Welcome to

10. Network Attacks Introduction to Networking

KEA Kompetence OB2 Software Security

Henrik Kramselund he/him han/ham hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Github

10-network-attacks-intro.tex in the repo security-courses

inspirational slides, not curriculum or for presentation

Plan for today

Subjects

- Internet Protocol
- TCP and UDP
- Firewalls and related issues
- Intrusion Detection
- Host and Networks Based Intrusion Detection (HIDS/NIDS)
- Network Security Monitoring

Exercises

- Nmap and SYN flooding exercises

Reading Summary

Hacking, 2nd Edition: The Art of Exploitation, Jon Erickson chapter 4, browse

Browse: *TCP Synfloods - an old yet current problem, and improving pf's response to it* Henning Brauer, BSDCan 2017 http://quigon.bsws.de/papers/2017/bsdcan/

Goals today: Networking with some pentesting

What is pentest

> A penetration test, informally pen test, is an attack on a computer system that looks for security weaknesses, potentially gaining access to the computer's features and data.[1][2]

Source: quote from https://en.wikipedia.org/wiki/Penetration_test

Penetration testing is a simulation, with good intentions

People around the world constantly *test your defenses*

Often better to test at planned times

How to create DDoS simulations, tools and process

I use and recommend Kali Linux as the base for this

Agreements for testing networks

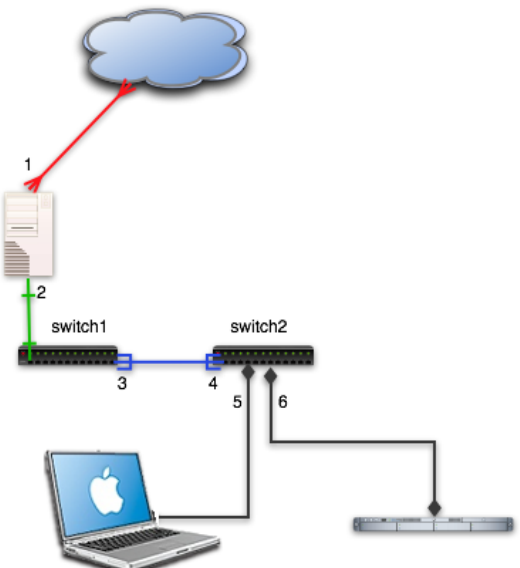
> Danish Criminal Code
> Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 1 år og 6 måneder straffes den, der uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et informationssystem.

Hacking can result in:

- Getting your devices confiscated by the police
- Paying damages to persons or businesses
- If older getting a fine and a record – even jail perhaps
- Getting a criminal record, making it hard to travel to some countries and working in security
- Fear of terror has increased the focus – so dont step over bounds!

Asking for permission and getting an OK before doing invasive tests, always!

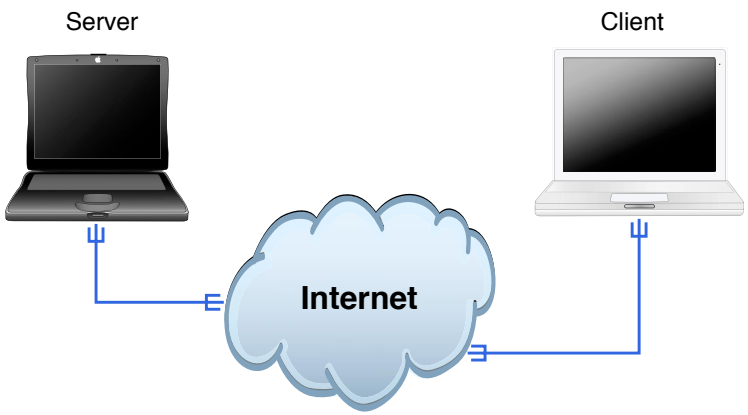
Course Network



Our network will be similar to regular networks, as found in enterprises

We have an isolated network, allowing us to sniff and mess with hacking tools.

