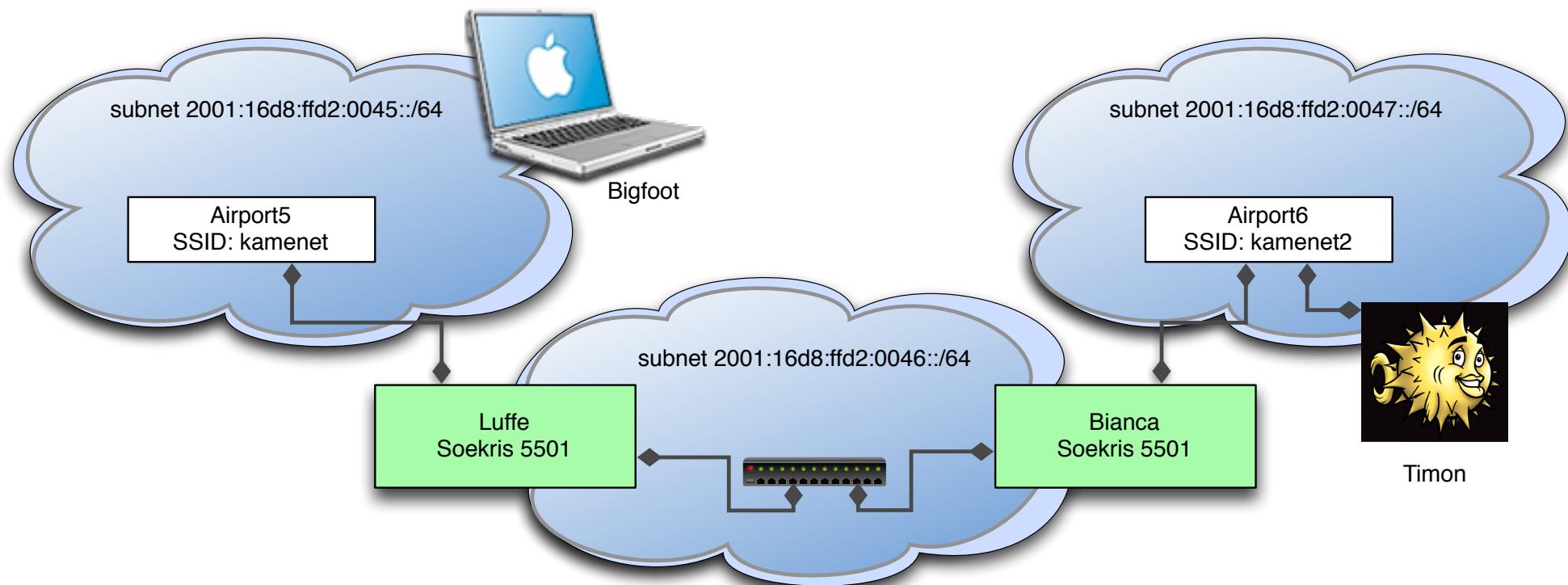
Firewalls, VPN protokoller og IPSec - kort introduktion

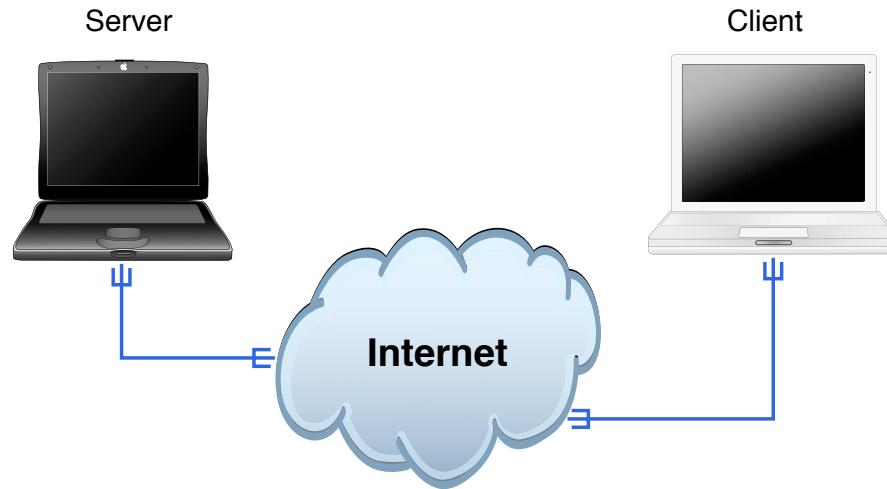
Infrastrukturer i praksis og netværksdesign

Afslutning og opsummering på kursus



# 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

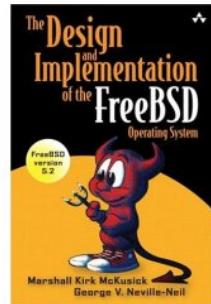
Kommunikation mellem mennesker!

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.



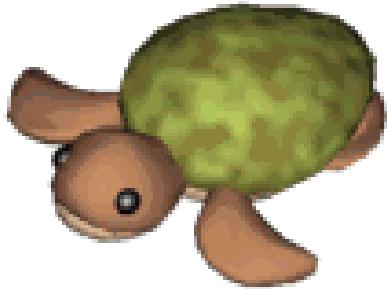
På Berkeley Universitetet blev der udviklet en del på Unix og det har givet anledning til en hel gren kaldet BSD Unix, BSD står for Berkeley Software Distribution

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

Specielt har BSD TCP/IP kernefunktionalitet været genbrugt mange steder

Tilsvarende genbruges KAME IPv6 implementationen mange steder

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



<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

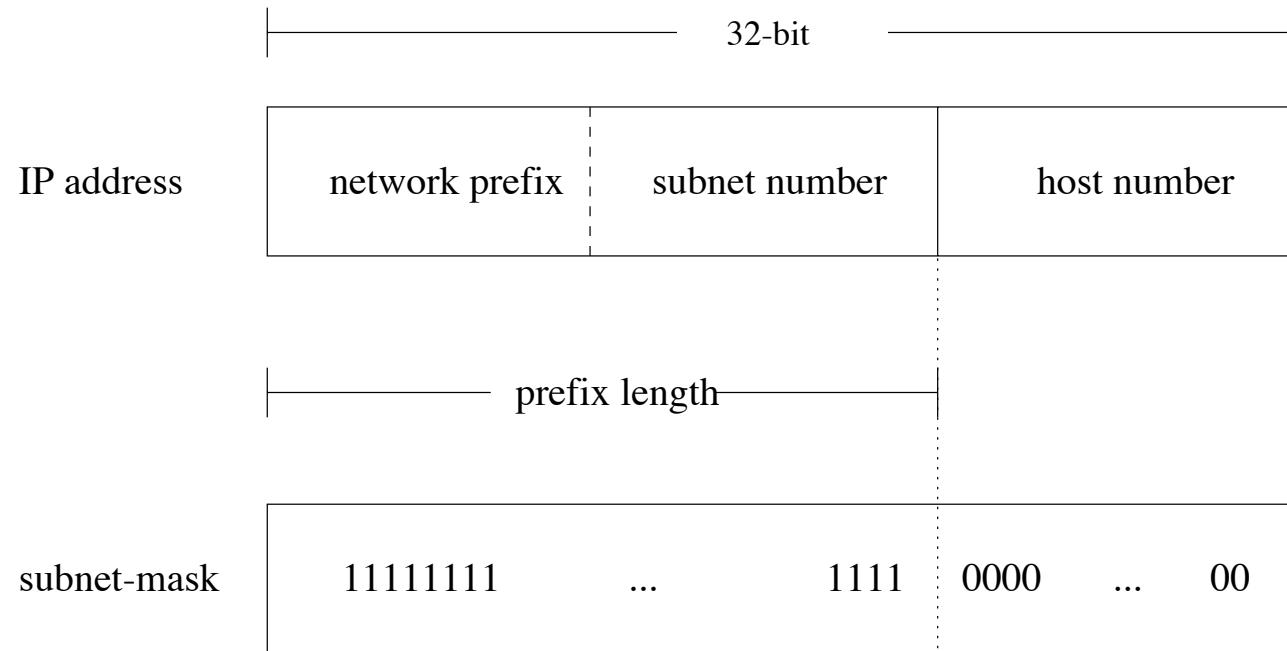
80'erne IP/TCP starten af 80'erne

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



Hvad kendetegner internet idag

Der er et fælles adresserum baseret på 32-bit adresser

En IP-adresse kunne være 10.0.0.1

```
hlk@bigfoot:hlk$ ipconvert.pl 127.0.0.1
Adressen er: 127.0.0.1
Adressen 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-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)

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 udregne subnetmasker til IPv4

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

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

- 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

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 webserver's IPv6 adresse kan skrives som: 2001:16d8:ff00:12f::2
- Specielle adresser: ::1 localhost/loopback og :: default route
- Læs mere i RFC-3513

Bemærk hvilket netværk vi bruger idag

Primære server fiona har IP-adressen 10.0.45.36

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 :-)

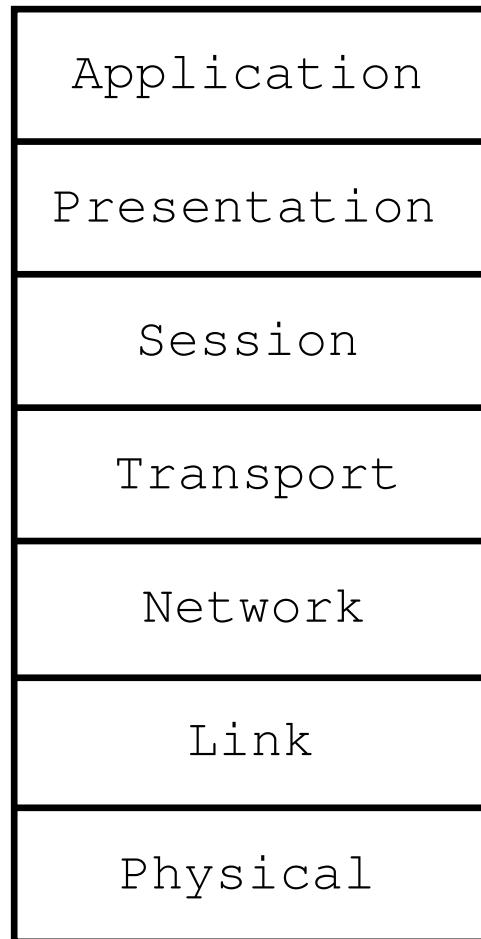
## **Det er forbudt at ændre IP-konfiguration og passwords**

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

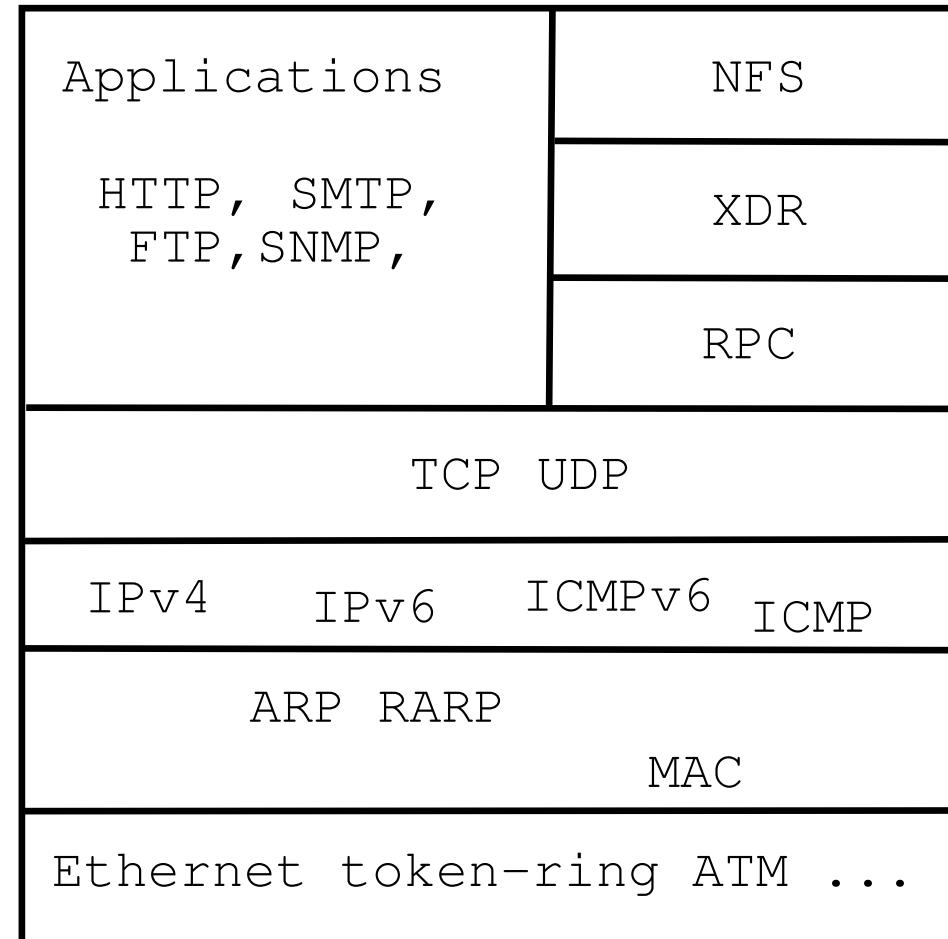
Det er nok at en PC i hver gruppe er på kursusnetværket

Pause for dem hvor det virker, mens vi ordner resten

OSI Reference Model



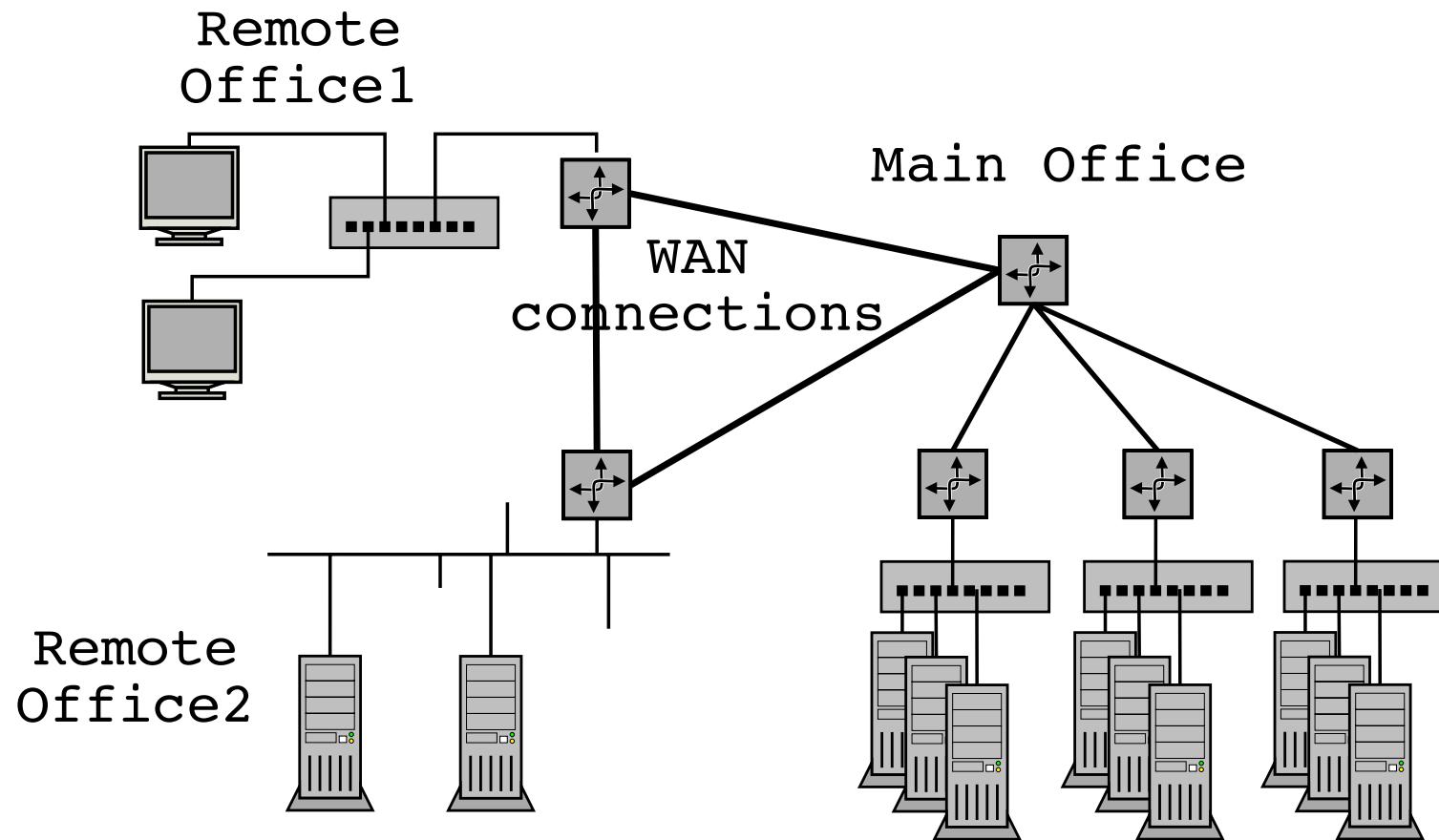
Internet protocol suite





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

Første dag bruger vi blot trådløse netværk, på dag 2 gennemgår vi 802.11



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

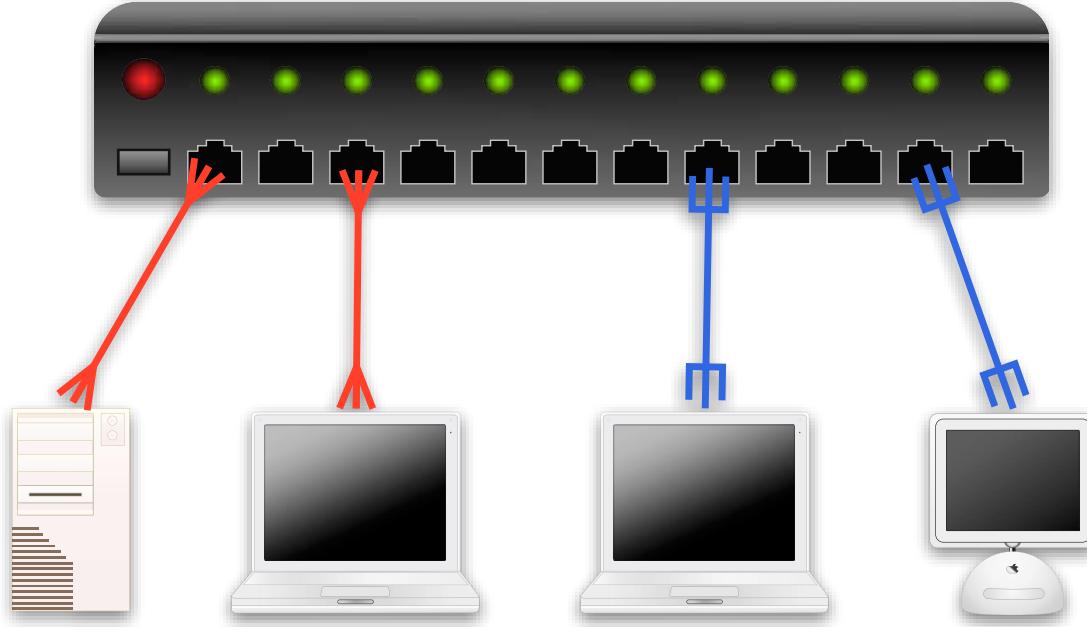
Den intelligente bro kender MAC adresserne på hver side

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

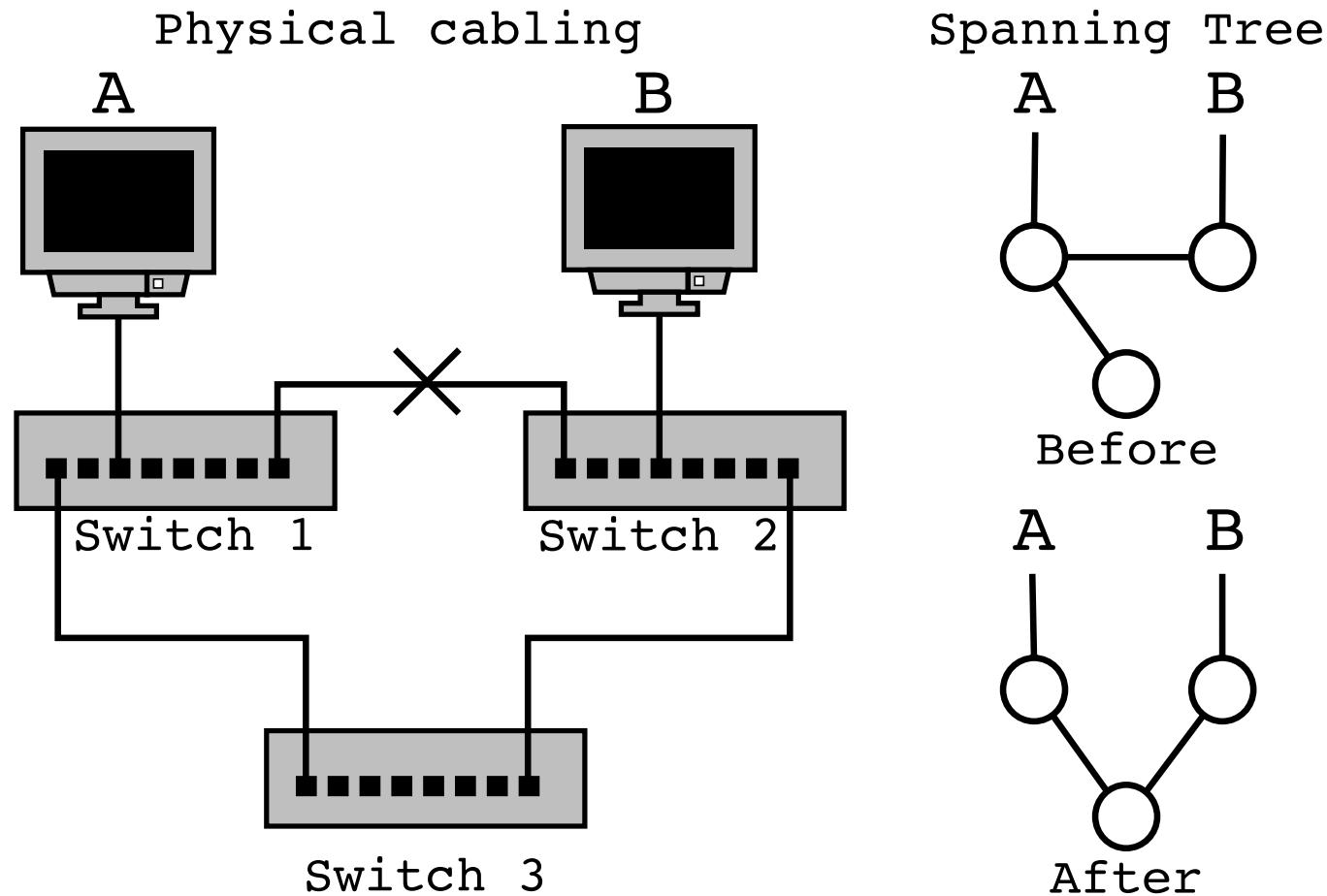


Ved at fortsætte udviklingen kunne man samle broer til en switch

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

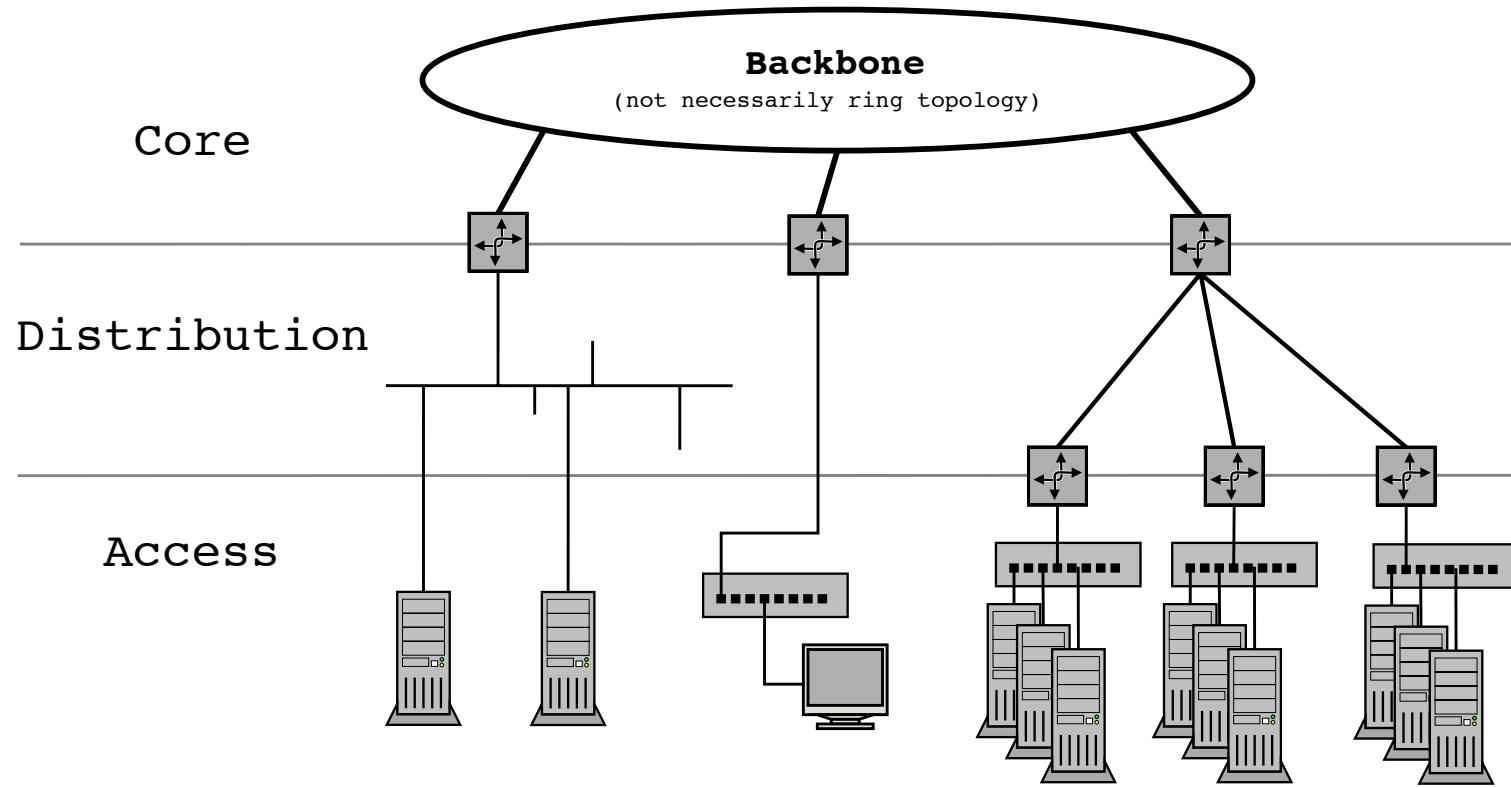
Bemærk performance begrænses af backplane i switchen

# Topologier og Spanning Tree Protocol

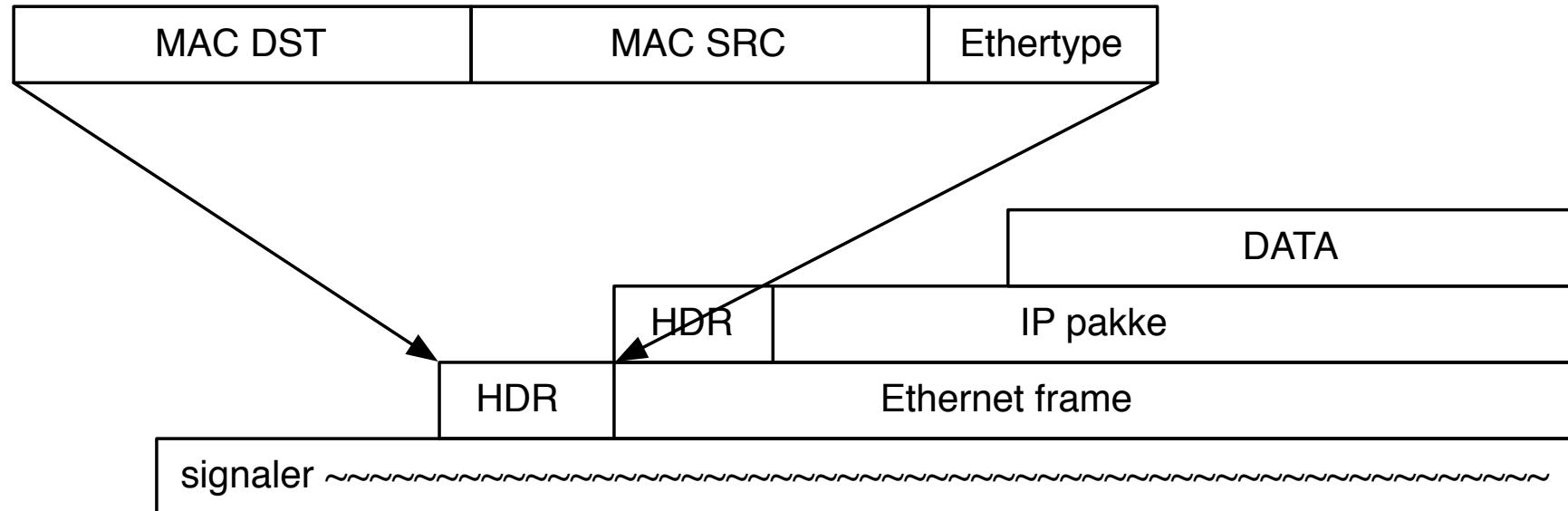


Se mere i bogen af Radia Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*

# Core, Distribution og Access net



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



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