Internet Protocol Suite



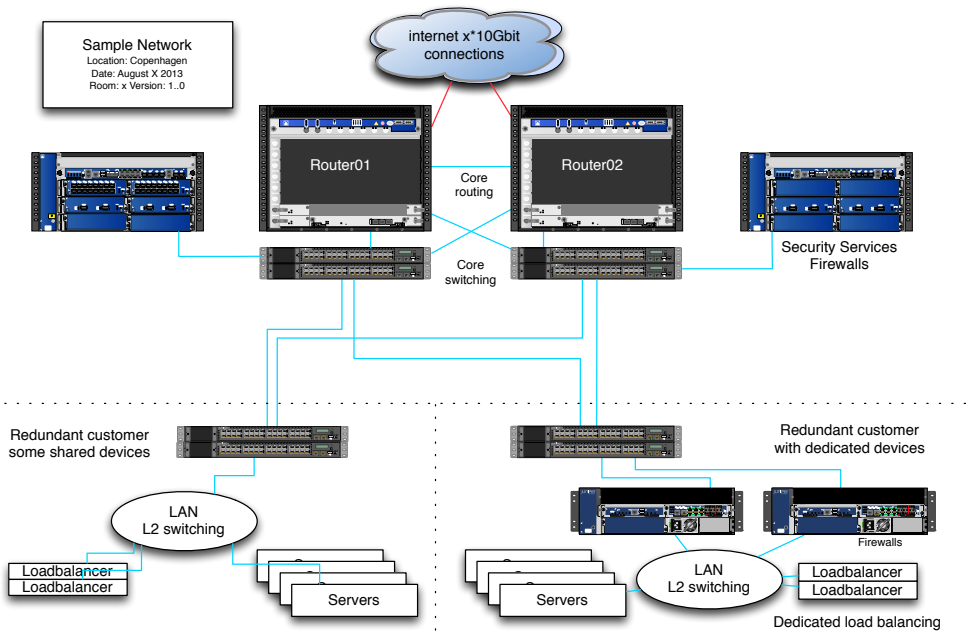
Client and servers

Rooted in academice circles

Protocols that are 20 years old, or more! TCP/IP version 4 from around 1983!

Very little encryption for many years. Today HTTPS, but almost nothing else encrypted.

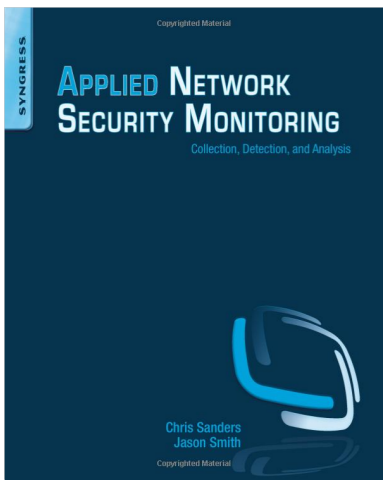
Networks today



Conclusion: Do as much as possible with your existing devices

Tuning and using features like stateless router filters works wonders

Book: Applied Network Security Monitoring (ANSM)

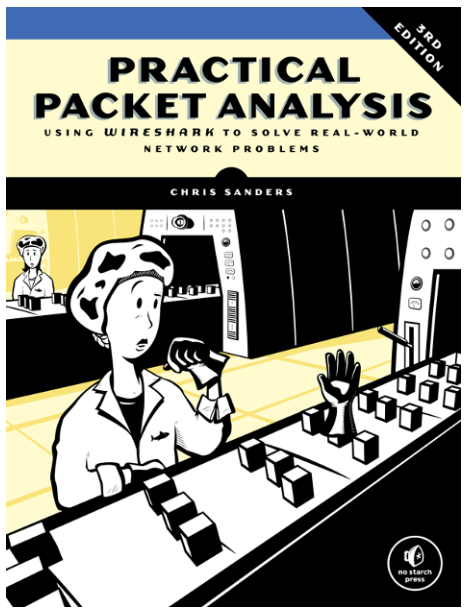


Applied Network Security Monitoring: Collection, Detection, and Analysis 1st Edition

Chris Sanders, Jason Smith eBook ISBN: 9780124172166 Paperback ISBN: 9780124172081 496 pp. Imprint: Syngress, December 2013

https://www.elsevier.com/books/applied-network-security-monitoring/unknown/978-0-12-417208-1

Book: Practical Packet Analysis (PPA)



Practical Packet Analysis, Using Wireshark to Solve Real-World Network Problems by Chris Sanders, 3rd Edition April 2017, 368 pp. ISBN-13: 978-1-59327-802-1

https://nostarch.com/packetanalysis3

Internet is Open Standards!