```
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`

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)

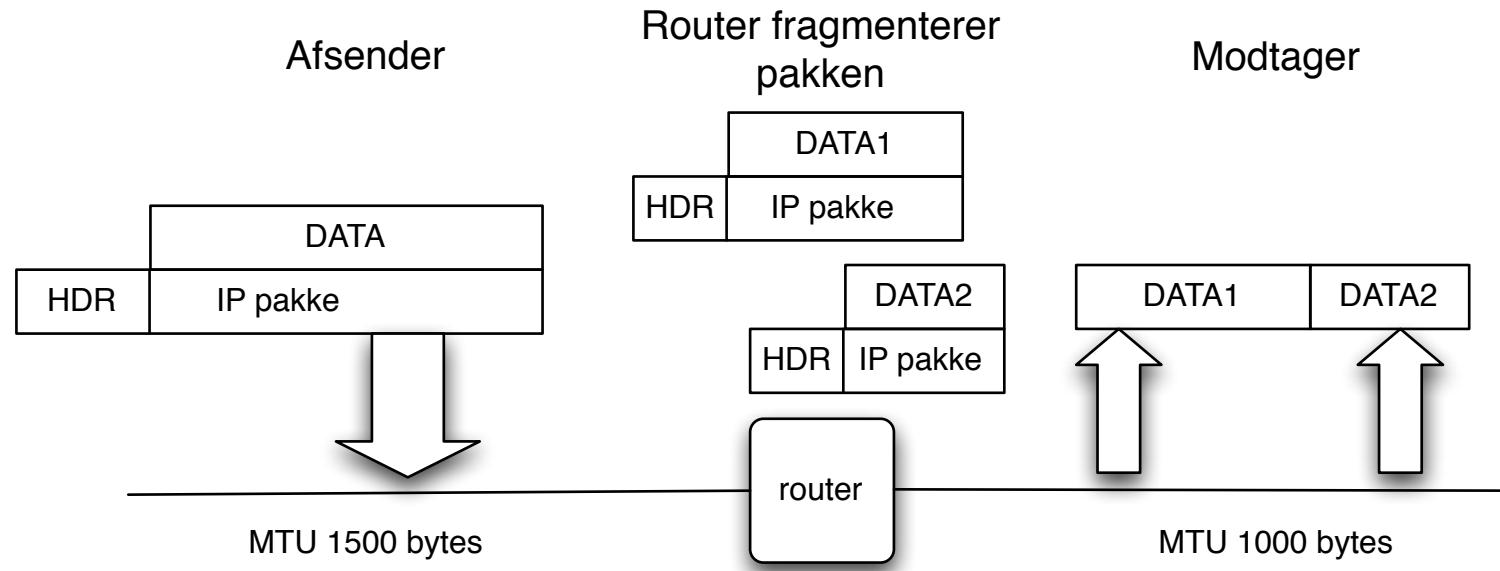
```
hlk@bigfoot:basic-ipv6-new$ arp -an
```

```
? (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]
```

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

Neighbor	Linklayer Address	Netif	Expires	St	Flgs	Prbs
::1	(incomplete)		1o0	permanent	R	
2001:16d8:fffd2:cf0f:21c:b3ff:fec4:e1b6	0:1c:b3:c4:e1:b6	en1	permanent	R		
fe80::1%lo0	(incomplete)		1o0	permanent	R	
fe80::200:24ff:fea8:b24c%en1	<b>0:0:24:c8:b2:4c</b>	en1	8h54m51s	S	R	
fe80::21c:b3ff:fea4:e1b6%en1	0:1c:b3:c4:e1:b6	en1	permanent	R		

# 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 *signalerings*

Defineret i RFC-792

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

## 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

## 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
```

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*

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 på Unix, og ICMP på windows

på Unix systemer er der ofte mulighed for at bruge ICMP, traceroute -I

```
$ 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
```

Husk at på Windows hedder kommandoen tracert

# 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
```

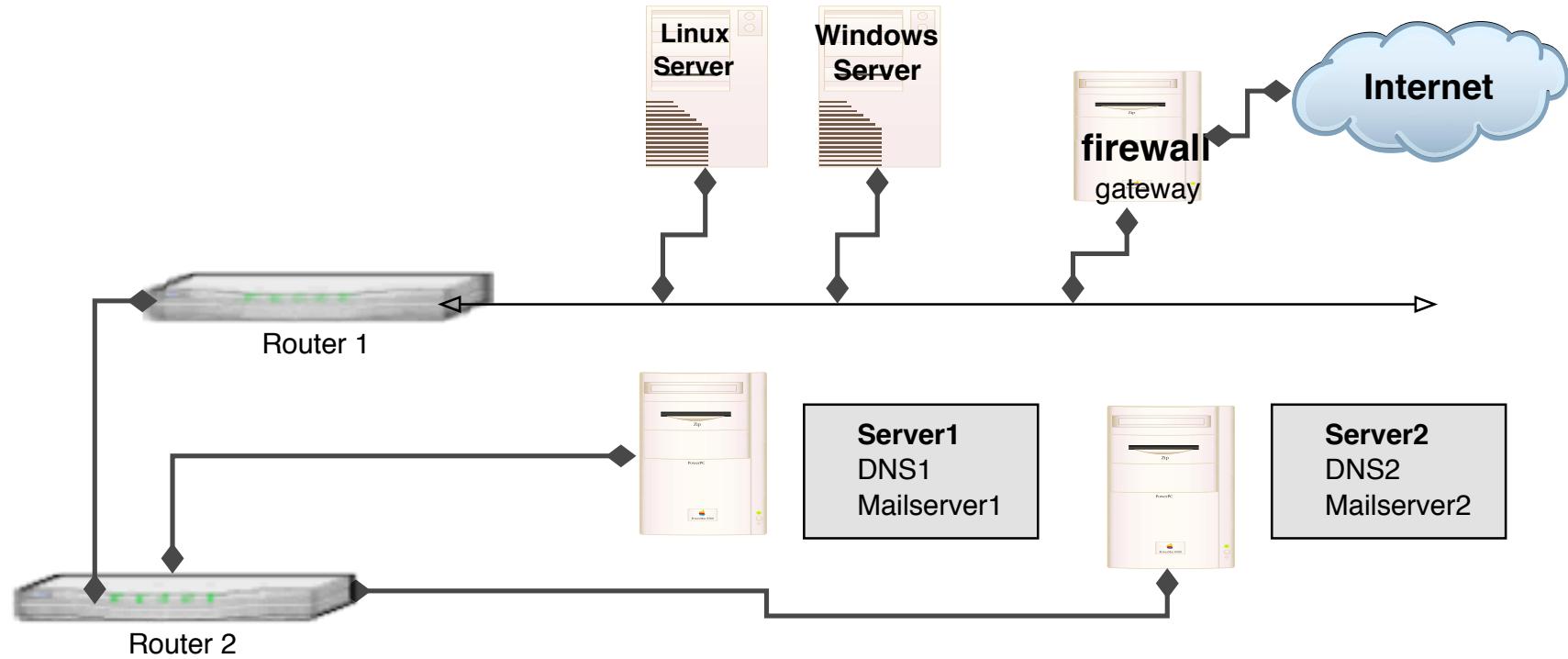
Diagnosticering af netværksproblemer - formålet med traceroute

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

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>

# Stop - vi gennemgår og tester vores setup

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

Vi prøver lige at se på diverse interfaces sammen

hvis alle prøver samtidig bliver det lidt kaos :-)

Huskliste til Henrik:

- Managed switch
- Airport Extreme access-point

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

```
[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

```
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)

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

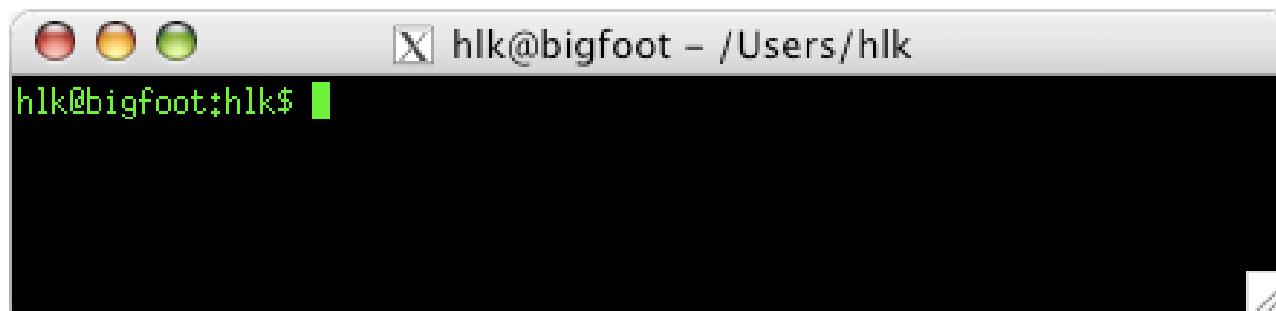
A cal command appeared in Version 6 AT&T Unix.



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

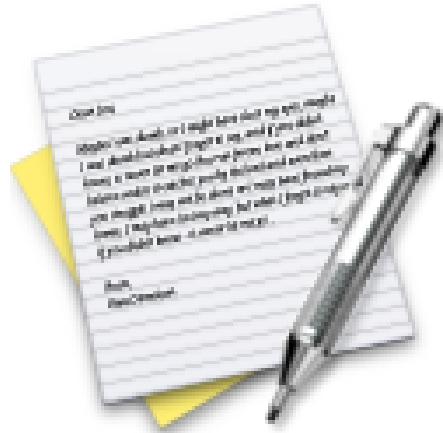
som er øvelse **2** fra øvelseshæftet.



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.

```
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.

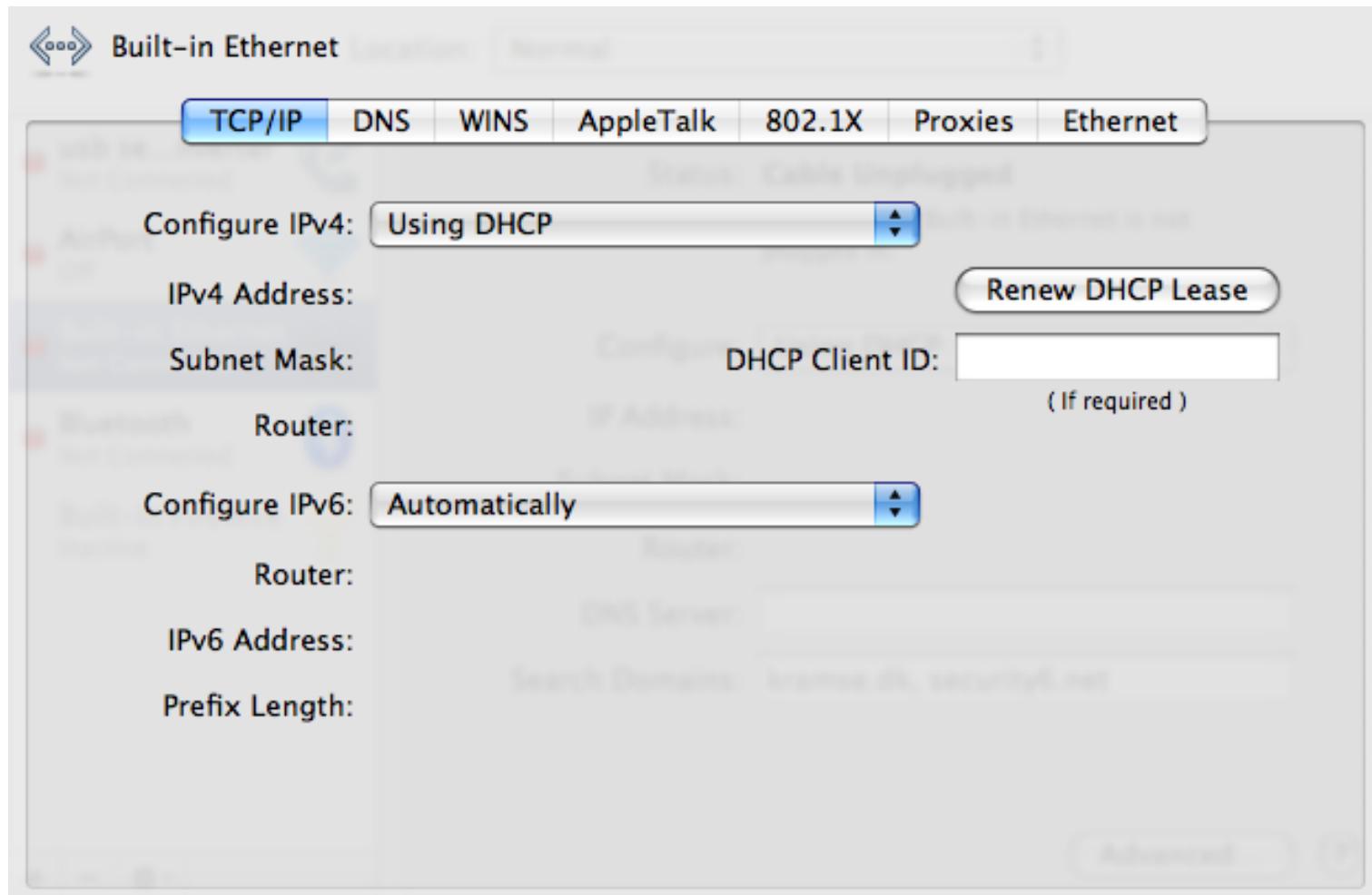
fejlsøgning foregår typisk med ping **og** traceroute

På Microsoft Windows benyttes ikke ifconfig  
men kommandoerne ipconfig **og** ipv6

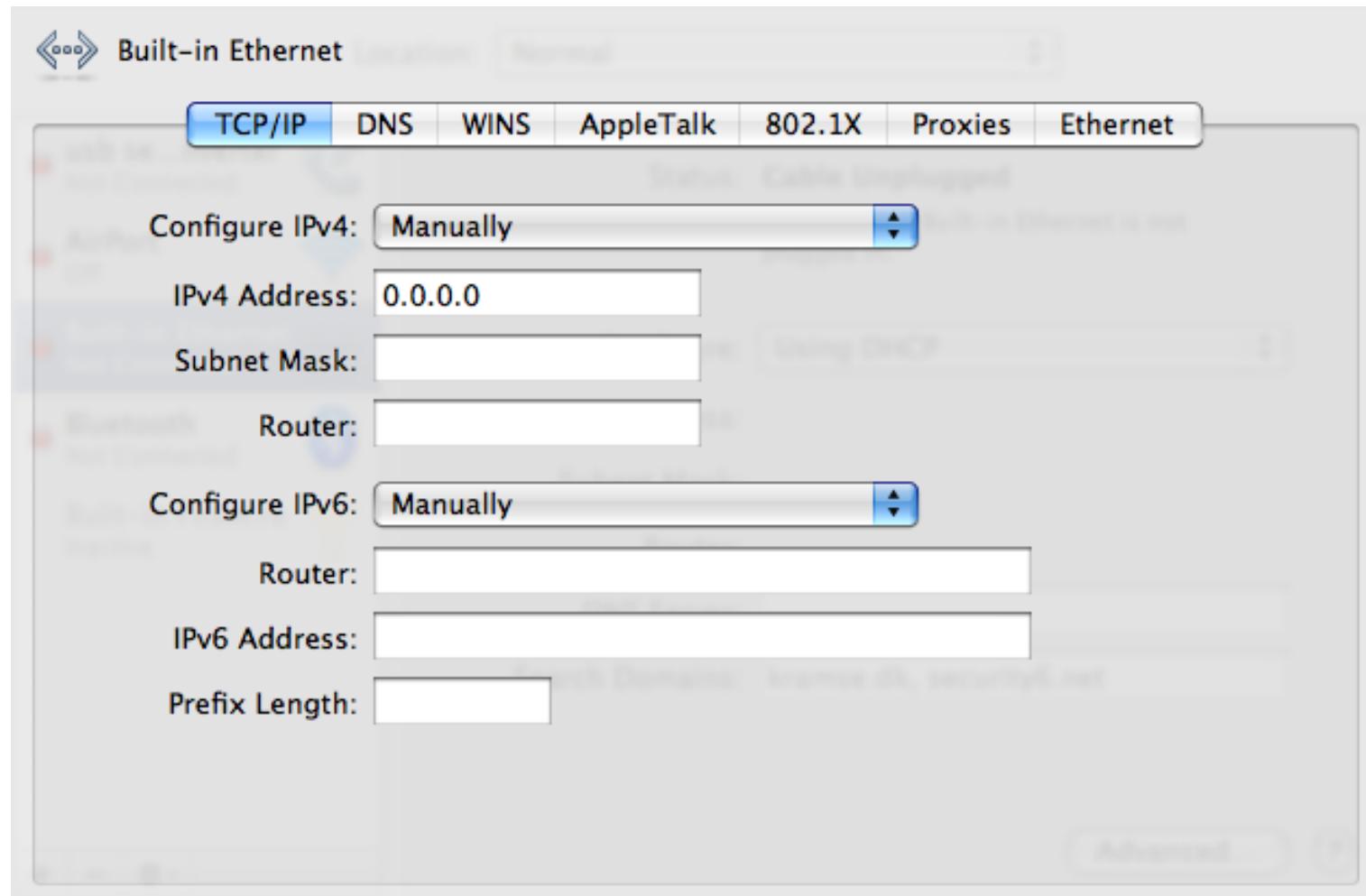
## Netværkskonfiguration på OpenBSD:

```
# 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
```

# GUI værktøjer - autoconfiguration



# GUI værktøjer - manuel konfiguration



# 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

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

UDP User Datagram Protocol

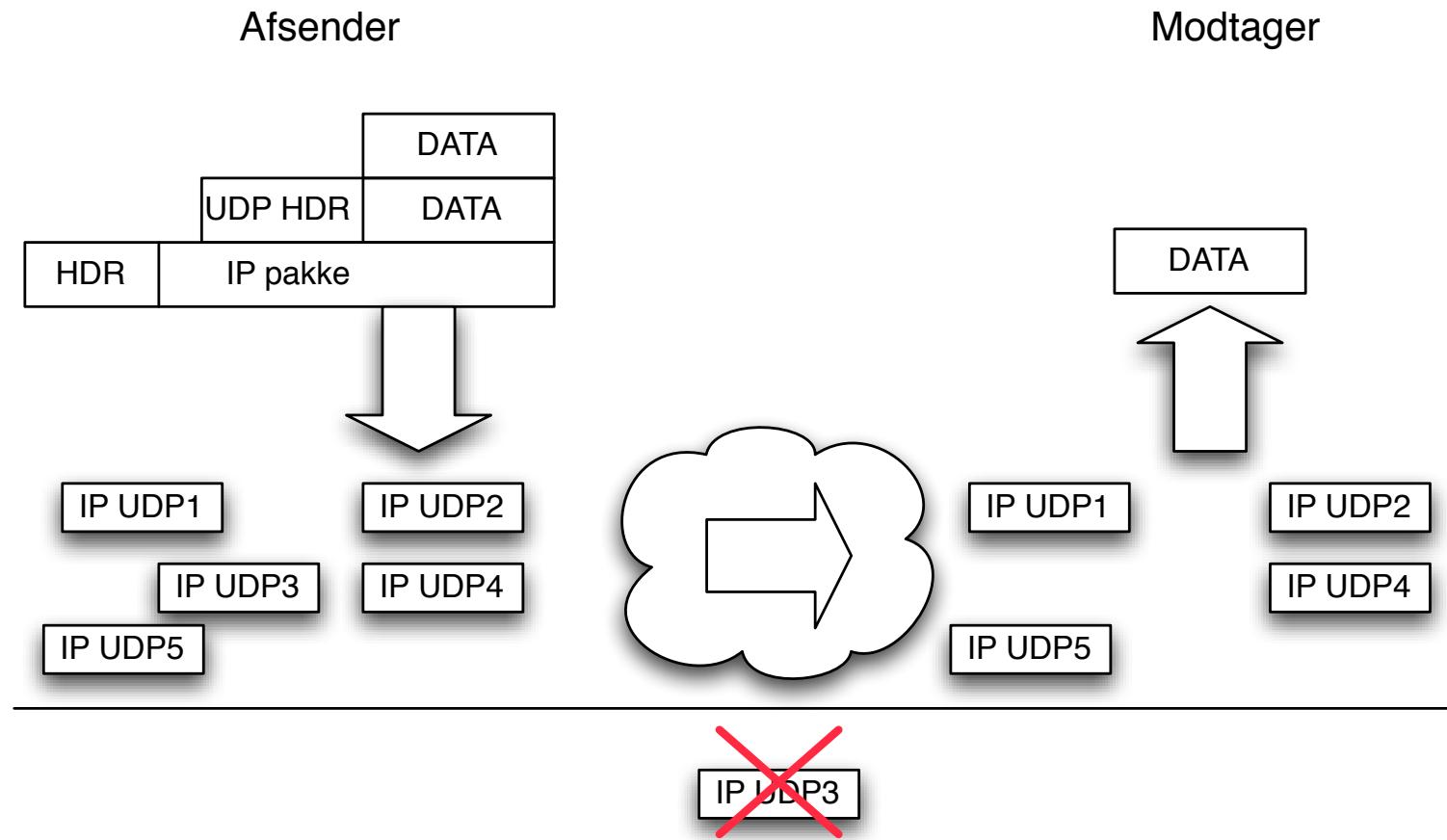
TCP Transmission Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Ovenstående er omtrent minimumskrav for at komme på internet

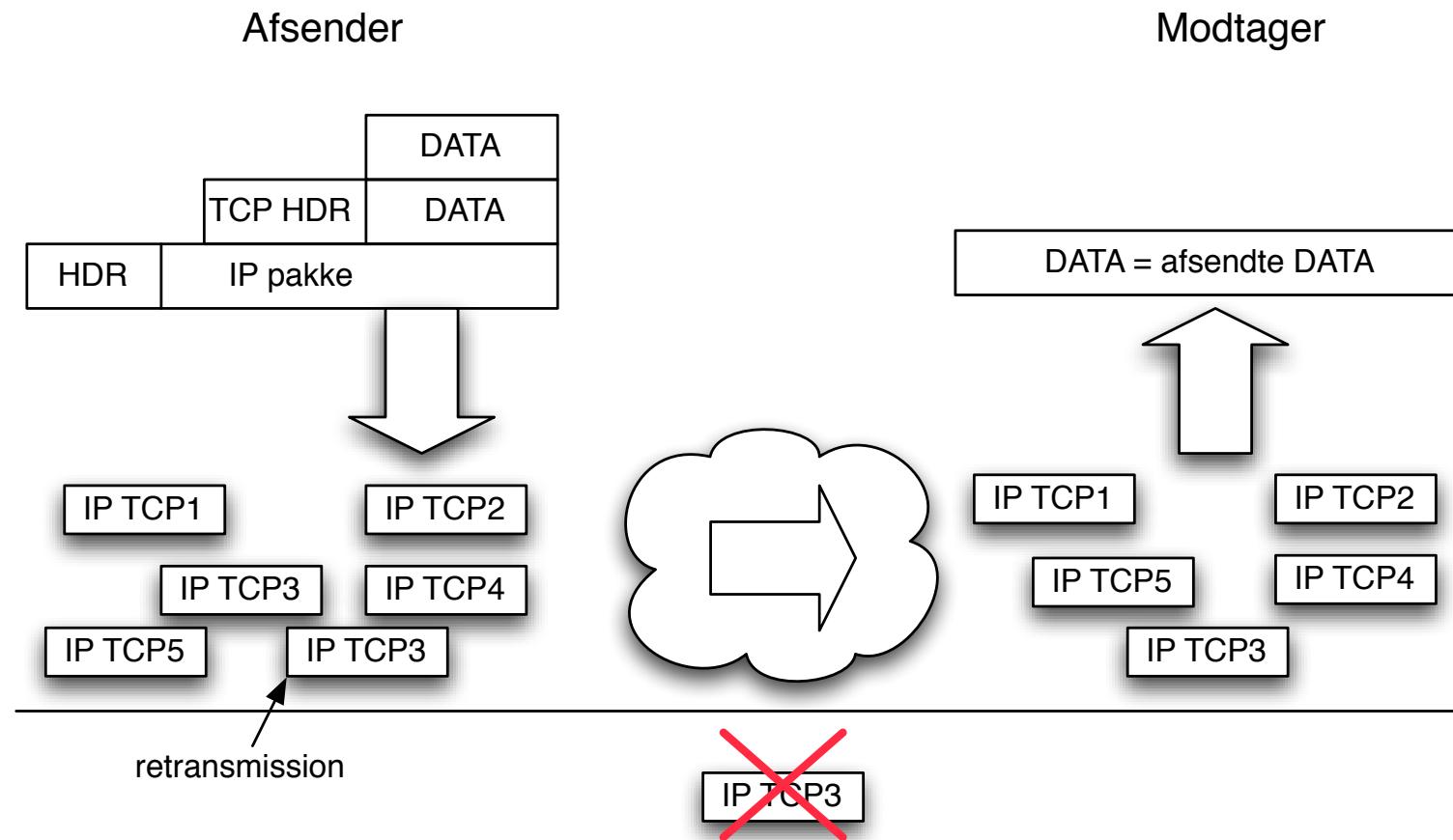
# UDP User Datagram Protocol



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

Kan benyttes til multicast/broadcast - flere modtagere

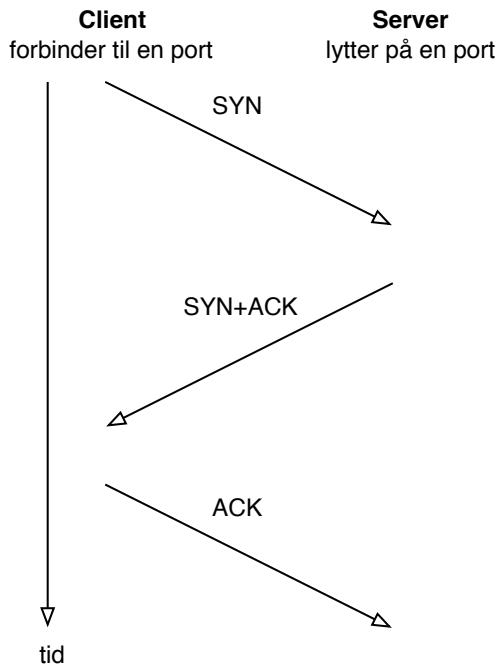
# TCP Transmission Control Protocol



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**



IANA vedligeholder en liste over magiske konstanter i IP

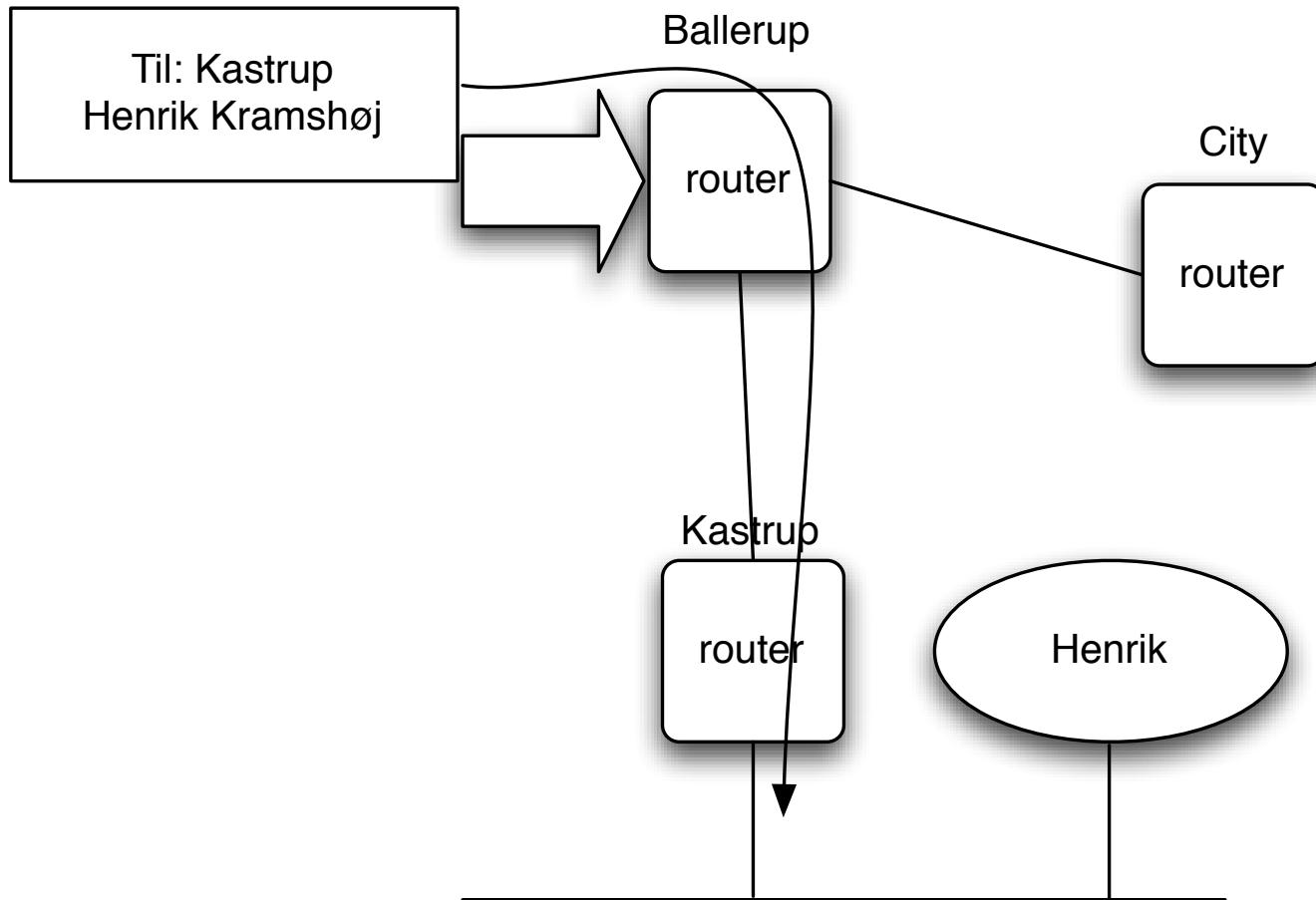
De har lister med hvilke protokoller har hvilke protokol ID m.v.

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

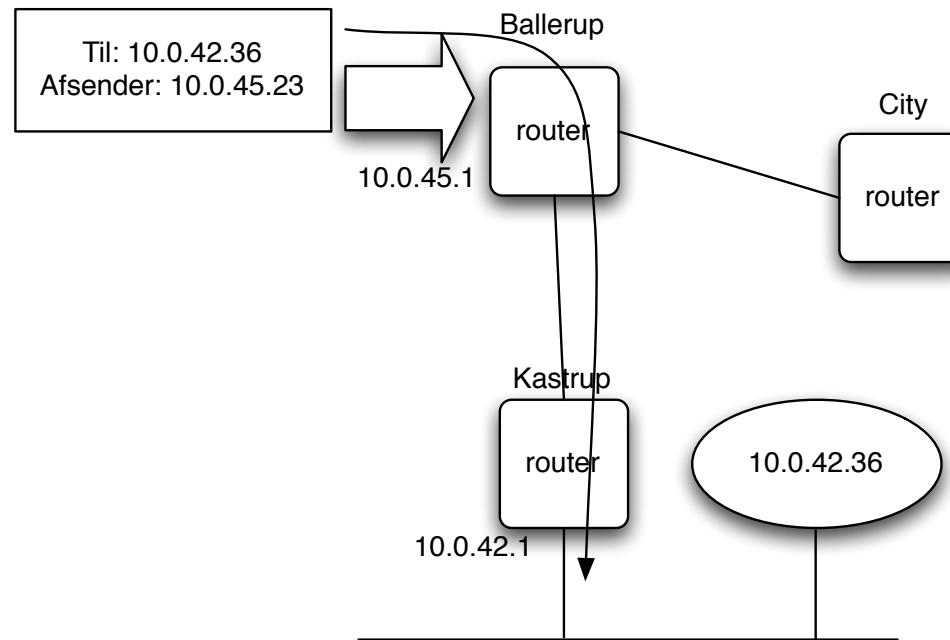
- Port 25 SMTP Simple Mail Transfer Protocol
- Port 53 DNS Domain Name System
- Port 80 HTTP Hyper Text Transfer Protocol over TLS/SSL
- Port 443 HTTP over TLS/SSL

Se flere på <http://www.iana.org>

# Hierarkisk routing



Hvordan kommer pakkerne frem til modtageren

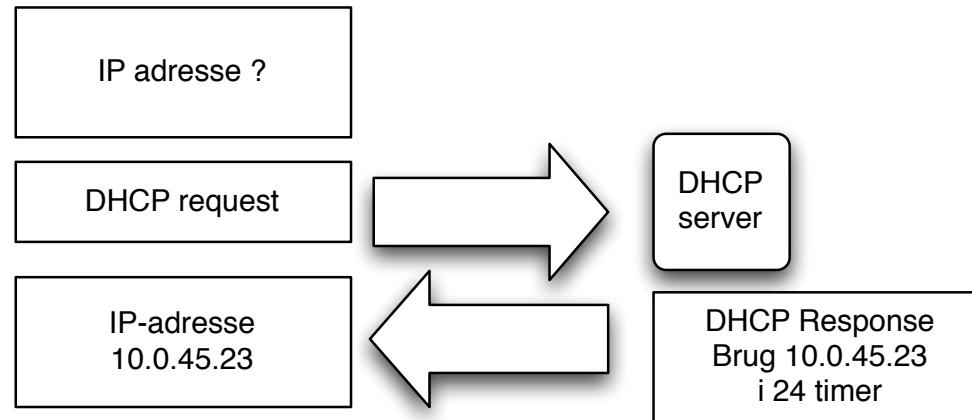


IP routing er nemt

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*



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

```
/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

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!

```
$ 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/ipconfig

som er øvelse 7 fra øvelseshæftet.



Vi laver nu øvelsen

## Netværksinformation: netstat

som er øvelse **8** fra øvelseshæftet.



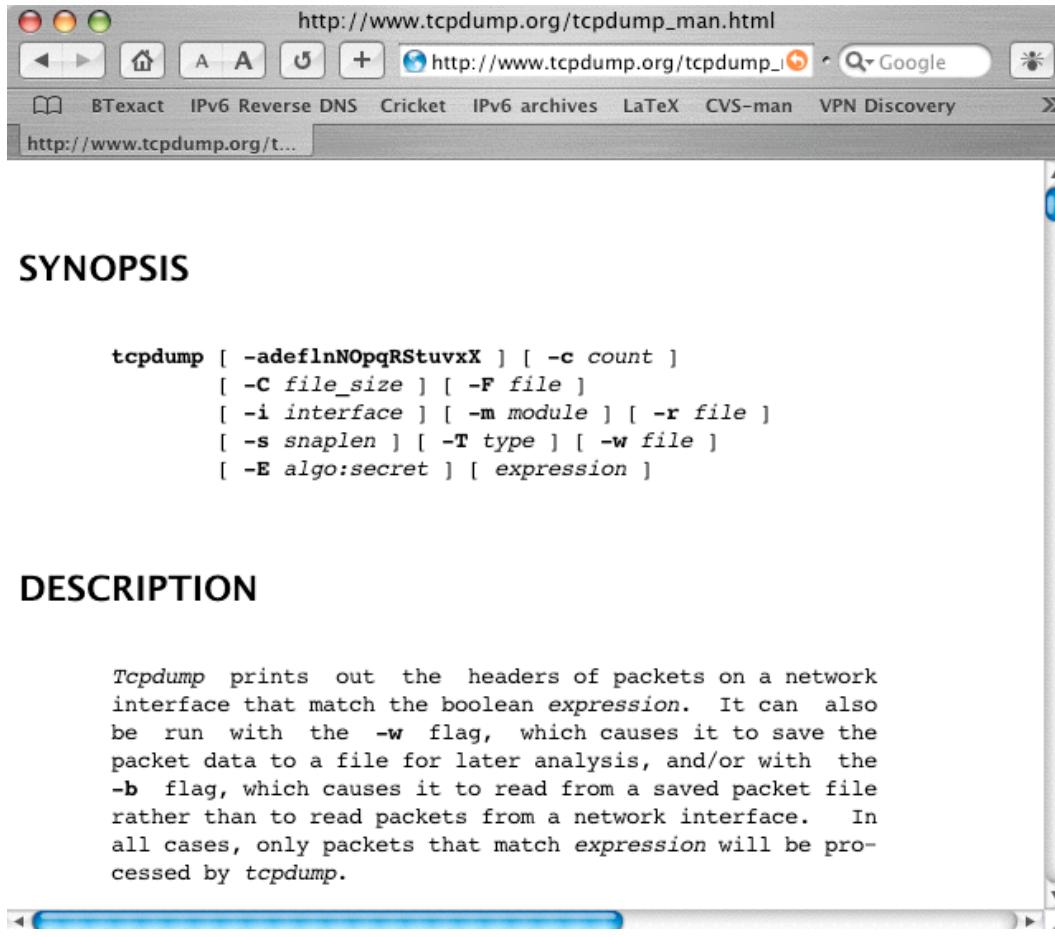
Vi laver nu øvelsen  
**ping og traceroute**  
som er øvelse **9** fra øvelseshæftet.



Vi laver nu øvelsen

## ping6 og traceroute6

som er øvelse **10** fra øvelseshæftet.



The screenshot shows a web browser window with the URL [http://www.tcpdump.org/tcpdump\\_man.html](http://www.tcpdump.org/tcpdump_man.html) in the address bar. The page content is as follows:

## SYNOPSIS

```
tcpdump [ -adeflnNOpqRStuvxX ] [ -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