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

Request for comments - RFC - er en serie af dokumenter

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

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

Standards track:
Proposed Standard $\rightarrow$ Draft Standard $\rightarrow$ Standard

Åbne standarder = åbenhed, ikke garanti for sikkerhed

Hvad er Internet

Kommunikation mellem mennesker!

Baseret på TCP/IP

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

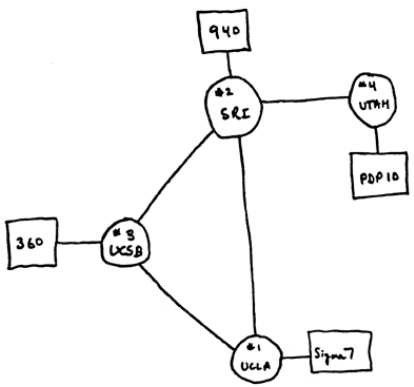
RFC-1958:

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

IP netværk: Internettet historisk set

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

Internet historisk set - anno 1969



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

De tidlige notater om Internet

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

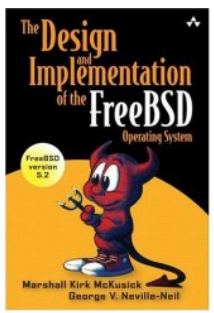
J.C.R. Licklider, MIT noter fra 1962 *On-Line Man Computer Communication*

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

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

De tidlige notater kan findes på nettet!

BSD UNIX



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

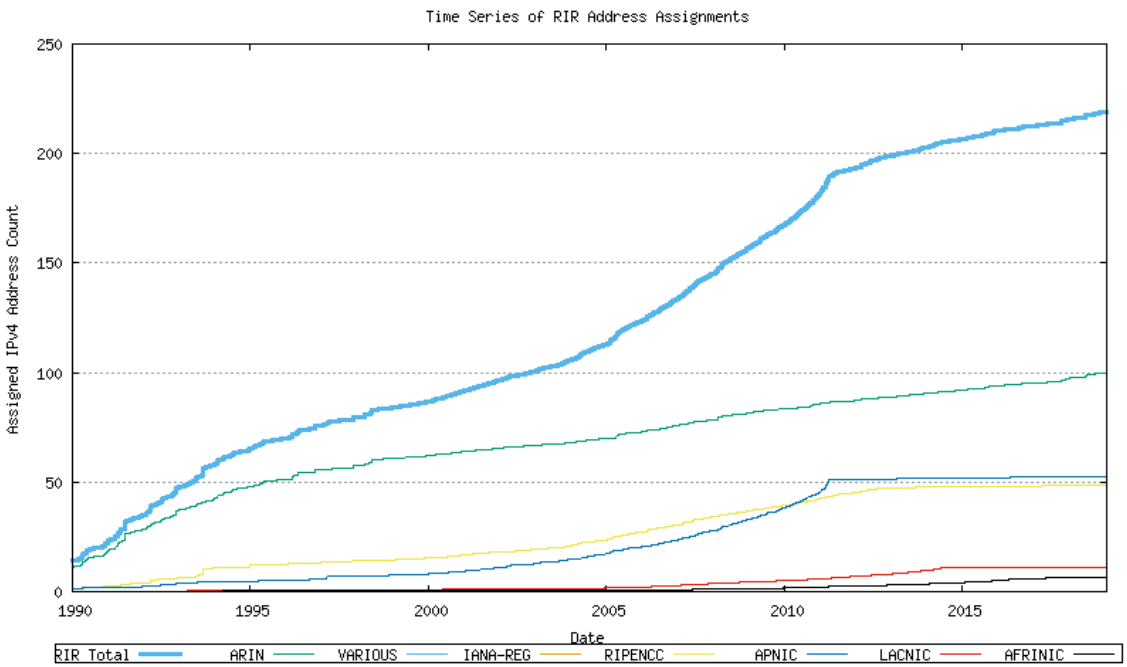
Bell Labs/AT&T var et telefonselskab - ikke et software hus

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

BSD står for Berkeley Software Distribution

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

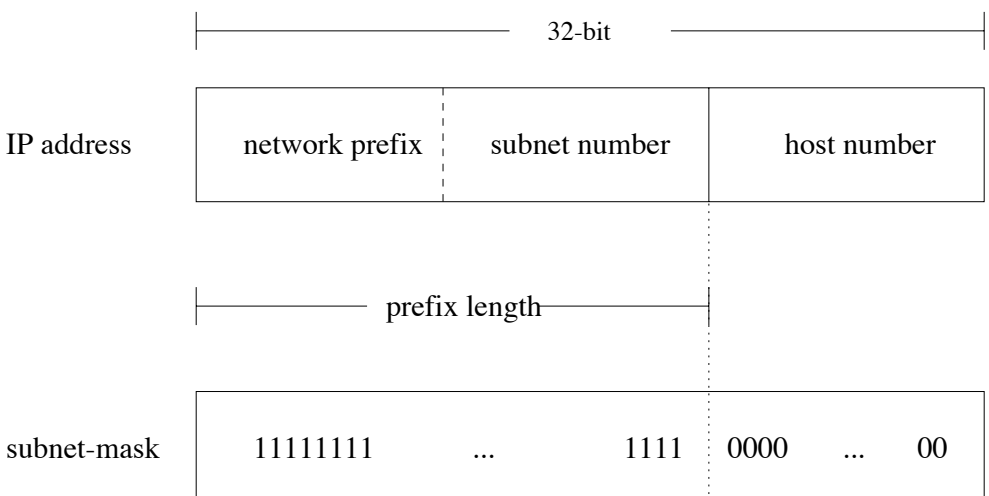
Hvad er Internet hosts



Cumulative RIR address assignments, per RIR

Source: IPv4 Address Report - 28-Jan-2019 http://www.potaroo.net/tools/ipv4/

Fælles adresserum



Hvad kendetegner internet idag

Der er et fælles adresserum baseret på 32-bit adresser, example 10.0.0.1

IPv4 addresser og skrivemåde

```
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-adresser som bits