- 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
...
```

filtrere 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

Host 10.1.2.3

Alle pakker hvor afsender eller modtager er 10.1.2.3

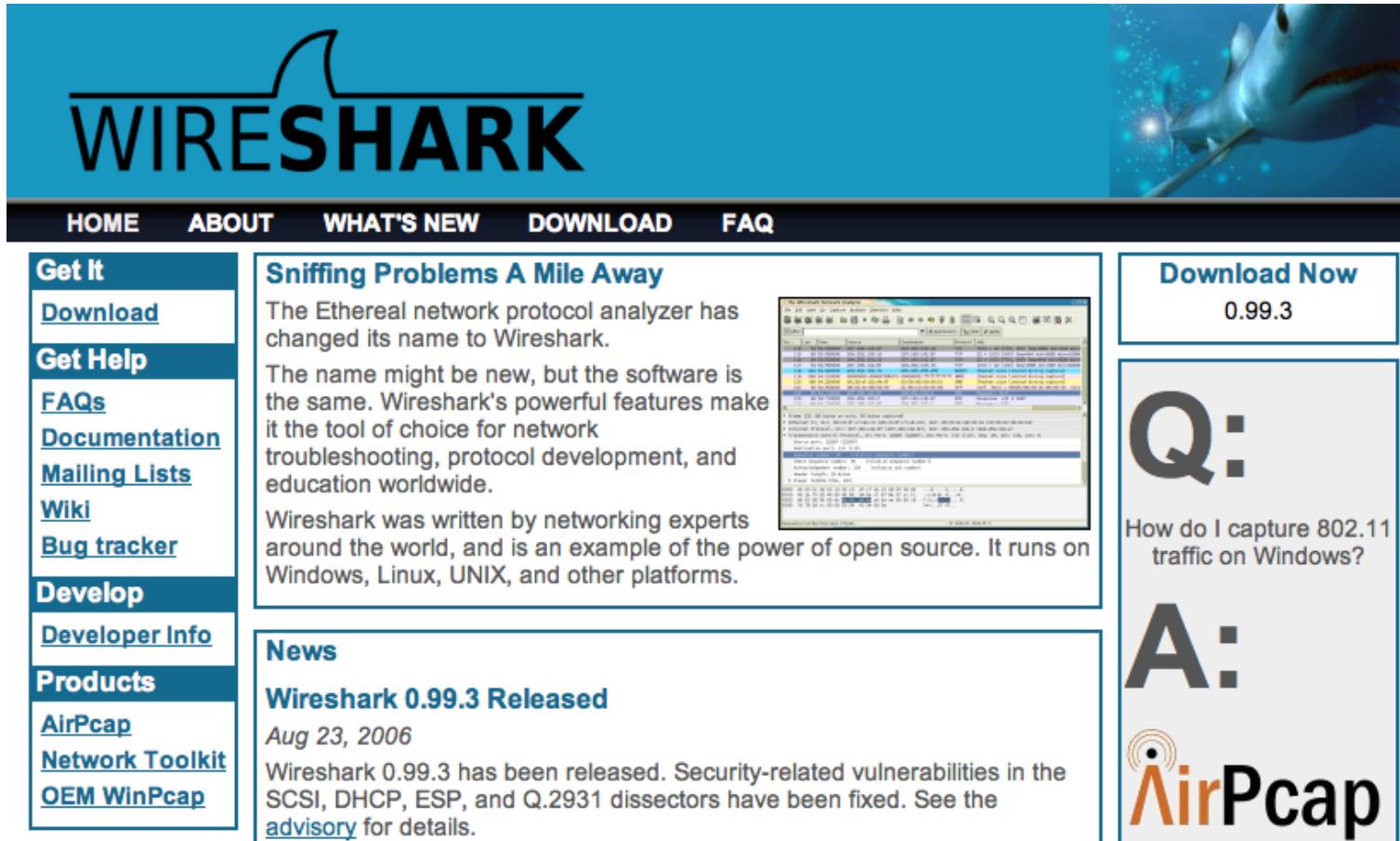
host 10.2.3.4 and not host 10.3.4.5

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

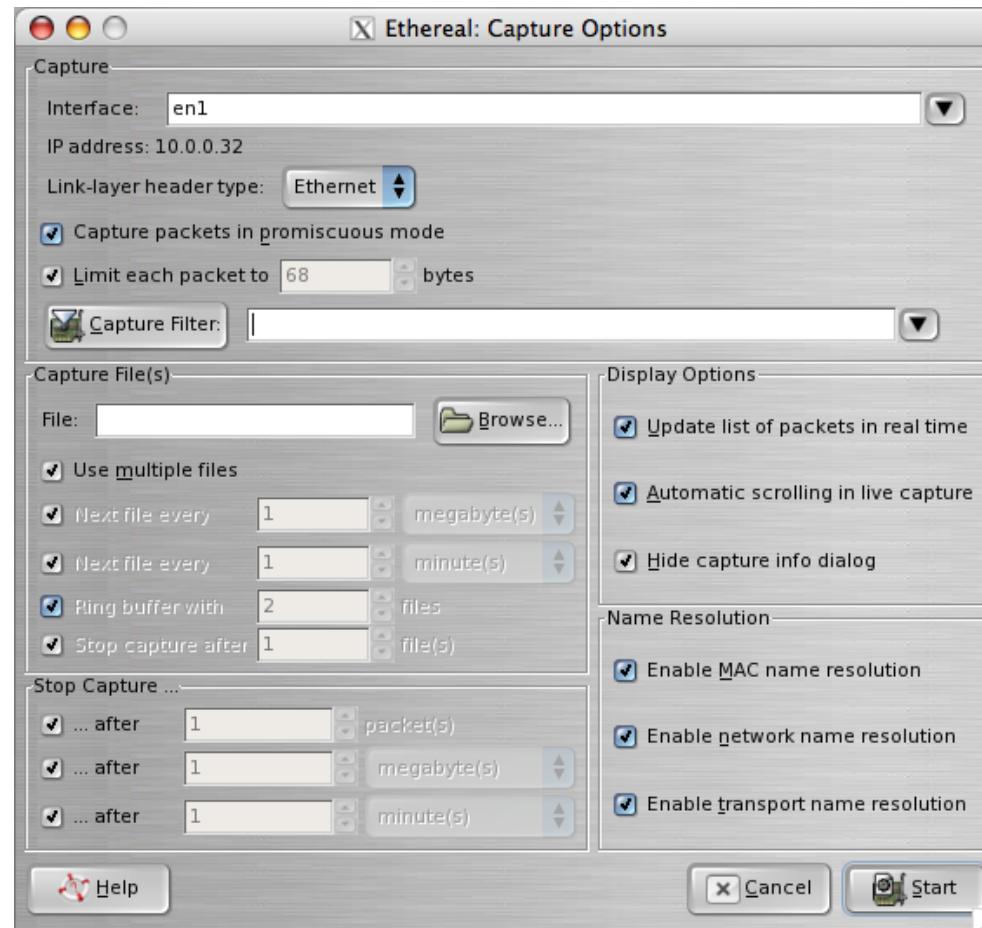
host foo and not port ftp and not port ftp-data

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



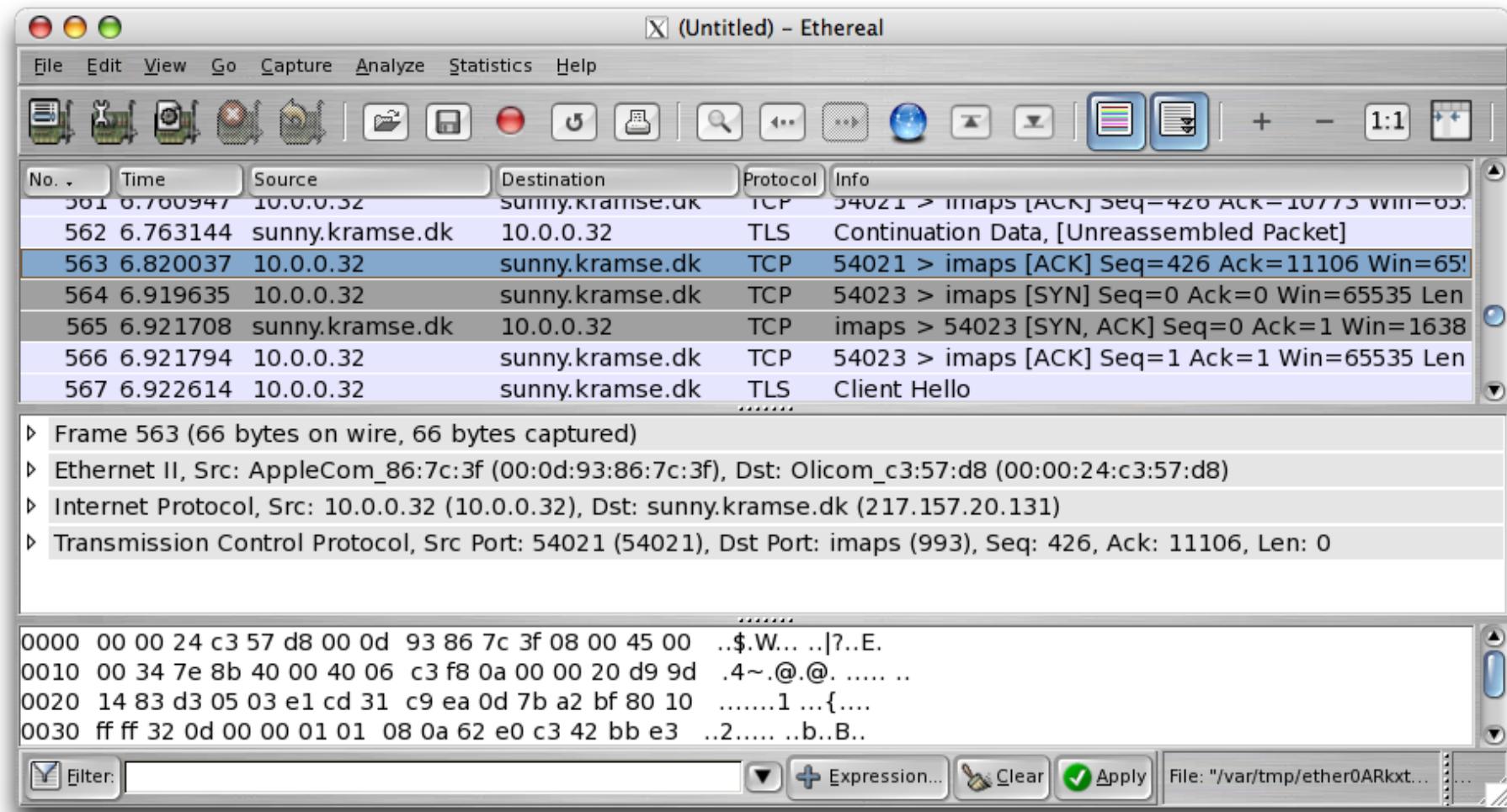
The screenshot shows the official Wireshark website. At the top, there's a large blue header with the "WIRESHARK" logo and a shark swimming in the background. Below the header is a black navigation bar with links: HOME, ABOUT, WHAT'S NEW, DOWNLOAD, and FAQ. On the left, there's a sidebar with sections for "Get It" (links to Download, Get Help, FAQs, Documentation, Mailing Lists, Wiki, Bug tracker), "Develop" (links to Developer Info, Products, AirPcap, Network Toolkit, OEM WinPcap), and "News". The main content area has several boxes: one about sniffing problems, another showing a screenshot of the Wireshark interface, and a "Download Now" section for version 0.99.3. There's also a Q&A section about capturing 802.11 traffic.

<http://www.wireshark.org>  
både til Windows og Unix, tidligere kendt som Ethereal



Man starter med Capture - Options

# Brug af Wireshark



Læg mærke til filtermulighederne



Vi laver nu øvelsen

## Wireshark netværksniffer

som er øvelse **11** fra øvelseshæftet.

en sniffer til mange usikre protokoller

inkluderer **arpspoof**

Lavet af Dug Song, [dugsong@monkey.org](mailto:dugsong@monkey.org)

dsniff is a password sniffer which handles FTP, Telnet, SMTP, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP, MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL\*Net, Sybase and Microsoft SQL protocols.

# Kommenteret dsniff

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

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

an ja
an jnaan ja
an ja
```

# Chaosreader

## Chaosreader Report

Created at: Sun Nov 16 21:04:18 2003, Type: snoop

[Image Report](#) - Click here for a report on captured images.

[GET/POST Report](#) (Empty) - Click here for a report on HTTP GETs and POSTs.

[HTTP Proxy Log](#) - Click here for a generated proxy style HTTP log.

### TCP/UDP/... Sessions

1.	Sun Nov 16 20:38:22 2003	30 s	192.168.1.3:1368 <-> 192.77.84.99:80	web	383 bytes	• <a href="#">as html</a>
2.	Sun Nov 16 20:38:22 2003	29 s	192.168.1.3:1366 <-> 192.77.84.99:80	web	381 bytes	• <a href="#">as html</a>

Med adgang til et netværksdump kan man læse det med chaosreader

Output er HTML med oversigter over sessioner, billeder fra datastrømmen osv.

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

## Kryptering af e-mail

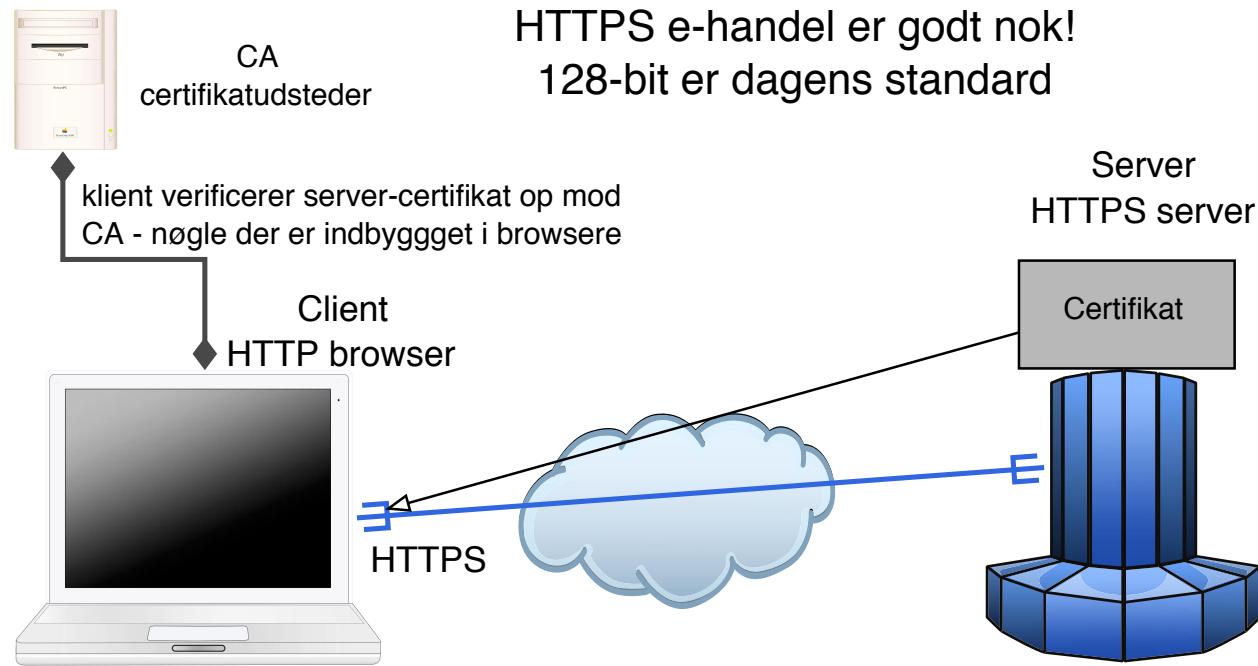
- Pretty Good Privacy - Phil Zimmermann
- GNU Privacy Guard - Open Source implementation af OpenPGP
- OpenPGP = mail sikkerhed, OpenPGP RFC-2440, PGP/MIME RFC 3156)

## 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.

## Kryptering af netværkstrafik - Virtual Private Networks VPN

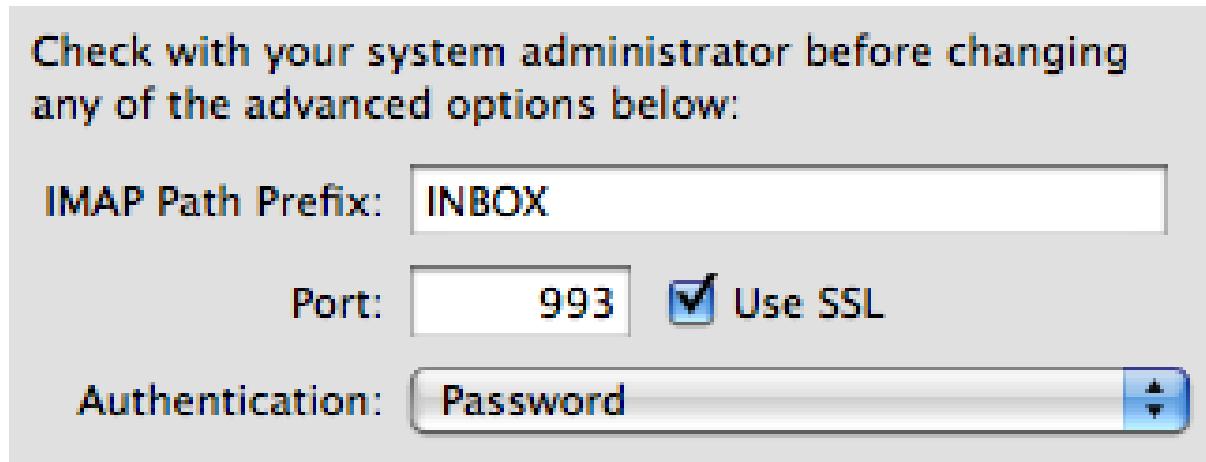
- IPsec IP Security Framework, se også L2TP
- PPTP Point to Point Tunneling Protocol - dårlig og usikker, brug den ikke mere!
- OpenVPN m.fl.



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



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

kommandoerne er:

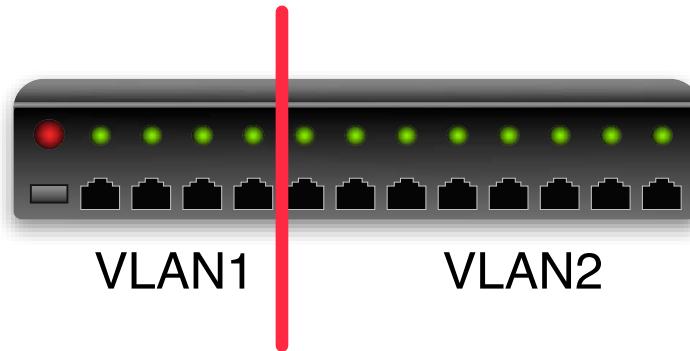
- ssh - Secure Shell
- scp - Secure Copy
- sftp - secure FTP

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!**

## Portbased VLAN



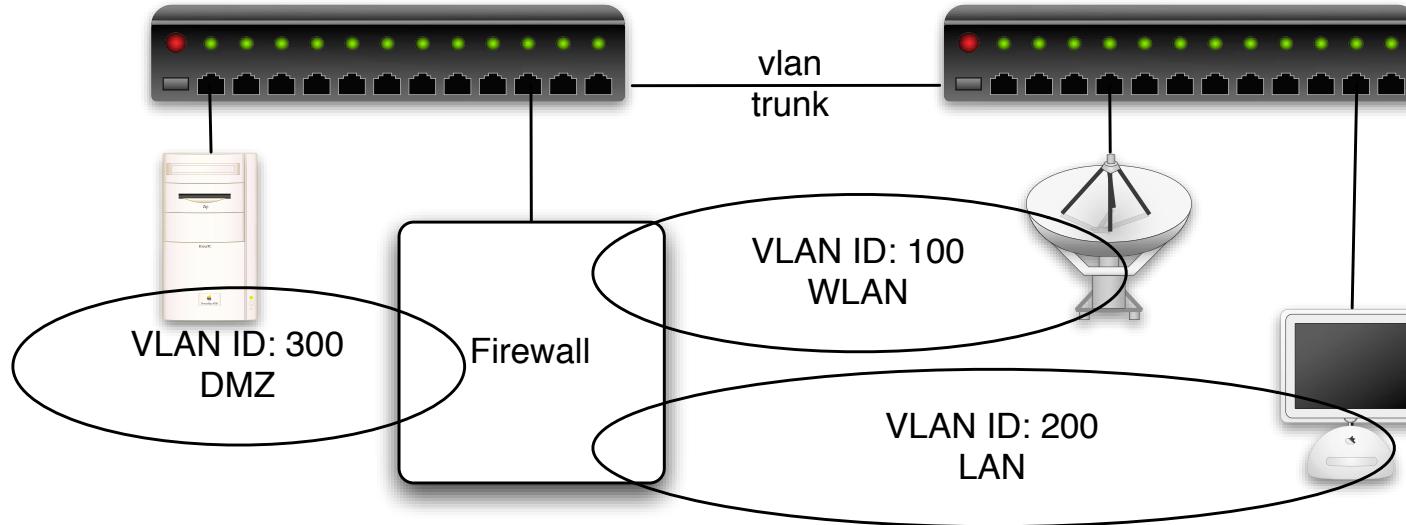
Nogle switcher tillader at man opdeler portene

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 konfiguration med 802.1q VLAN tagging på Ethernet niveau

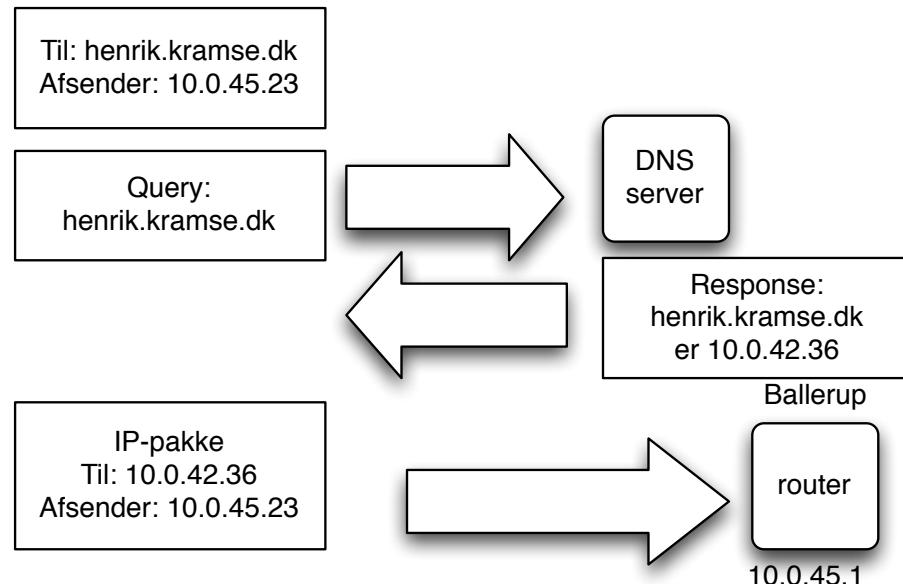
Data skal omkring en firewall eller en router for at krydse fra VLAN1 til VLAN2

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

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



Vi laver nu øvelsen  
**VLAN 802.1q**  
som er øvelse **12** fra øvelseshæftet.



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

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 ....

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

/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
```

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;   };
...
}
```



Vi laver nu øvelsen

## DNS og navneopslag

som er øvelse **13** fra øvelseshæftet.



Vi laver nu øvelsen

## DNS og navneopslag - IPv6

som er øvelse **14** fra øvelseshæftet.

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:

## 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

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
- AfriNIC The Registry of Internet Number Resources for Africa

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

## 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/>

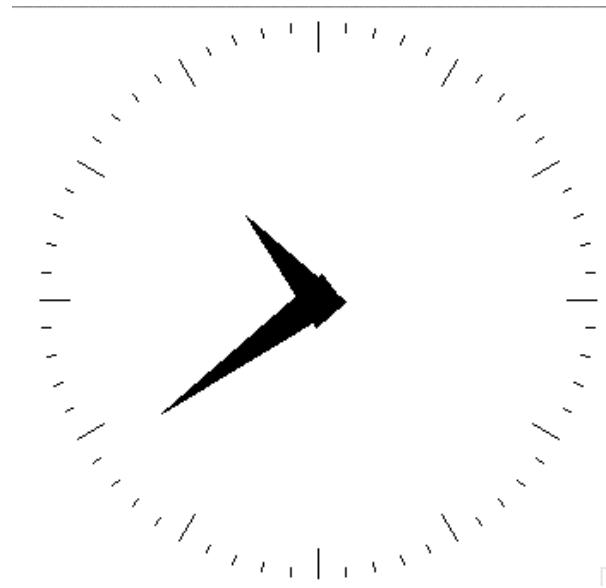
Eksempelvis:

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

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

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

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



Hvad er klokken?

Hvad betydning har det for sikkerheden?

Brug NTP Network Time Protocol på produktionssystemer

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



Vi laver nu øvelsen

## Opslag i whois databaser

som er øvelse **15** fra øvelseshæftet.



Vi laver nu øvelsen

## Ekstraopgave: ICMP tool icmpush

som er øvelse **16** fra øvelseshæftet.



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

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

```
# 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.semmax=256
kern.seminfo.semns=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 *thrash*

brug top og andre statistikprogrammer til at se disse data

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

```
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

Vi prøver lige Iperf sammen

hvis alle prøver samtidig giver det stor variation i resultaterne

Husk også Jperf Java GUI program til Iperf ☺

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

```
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)

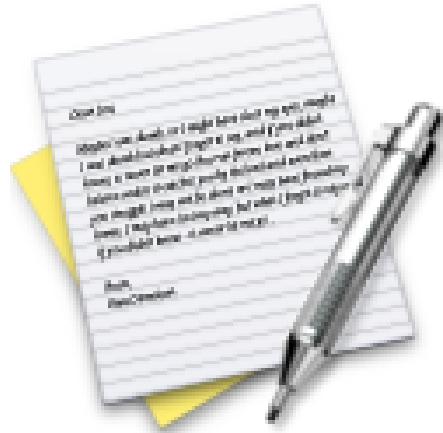
50%	124
66%	126
75%	128
80%	130
90%	143
95%	153
98%	189
99%	446
100%	446 (longest request)



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

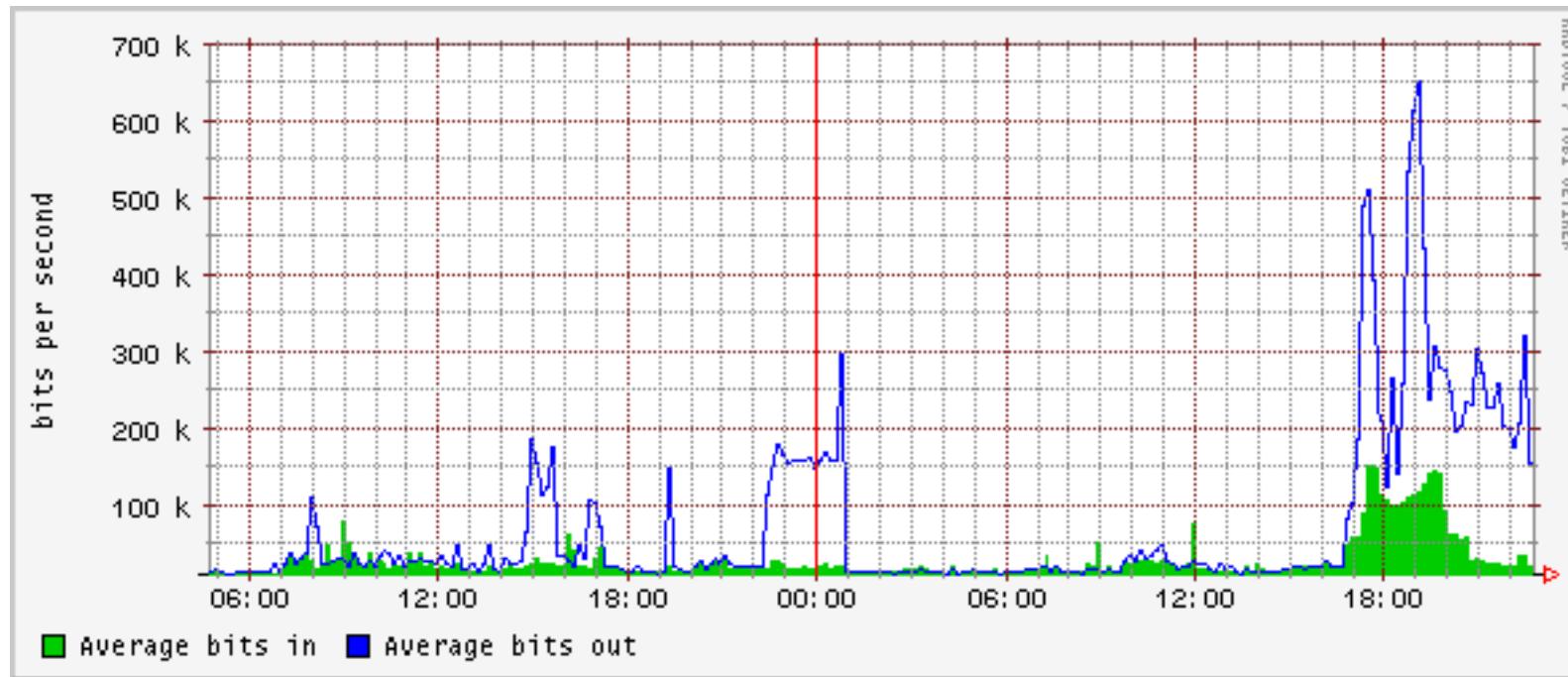
## Afprøv Apache Benchmark programmet

som er øvelse **19** fra øvelseshæftet.

Hvad har vi lært?

Fri leg på udstyret

# Agenda - dag 2 Avancerede netværksteknologier og 802.11



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

På Unix:

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

Windows:

- kontrolpanelet
- ipconfig/ipv6

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

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.

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

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:

<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
```

der findes andre syslog systemer eksempelvis syslog-ng

konfigureres gennem /etc/syslog-ng/syslog-ng.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 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

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

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

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

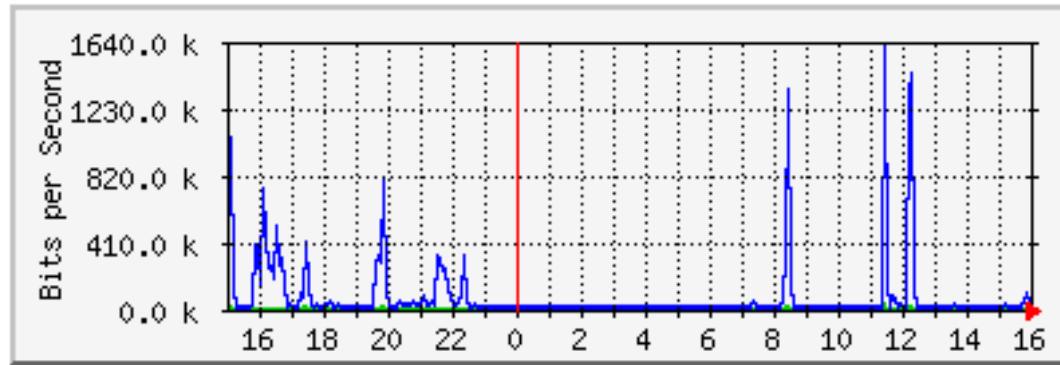
## SNMP walk

som er øvelse **20** fra øvelseshæftet.

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)



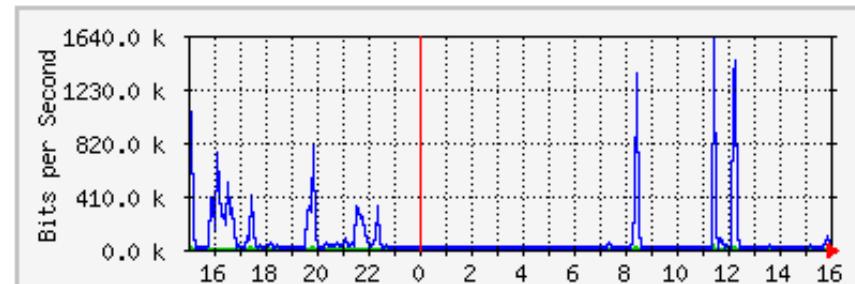
	Max	Middel	Nu
Ind	35.5 kb/s (0.0%)	2392.0 b/s (0.0%)	5280.0 b/s (0.0%)
Ud	1604.6 kb/s (1.6%)	57.6 kb/s (0.1%)	51.4 kb/s (0.1%)

Monitorering af SNMP enheder og grafer

Inkluderer en nem configmaker og benytter idag RRDTool til data

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

## '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%)

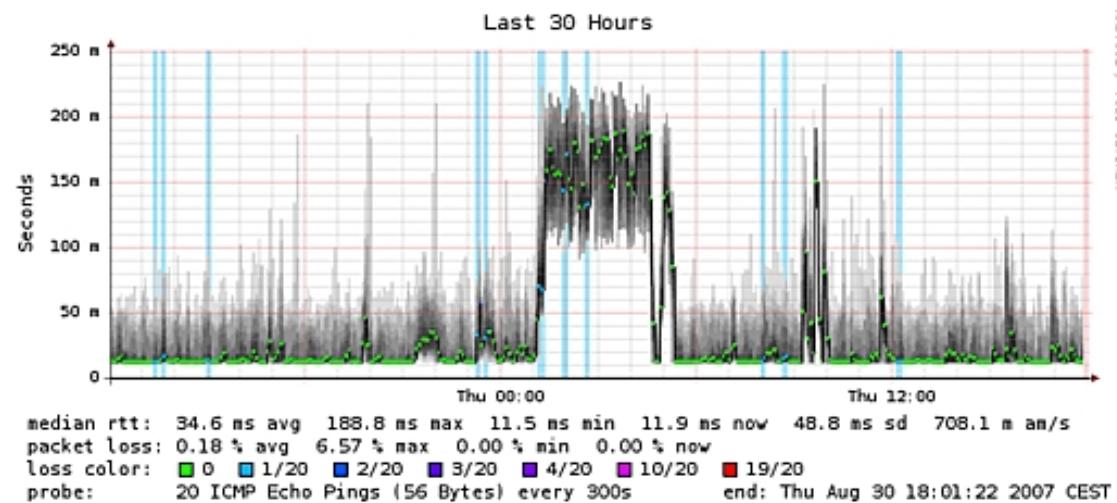
Round Robin Database Tool er en måde at gemme data på

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/>

Brug lidt tid på at se på vores netværk

Valgfrit om I vil se på Administrationsinterface på switcher

eller SNMP indstillinger eksempelvis

eller Nagios og SmokePing på mine servere

# 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>



Vi laver nu øvelsen  
**BIND version**  
som er øvelse **22** fra øvelseshæftet.

802.11 er arbejdsgruppen under IEEE

De mest kendte standarder idag indenfor trådløse teknologier:

- 802.11b 11Mbps versionen
- 802.11g 54Mbps versionen
- 802.11n endnu hurtigere, og draft
- 802.11i Security enhancements

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>

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

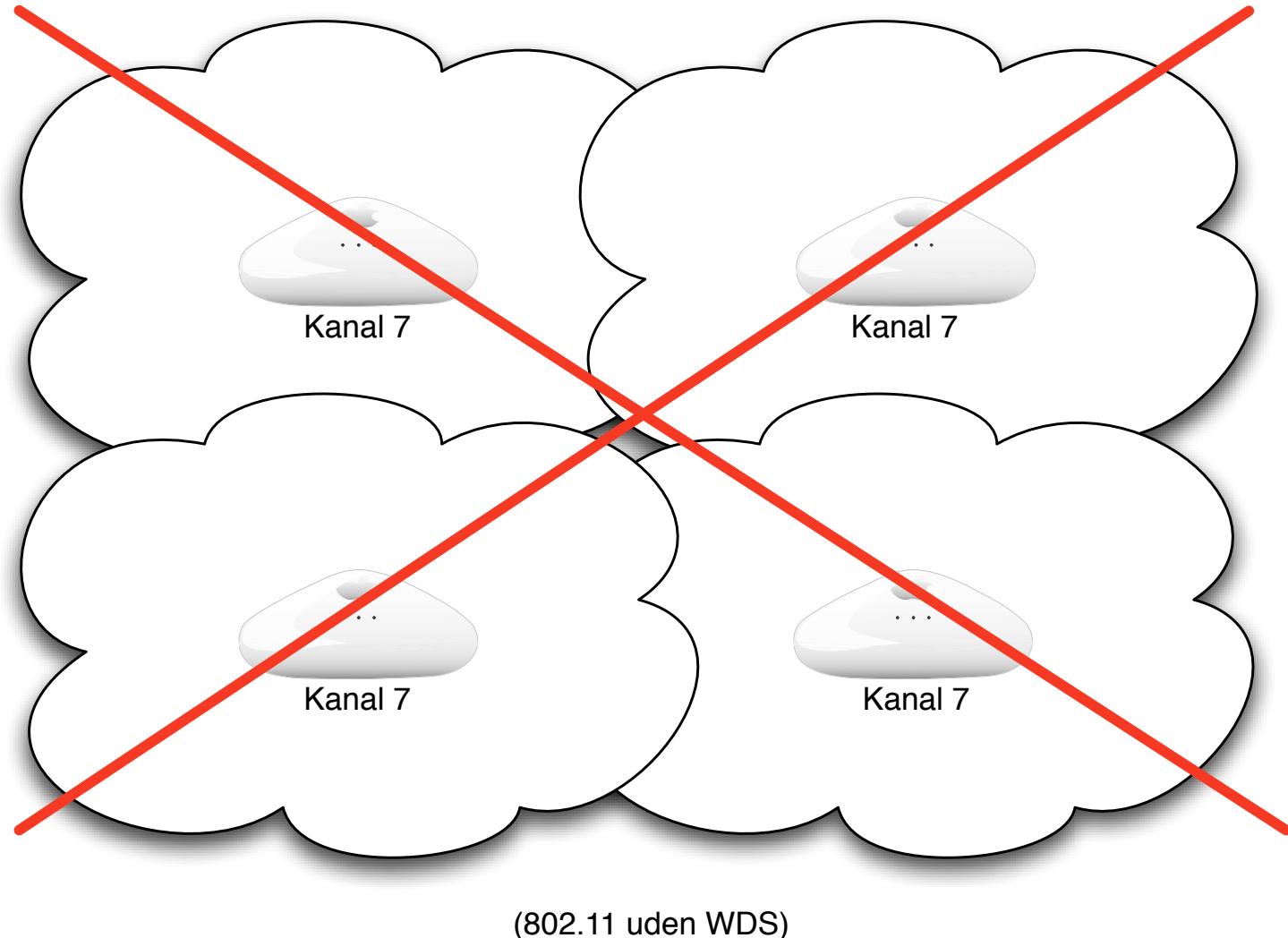
Alternativt kan wireless kort oprette ad-hoc netværk - hvor trafikken går direkte mellem netkort

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

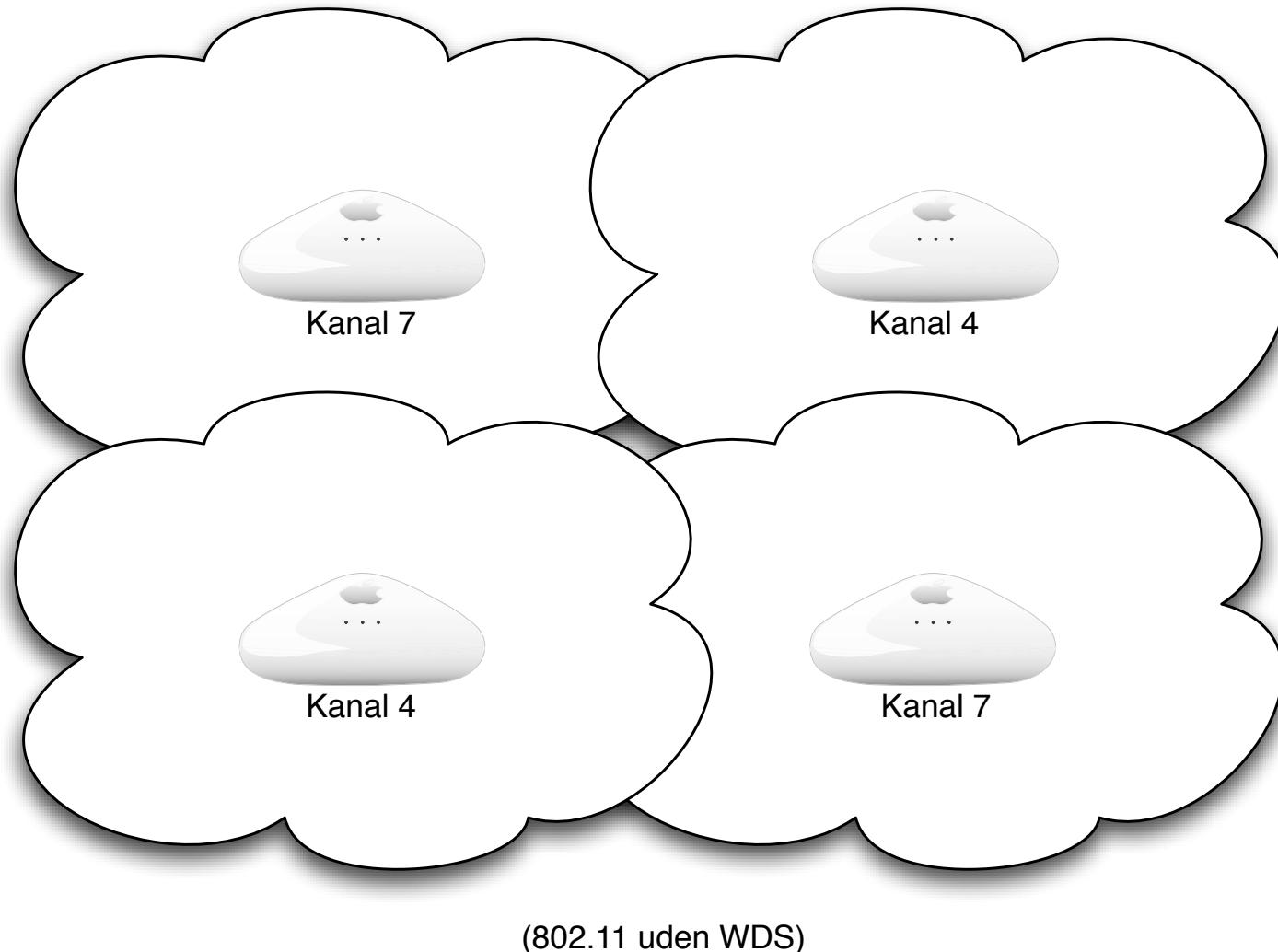
Højest 2 kanaler spring for 802.11b AP der placeres indenfor rækkevidde

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

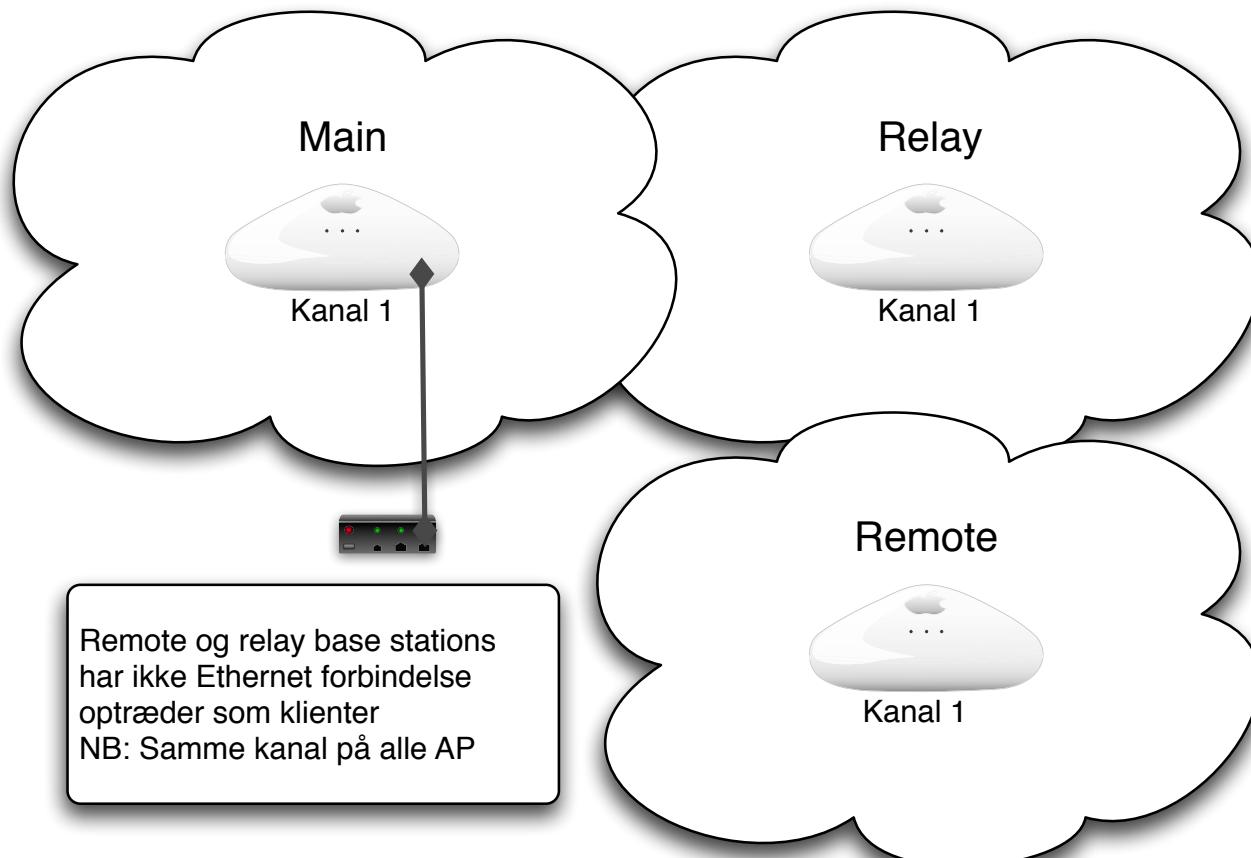
# Eksempel på netværk med flere AP'er



# Eksempel på netværk med flere AP'er



# Wireless Distribution System WDS



(802.11 med WDS)

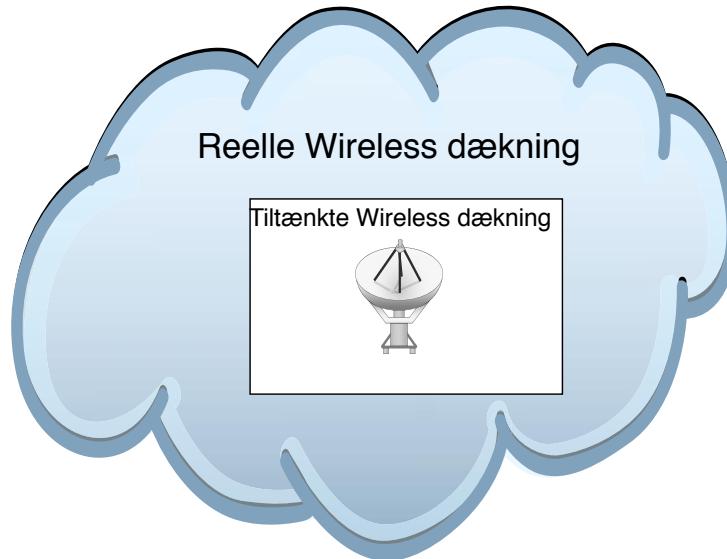
Se også: [http://en.wikipedia.org/wiki/Wireless\\_Distribution\\_System](http://en.wikipedia.org/wiki/Wireless_Distribution_System)

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!