```
IP-adresse: 127.0.0.1
Heltal: 2130706433
Binary: 11111110000000000000000000000001
```

IP-adresser kan også konverteres til bits

Computeren regner binært, vi bruger dot-notationen

Internet ABC

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

Stop saying C, say /24

CIDR Classless Inter-Domain Routing

Classful routing		Classless routing CIDR	
4 class C networks	Inherent subnet mask	Supernet	Subnet mask
192.0.8.0	255.255.255.0	192.0.8.0	255.255.252.0
192.0.9.0	255.255.255.0		252d=11111100b
192.0.10.0	255.255.255.0		
192.0.11.0	255.255.255.0	Base network/prefix	
		192.10.8.0/22	

Subnetmasker var oprindeligt indforstået

Man tildelte flere C-klasser - 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

IPv4 addresser opsummering

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

RFC-1918 Private Networks

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)

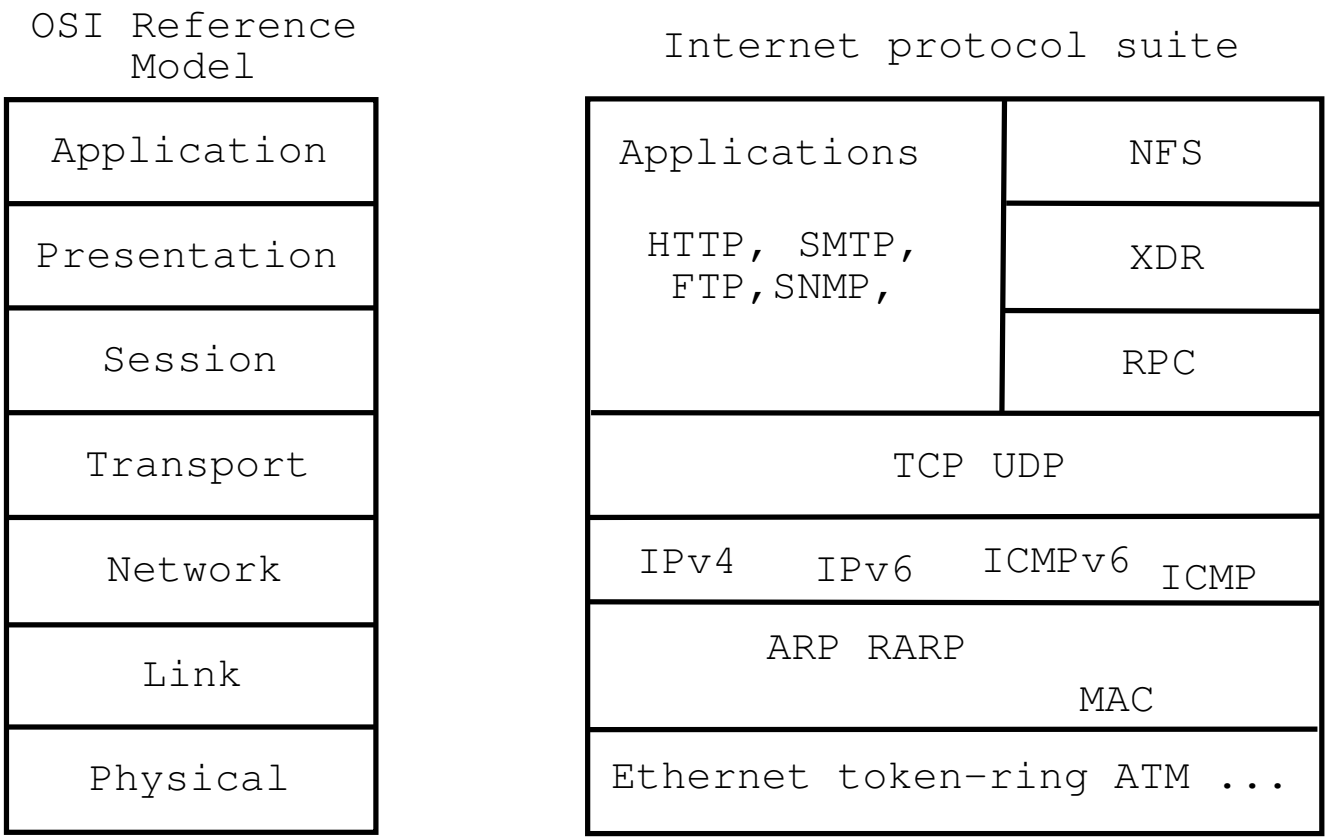
Address Allocation for Private Internets RFC-1918 adresserne!

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

```
The blocks 192.0.2.0/24 (TEST-NET-1), 198.51.100.0/24 (TEST-NET-2),
and 203.0.113.0/24 (TEST-NET-3) are provided for use in
documentation.

169.254.0.0/16 has been ear-marked as the IP range to use for end node
auto-configuration when a DHCP server may not be found
```

OSI og Internet modellerne



Netværkshardware

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

Ofte benyttede idag er:

- Ethernet - varianter 10mbit, 100mbit, gigabit, 10G, 100G, 200G, 400G, ...
- Wireless 802.11 teknologier
- ADSL/ATM teknologier til WAN forbindelser
- MPLS ligeledes til WAN forbindelser

Ethernet kan bruge kobberledninger eller fiber

WAN forbindelser er typisk fiber på grund af afstanden mellem routere

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

Ethernet stik, kabler og dioder

Dioder viser typisk om der er link, hastighed samt aktivitet

Trådløse teknologier

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

MAC adresser

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

Netværksteknologierne benytter adresser på lag 2

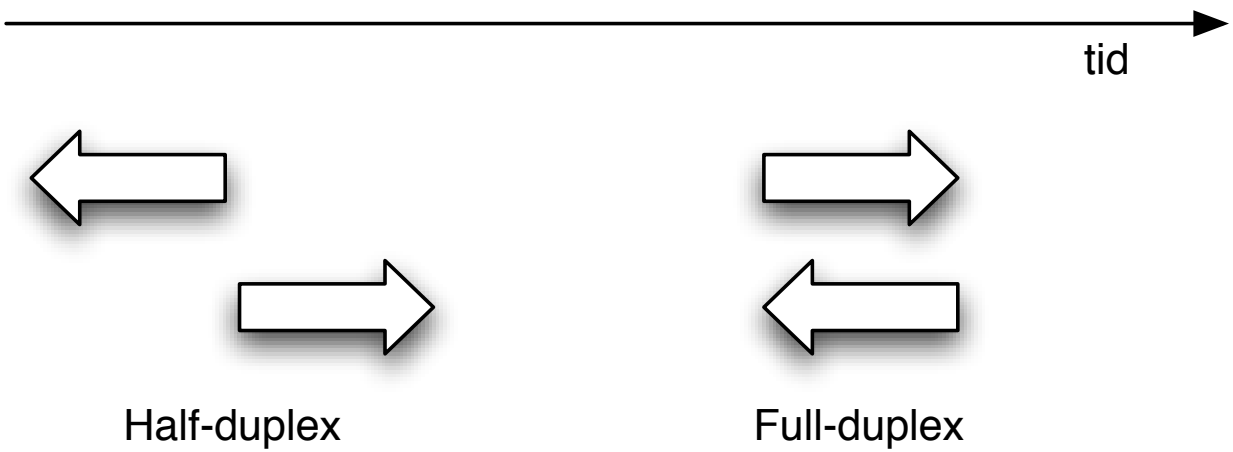
Typisk svarende til 48-bit MAC adresser som kendes fra Ethernet MAC-48/EUI-48

Første halvdel af adresserne er Organizationally Unique Identifier (OUI)

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

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

Half/full-duplex og speed



Hvad hastighed overføres data med?

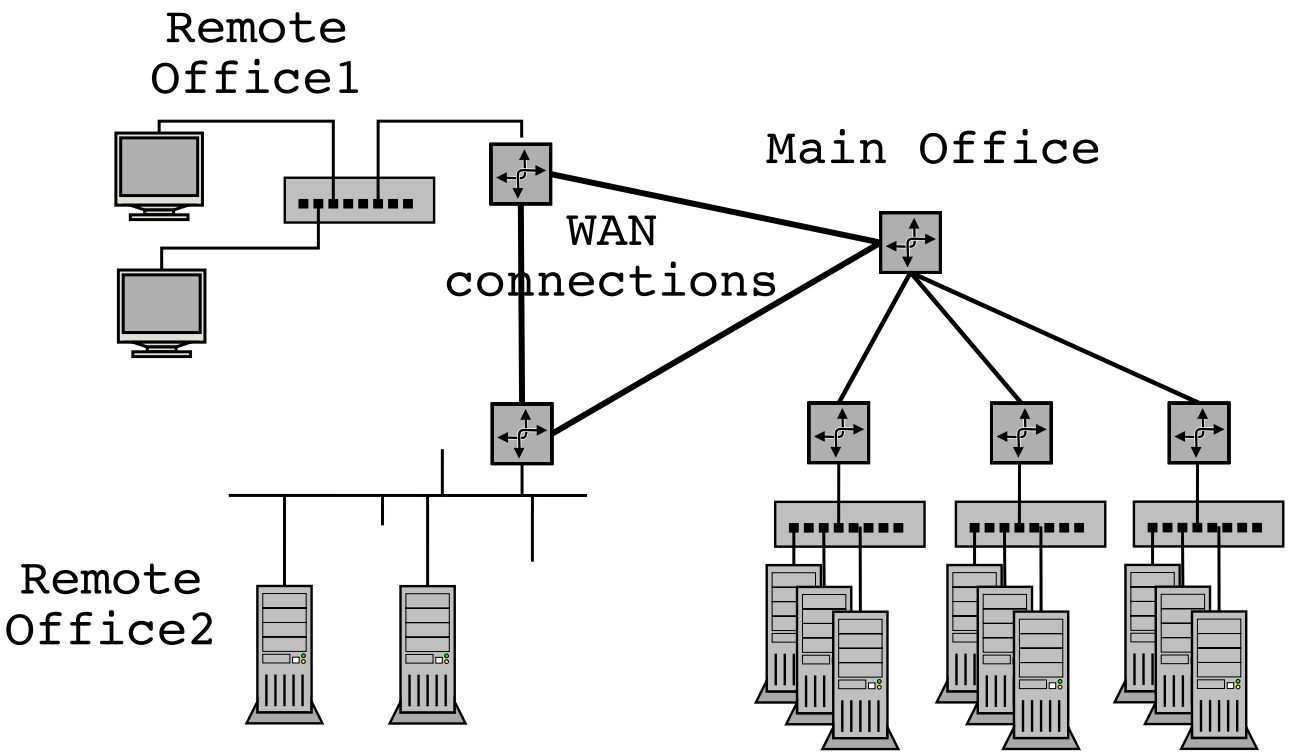
De fleste nyere Ethernet netkort kan køre i fuld-duplex

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

Ethernet kan benytte auto-negotiation - der ofte virker