Laptop gerne med PC-CARD slot

Trådløse kort Atheros chipset anbefales, de indbyggede er til tider ringe ;-)

Access Points - jeg anbefaler Airport Express

Antenner hvis man har lyst

Bøger: *Real 802.11 security*

Internetressourcer:

- BackTrack - 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>
- Aircrack-ng suite af programmer <http://www.aircrack-ng.org/>



## Wireless Access Point



**netværket - typisk Ethernet**

et access point - forbides til netværket

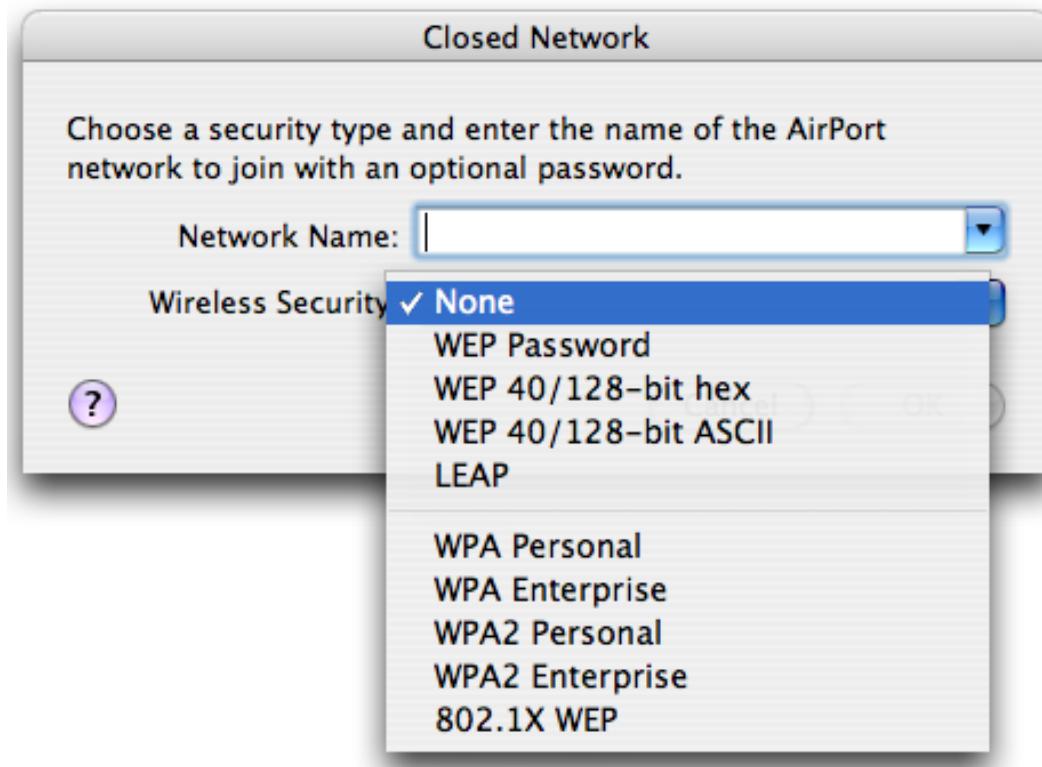
Når man tager fat på udstyr til trådløse netværk opdager man:

SSID - nettet skal have et navn

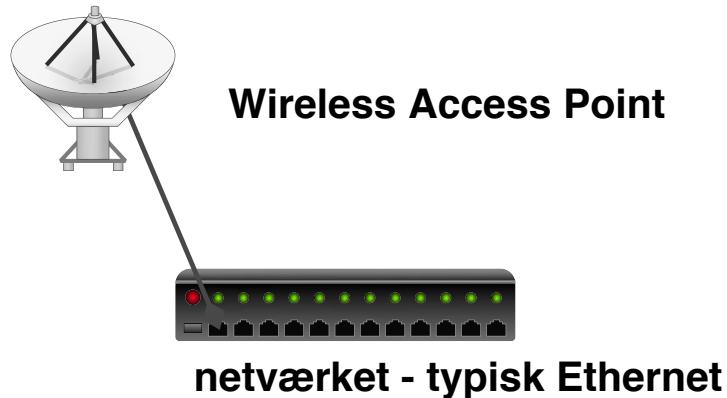
frekvens / kanal - man skal vælge en kanal, eller udstyret vælger en automatisk  
der er nogle forskellige metoder til sikkerhed



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



- 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

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

Service Set Identifier (SSID) - netnavnet

32 ASCII tegn eller 64 hexadecimale cifre

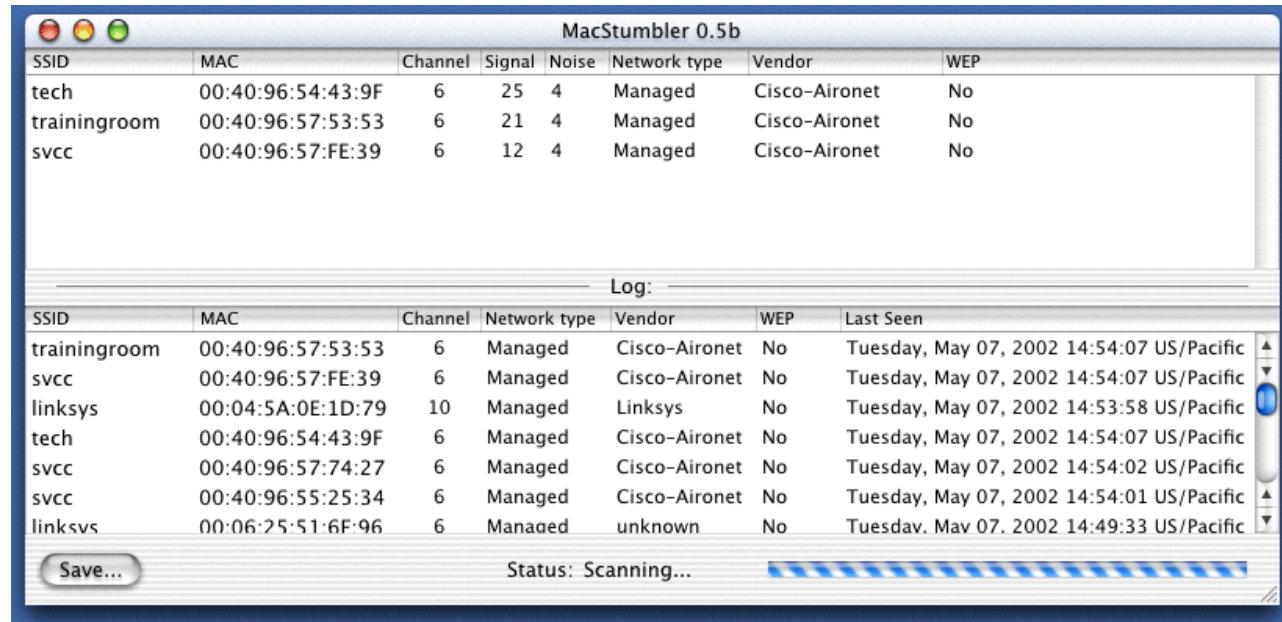
Udstyr leveres typisk med et standard netnavn

- 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

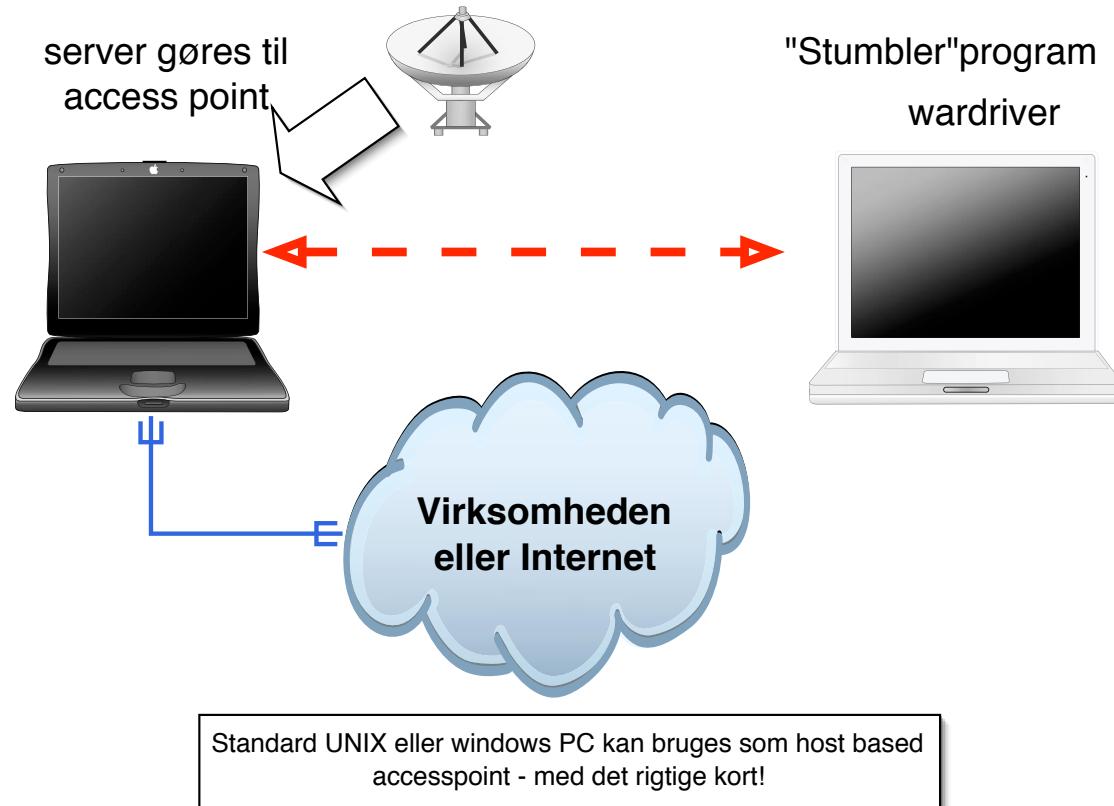
# Demo: wardriving med stumbler programmer



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

- Netstumbler - Windows <http://www.netstumbler.com>
- dstumbler - UNIX <http://www.dachboden.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*

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?

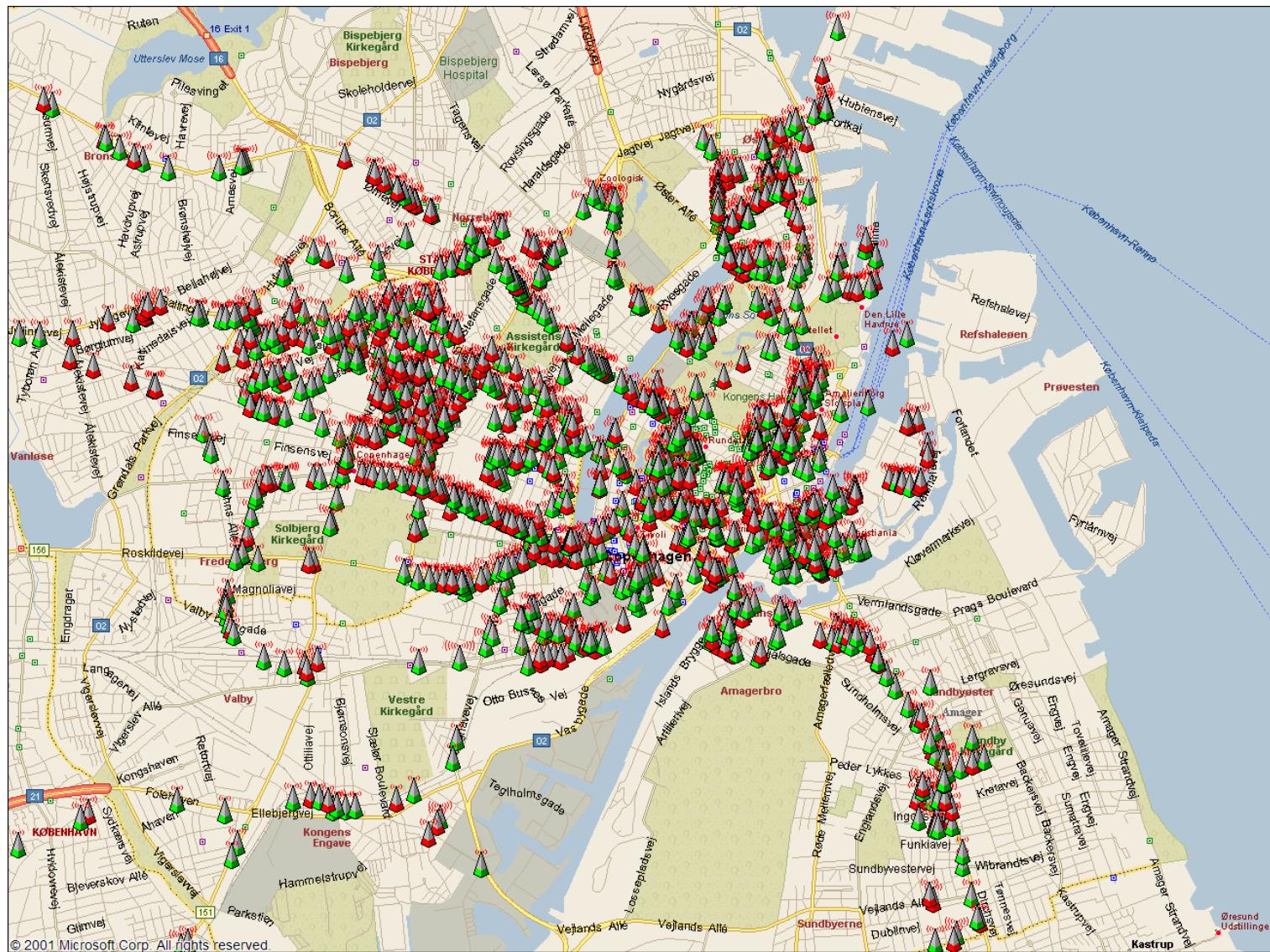
MAC adressen kan derfor overtages, når en af de tilladte stationer forlader området ...

## 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.





Vi laver nu øvelsen

## Wardriving på Windows - inSSIDer

som er øvelse **24** fra øvelseshæftet.



Vi laver nu øvelsen

## Wardriving på Unix - Kismet

som er øvelse **25** fra øvelseshæftet.

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* - 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

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



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/>

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 integritetscheck 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**



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. 100.000 er godt

Links:

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

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

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 -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 ]**

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



Vi laver nu øvelsen

## Ekstraopgave: Airodump-ng lavniveau sniffer

som er øvelse **26** fra øvelseshæftet.

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.

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 som kræver CCMP

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)

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

Personal - en delt hemmelighed, preshared key kaldes WPA-PSK

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
- CCMP - Counter-Mode/CBC-MAC Protocol, IEEE 802.11i AES kryptering
- CBC-MAC - Cipher Block Chaining - Message Authentication Code
- AES - Advanced Encryption Standard, Rijndael algoritmen
- RSN - Robust Secure Network, en del af IEEE 802.11i

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

Desværre ikke!

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

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

Vi konfigurerer AP med Henrik42 som WPA-PSK/passthase

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-PSK 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

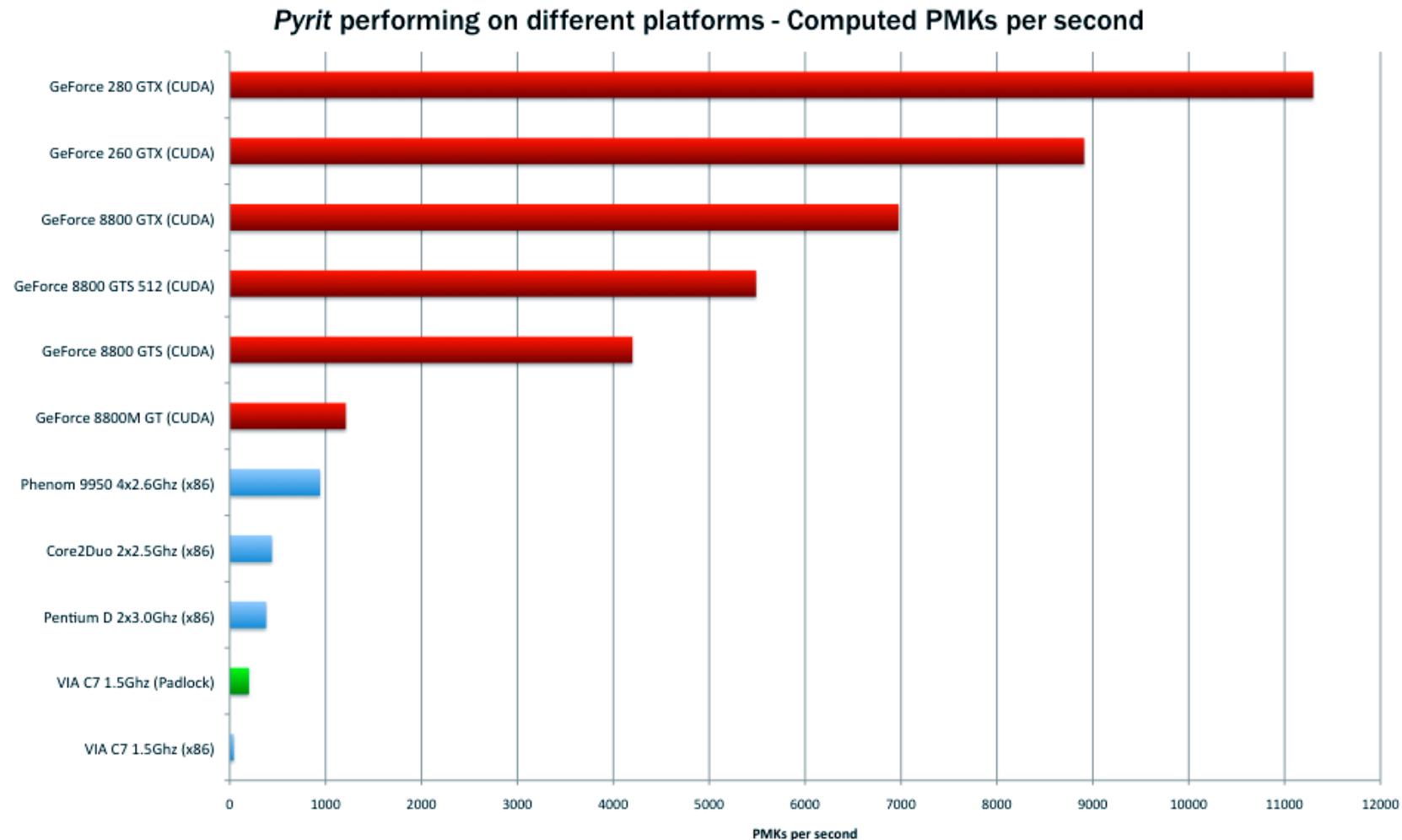
*Pyrit* takes a step ahead in attacking WPA-PSK and WPA2-PSK, the protocol that today de-facto protects public WIFI-airspace. The project's goal is to estimate the real-world security provided by these protocols. Pyrit does not provide binary files or wordlists and does not encourage anyone to participate or engage in any harmful activity. **This is a research project, not a cracking tool.**

*Pyrit*'s implementation allows to create massive databases, pre-computing part of the WPA/WPA2-PSK authentication phase in a space-time-tradeoff. The performance gain for real-world-attacks is in the range of three orders of magnitude which urges for re-consideration of the protocol's security. Exploiting the computational power of GPUs, *Pyrit* is currently by far the most powerful attack against one of the world's most used security-protocols.

sloooow, plejede det at være - 150 keys/s på min Thinkpad X31

Kryptering afhænger af SSID! Så check i tabellen er minutter.

<http://pyrit.wordpress.com/about/>

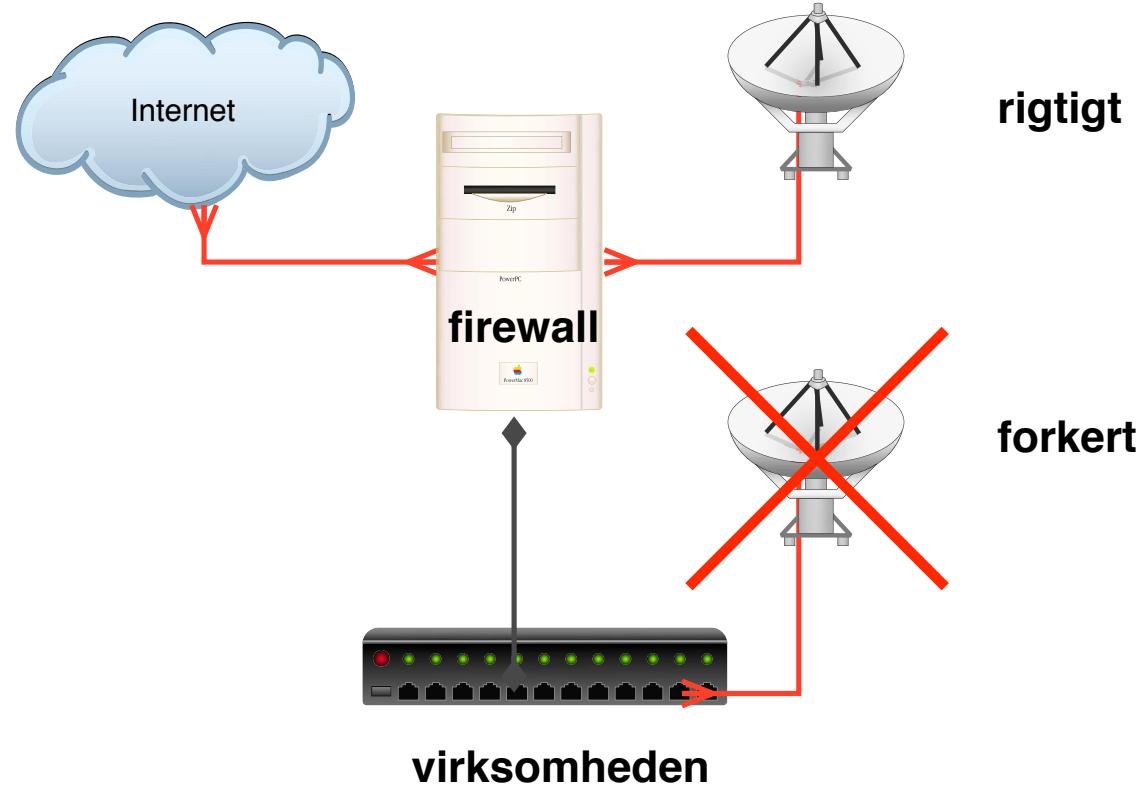


Kilde: <http://code.google.com/p/pyrit/>

Så går man igang med de almindelige værktøjer

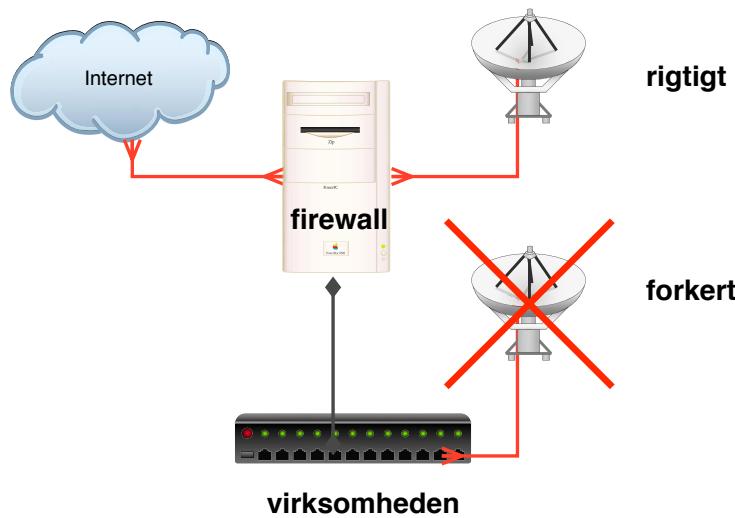
Fyodor Top 100 Network Security Tools <http://www.sectools.org>

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

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

LDAP er en protokol til directory access, opslag i database

Light udgave af den X.500 directory access standard

LDAP er beskrevet i RFC-3377: Lightweight Directory Access Protocol (v3): Technical Specification

Standard interface for opslag, men desværre ikke for data

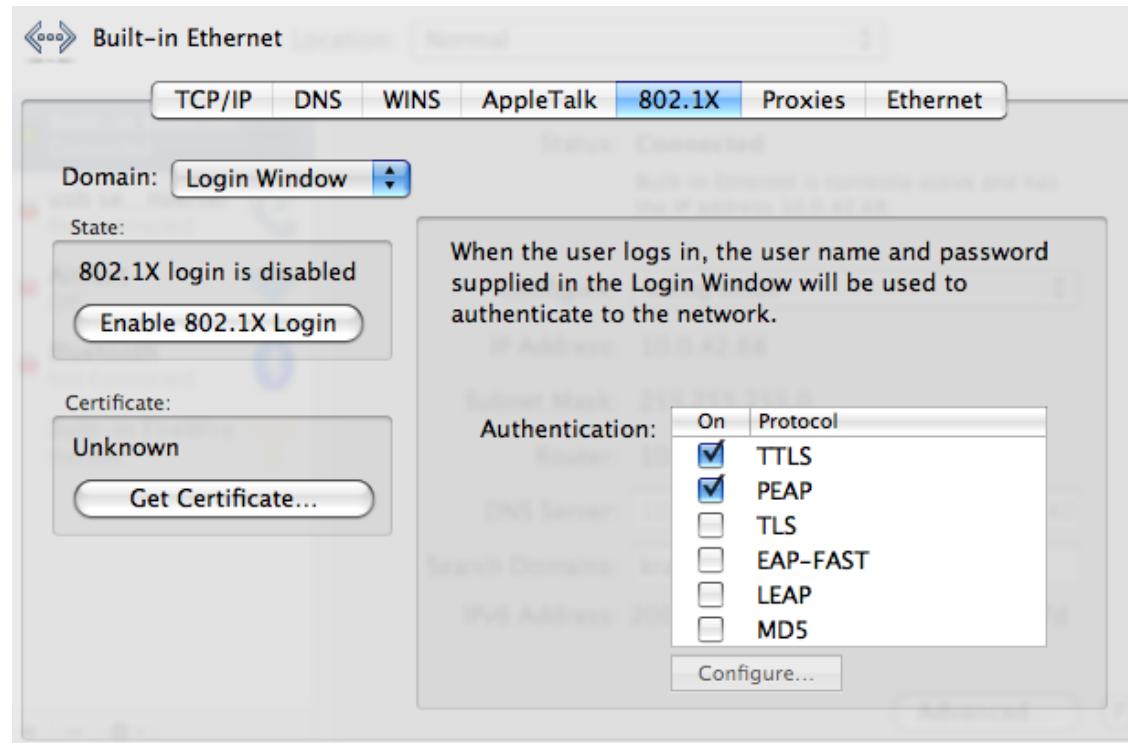
Bruges meget typisk til brugere, grupper og passwords



Vi laver nu øvelsen  
**RADIUS client**  
som er øvelse **27** fra øvelseshæftet.



Vi laver nu øvelsen  
**LDAP client**  
som er øvelse **28** fra øvelseshæftet.



Nogle switcher tillader at man benytter 802.1x brugervalidering på portniveau

Adgang til porten baseret på brugernavn og kodeord/certifikat

Denne protokol indgår også i WPA Enterprise

802.1x giver væsentlige fordele

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

WEP og WPA-PSK er en nøgle til alle brugere, 802.1x er individuel adgang

Typisk benyttes RADIUS integration mod LDAP eller Active Directory

Vi vil nu gennemgå netværksdesign med udgangspunkt i vores setup

Vores setup indeholder:

- Routere
- Firewall
- Wireless
- DMZ
- DHCPD, BIND, BGPD, ...

Den kunne udvides med flere andre teknologier vi har til rådighed:

- VLAN inkl VLAN trunking/distribution
- WPA Enterprise

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/>

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
- Kompatabilitet - hvilke protokoller, codecs, standarder, ...

Der er flere store spillere

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/>

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

FTP bruger TCP men sender også i klartekst: **USER brugernavn** og  
**PASS hemmeligt-kodeord**

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

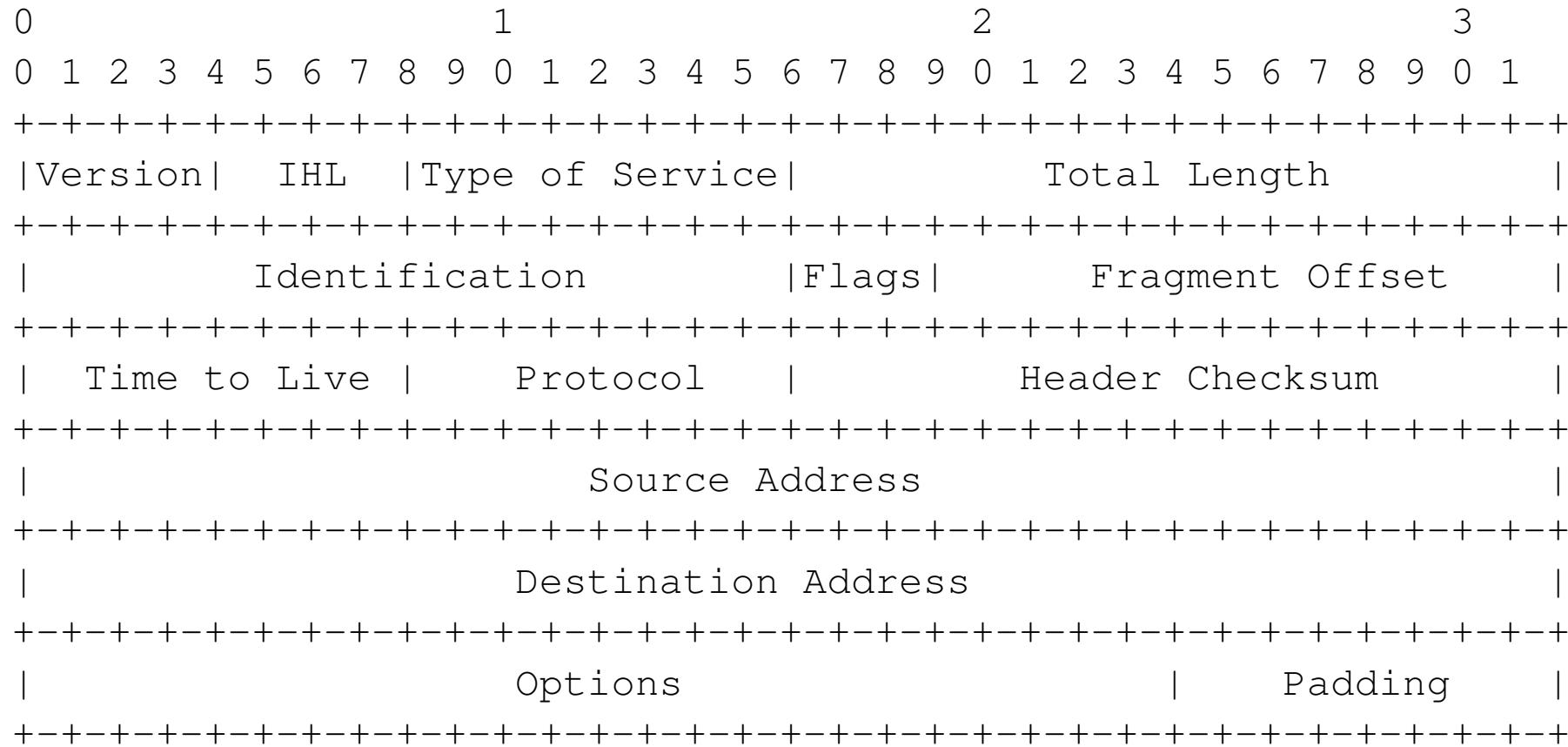
ADSL-Bandwidth-Management-HOWTO, ADSL Bandwidth Management HOWTO

Adv-Routing-HOWTO, Linux Advanced Routing & Traffic Control HOWTO

<http://www.knowplace.org/shaper/resources.html> Linux resources

Det kaldes også traffic shaping

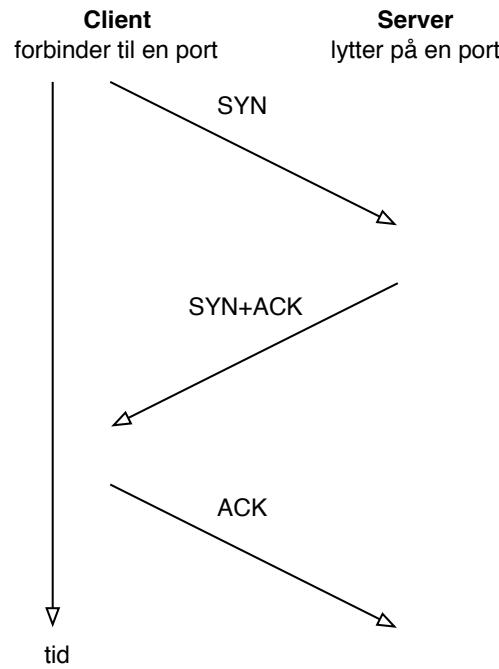
# Firewalls - packet filtering



Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste statefull inspection

# TCP three way handshake



- Hvis en maskine modtager mange SYN pakker kan dette fyldes tabellen over connections op - og derved afholde nye forbindelser fra at blive oprettet - **SYN-flooding**
- Mange firewalls kan udføre SYN handshake idag, før forbindelsen overlades til serveren bagved
- Beskytter mod **TCP SYN flooding**

- 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

- 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 kommersielle 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>
- FreeBSD inkluderer også OpenBSD PF
- Mac OS X benytter IPFW og har en application socket firewall
- NetBSD - bruger IPF og OpenBSD PF

NB: kun eksempler og dem jeg selv har brugt

Man hører indimellem begrebet *hardware firewall*

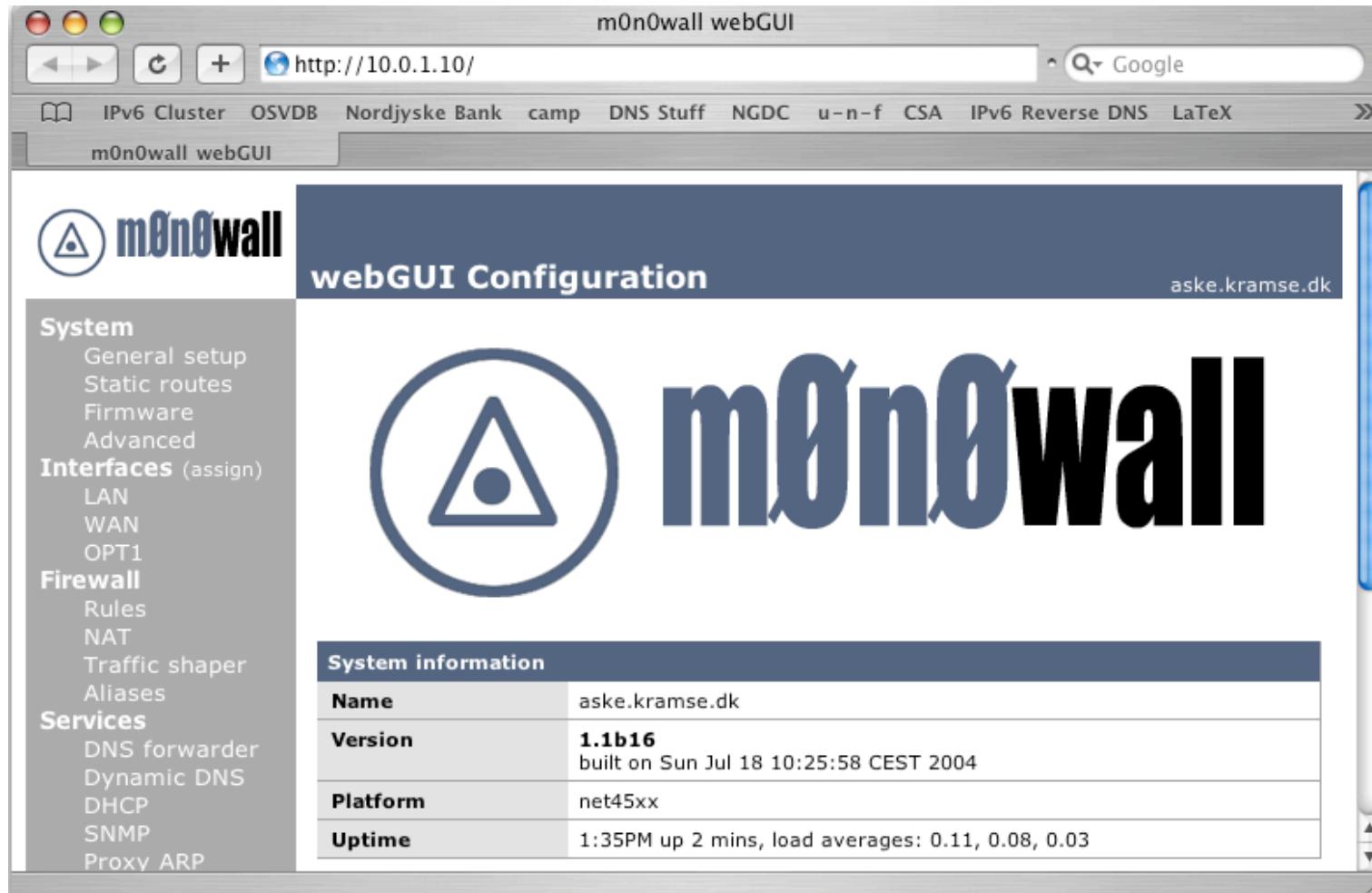
Det er dog et faktum at en firewall består af:

- Netværkskort - som er hardware
- Filtreringsssoftware - 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



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

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

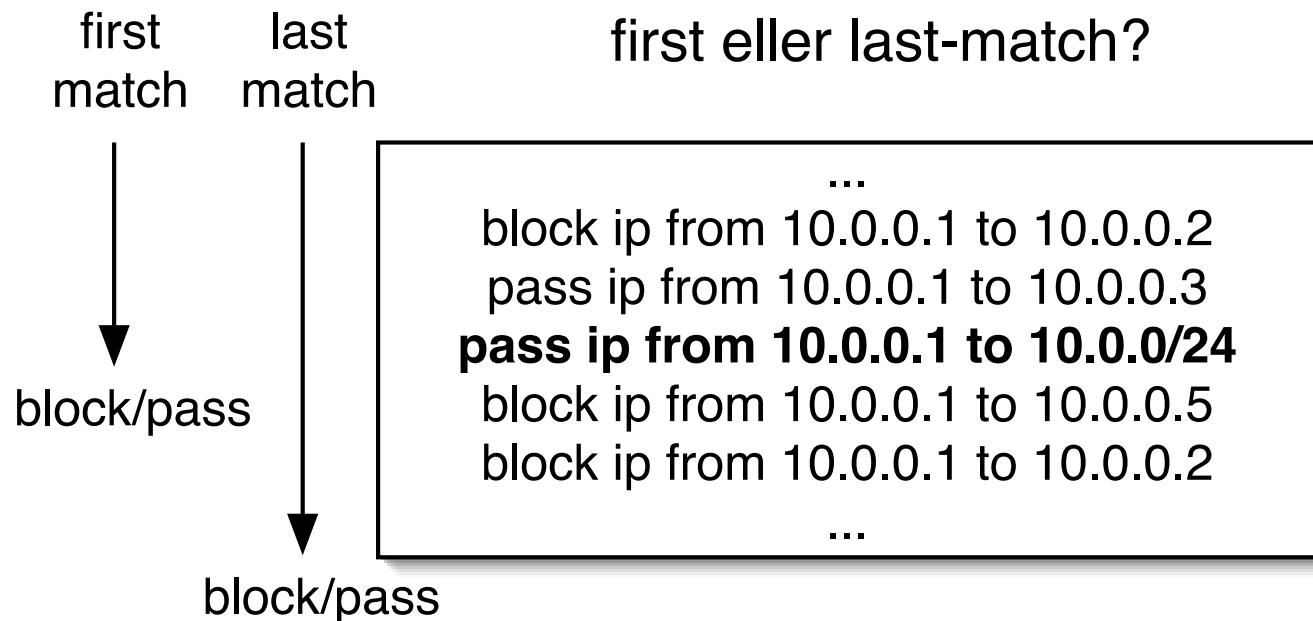
**Det er ekstremt vigtigt at vide hvilken type firewall man bruger!**

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

```
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!

```
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
```

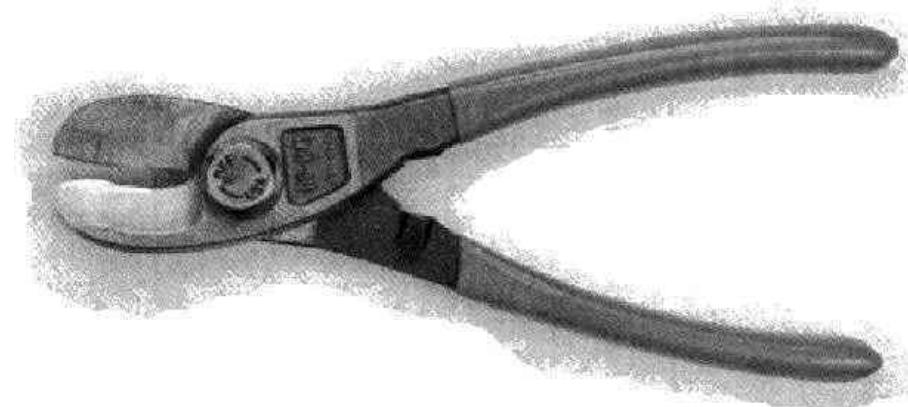
Pakkerne markeres med block eller pass indtil sidste regel  
nøgleordet *quick* afslutter match - god til store regelsæt

```
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



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

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

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

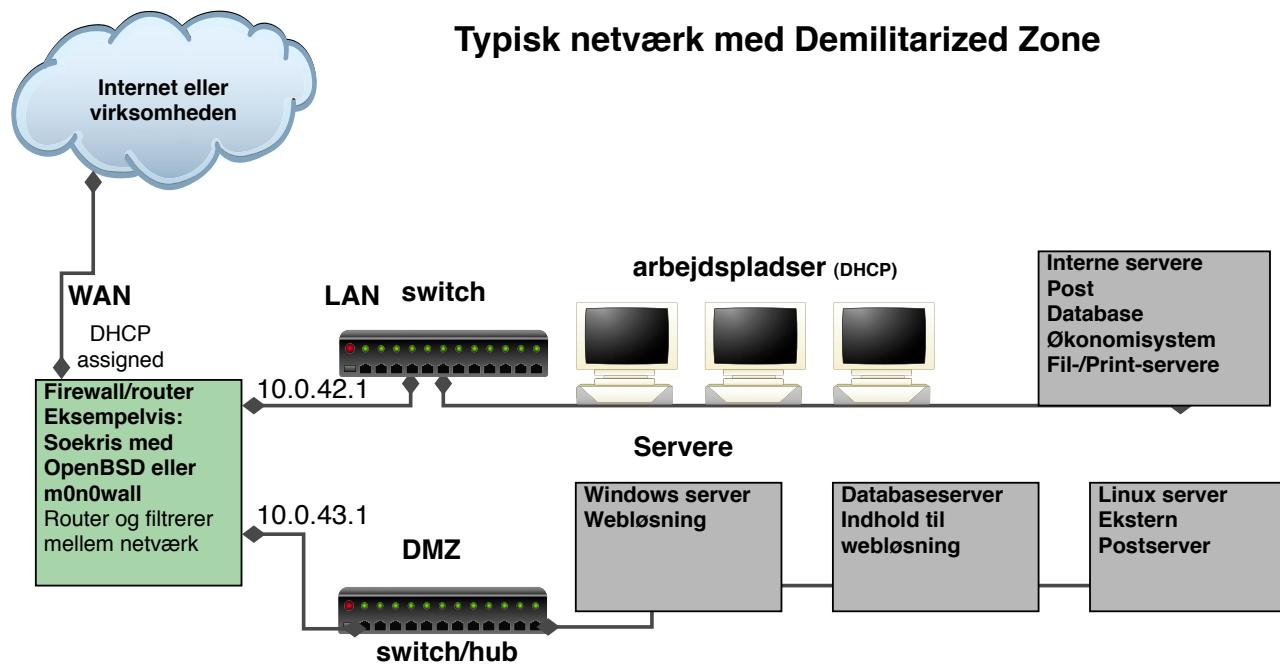
- En firewall skal passes
- En firewall skal opdateres
- Systemerne bagved skal hærdes!

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

# En typisk firewall konfiguration



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

Forskellige computere til forskellige formål, en server er mail-server en anden er web-server

Brug en sikker konfiguration, minimumskonfiguration

Brug sikre protokoller, kryptering SSH fremfor Telnet

Opsætning af netværk, hvordan? Security Configuration Guides + paranoia

- <http://csrc.nist.gov/publications/PubsSPs.html>
- <http://www.nsa.gov/research/publications/index.shtml>
- [http://www.nsa.gov/ia/guidance/security\\_configuration\\_guides/index.shtml](http://www.nsa.gov/ia/guidance/security_configuration_guides/index.shtml)

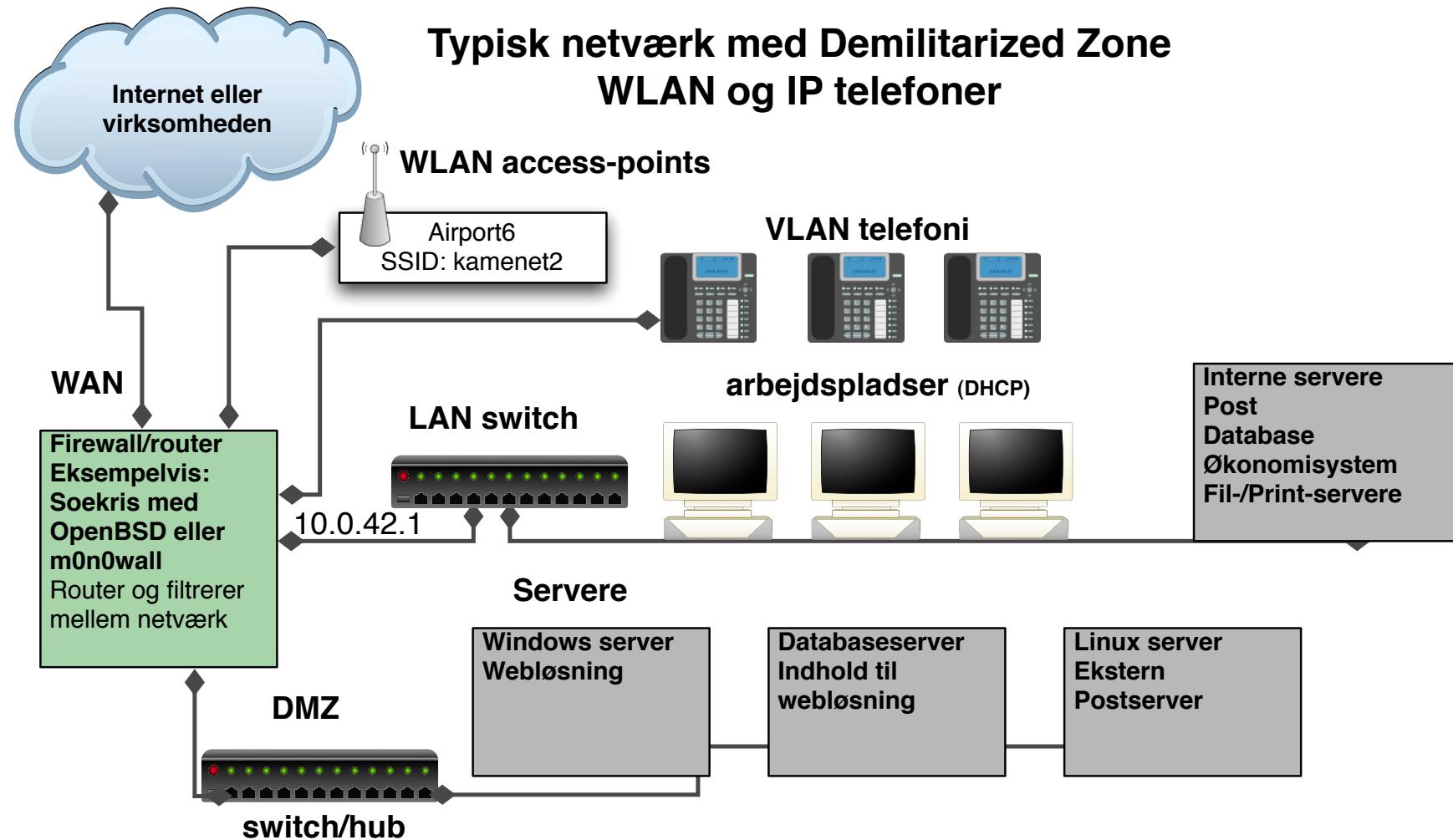
Filtrering på højere niveauer i OSI modellen er muligt

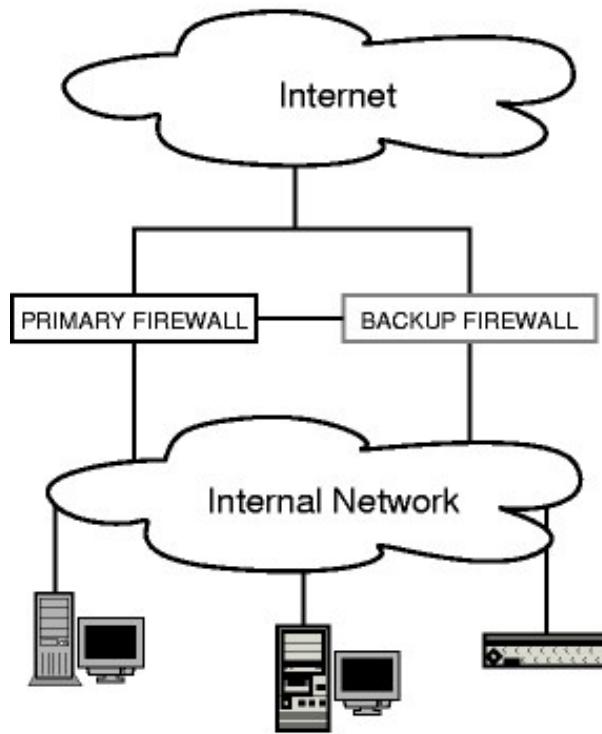
Idag findes proxy applikationer til de mest almindelige funktioner

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





- 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/>



Vi laver nu øvelsen

## Ekstraopgave: Firewallkonfiguration

som er øvelse **29** fra øvelseshæftet.



Vi laver nu øvelsen

## Ekstraopgave: Find maskiner

som er øvelse **30** fra øvelseshæftet.



Vi laver nu øvelsen

## Ekstraopgave: nmap portscanning

som er øvelse **31** fra øvelseshæftet.

De fleste firewalls giver mulighed for at lave krypterede tunneler

Nyttigt til fjernkontorer der skal have usikker traffik 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

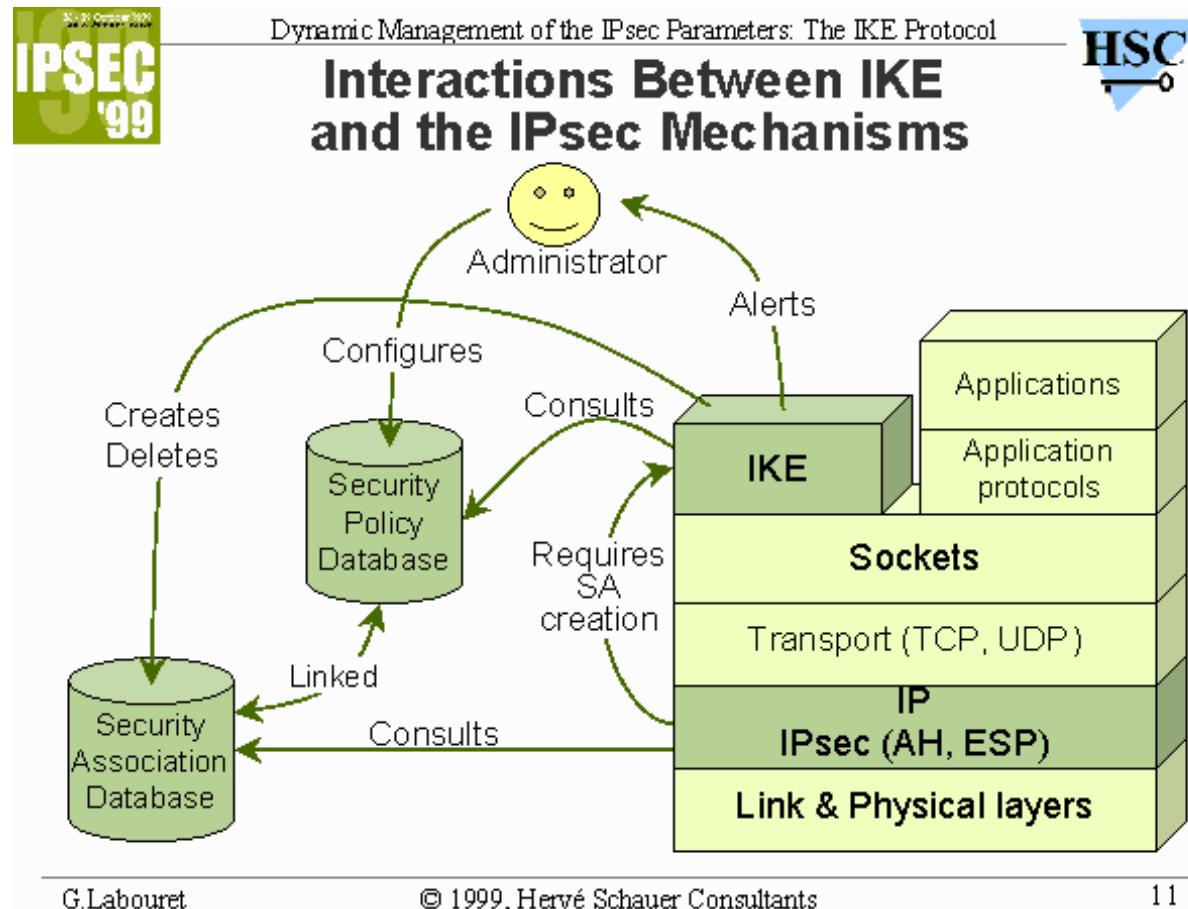
- RFC-2401 Security Architecture for the Internet Protocol
- RFC-2402 IP Authentication Header (AH)
- RFC-2406 IP Encapsulating Security Payload (ESP)
- RFC-2409 The Internet Key Exchange (IKE) - dynamisk keying

Både til IPv4 og IPv6

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

Der findes IKEscan til at scanne efter IKE porte/implementationer

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



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

## Pakkerne før og efter:

BEFORE APPLYING ESP

IPv6		ext	hdrs			
	orig	IP	hdr	if present	TCP	Data

AFTER APPLYING ESP

IPv6	orig	hop-by-hop, dest*,	dest			ESP		ESP
	IP	hdr routing, fragment.	ESP opt*	TCP Data	Trailer Auth			

| <---- encrypted ----> |  
| <---- authenticated ----> |

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/>

Hvad taler for og imod - de næste slides gennemgår nogle standardsetups

En slags Patterns for networking

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!

Hvis der kun skal distribueres filer kan man ofte benytte HTTP istedet for FTP

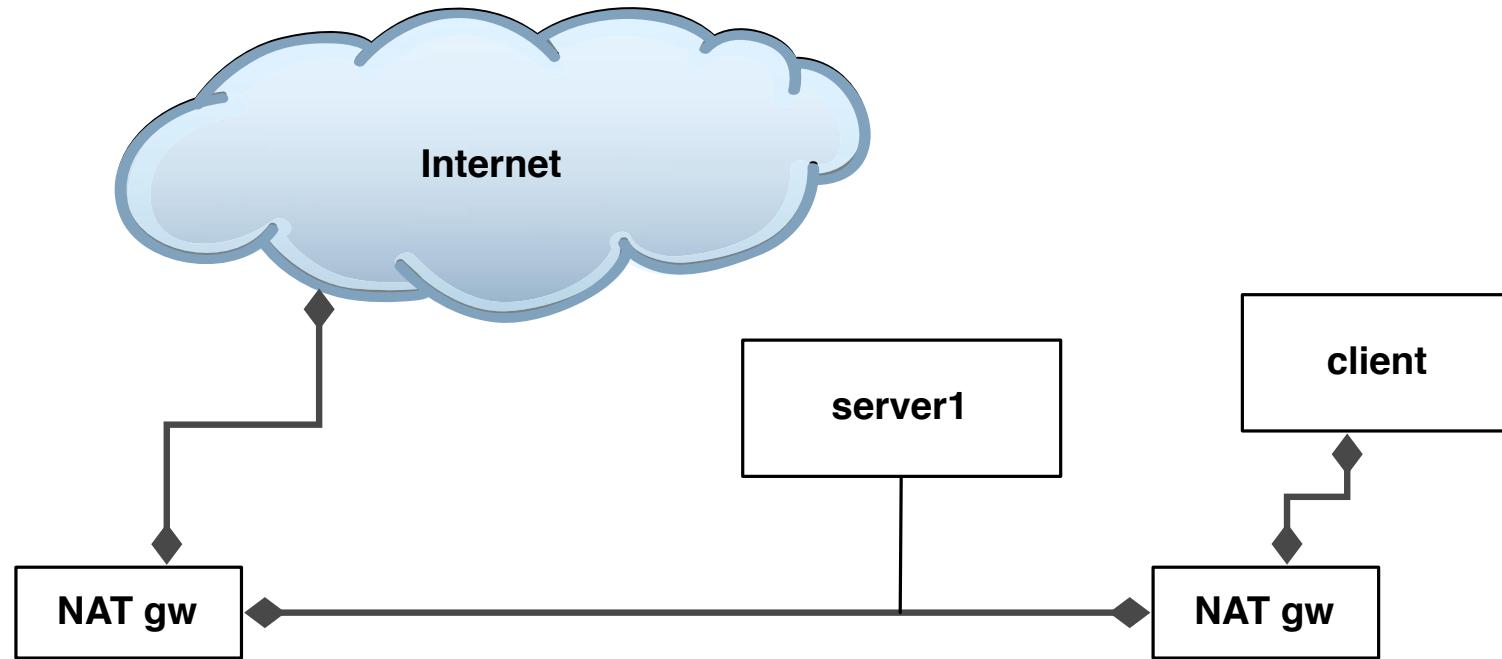
Hvis der skal overføres med password er SCP/SFTP fra Secure Shell at foretrække

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!

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

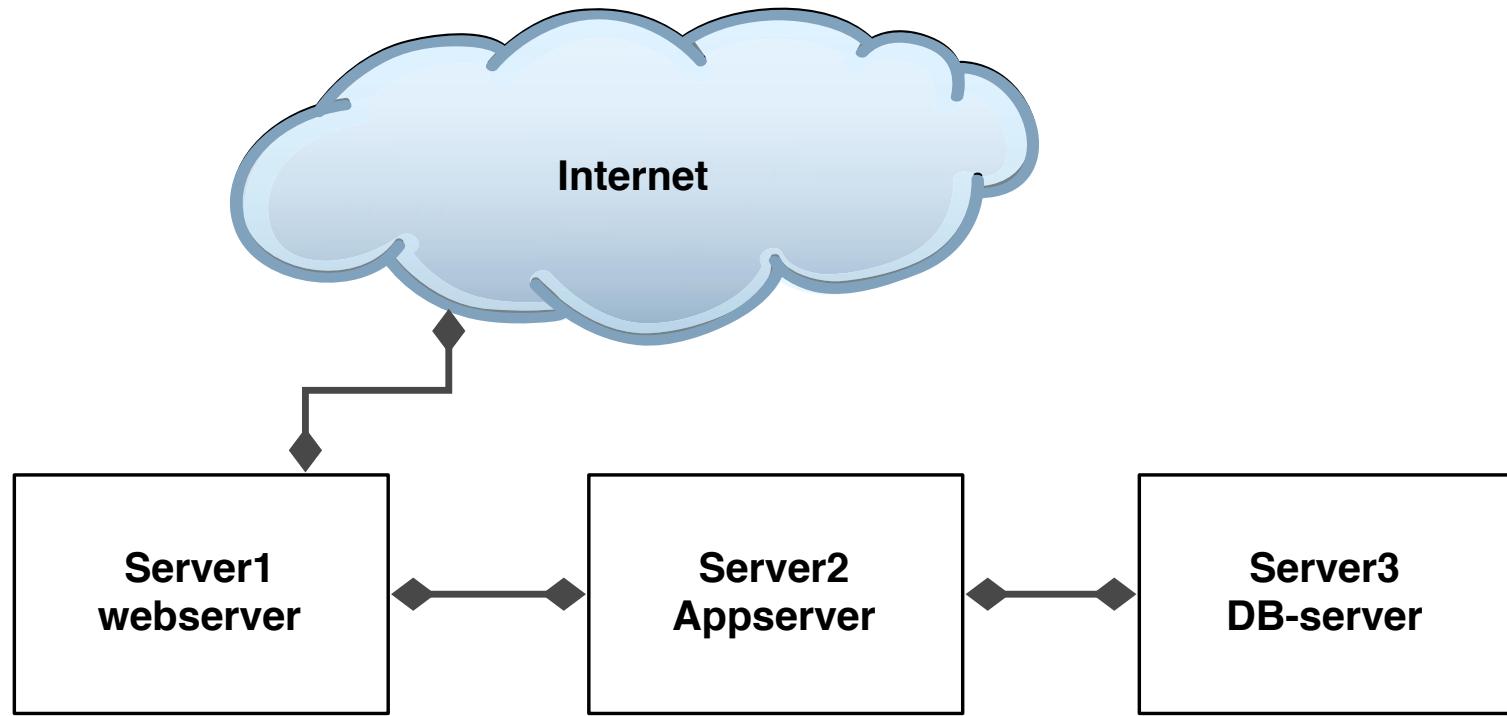
Lad være med at blokere for alt ICMP, så ødelægger du funktionaliteten i dit net

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

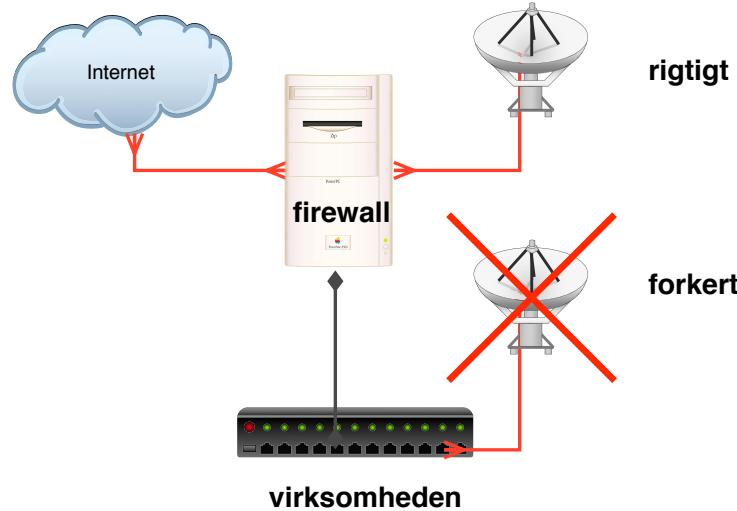
# Anti-pattern daisy-chain



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 større risiko for at sikkerheden brydes

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!

## 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 :-)

Henrik Lund Kramshøj, internet samurai  
[hlk@solido.net](mailto:hlk@solido.net)

<http://www.solidonetworks.com>

I er altid velkomne til at sende spørgsmål på e-mail

- O'Reilly Network Warrior - god allround bog, men også Cisco centrisk
- Stevens, Comer klassiske bøger om TCP/IP
- TCP/IP bogen på dansk måske
- O'Reilly IPv6 Network Administration
- KAME bøgerne om IPv6 protokollerne, meget detaljerede
- O'Reilly cookbooks: Cisco, BIND og Apache HTTPD m.fl.
- Cisco Press og website
- Juniper website
- Firewall bøger Cheswick
- Der findes mange gode bøger om netværk

*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