Klart bedre i gigabitnetkort men pas på

Broer og routere



Fysisk er der en begrænsing for hvor lange ledningerne må være

Bridges

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

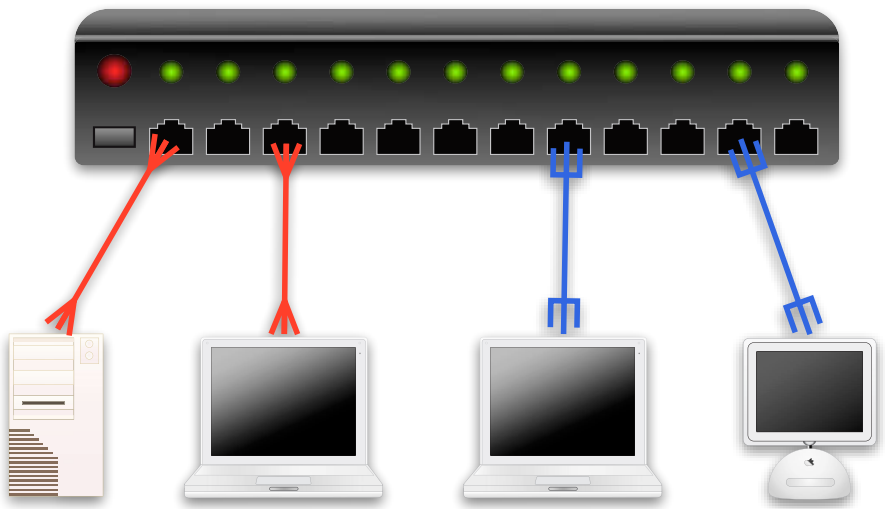
Virker som en forstærker på signalet, men mere intelligent

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 søg efter Aloha-net
http://en.wikipedia.org/wiki/ALOHAnet

En switch

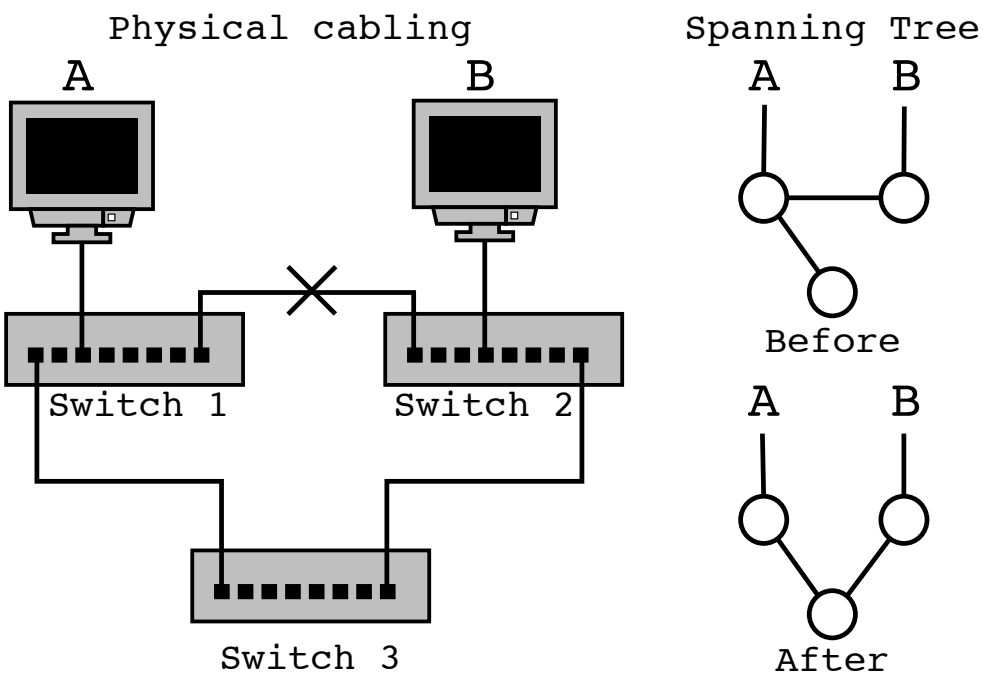


Ved at fortsætte udviklingen kunne man samle broer til en switch

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

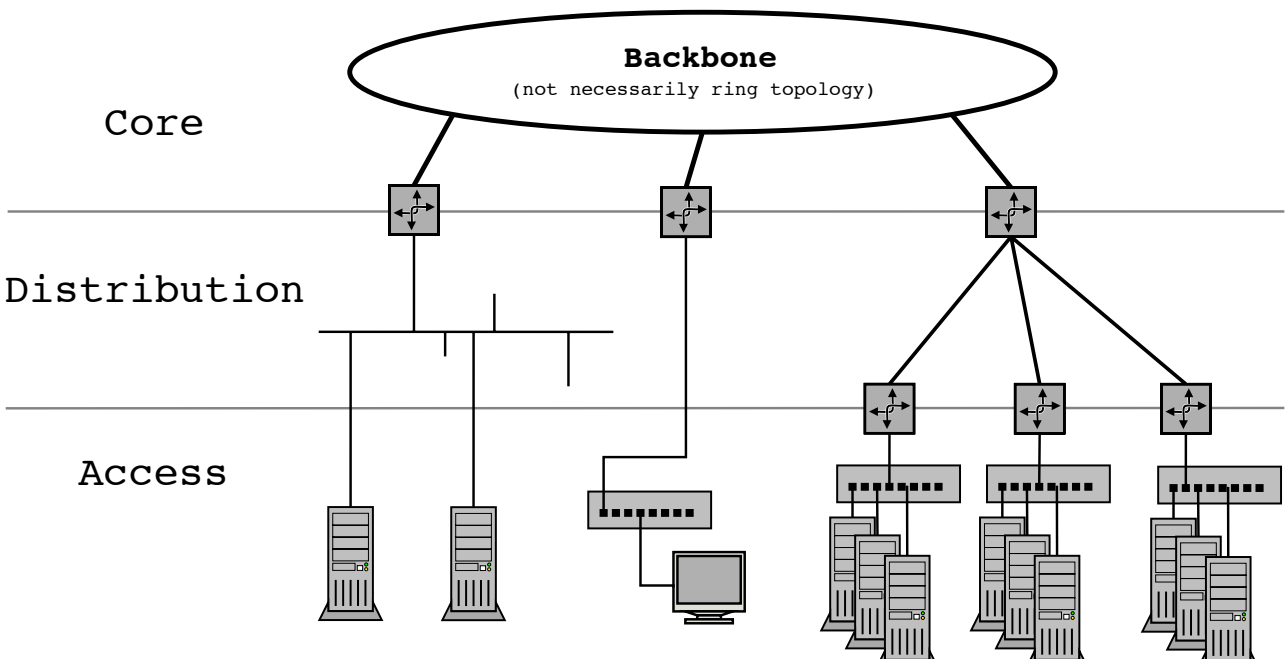
Bemærk performance begrænses af backplane i switchen

Topologier og Spanning Tree Protocol



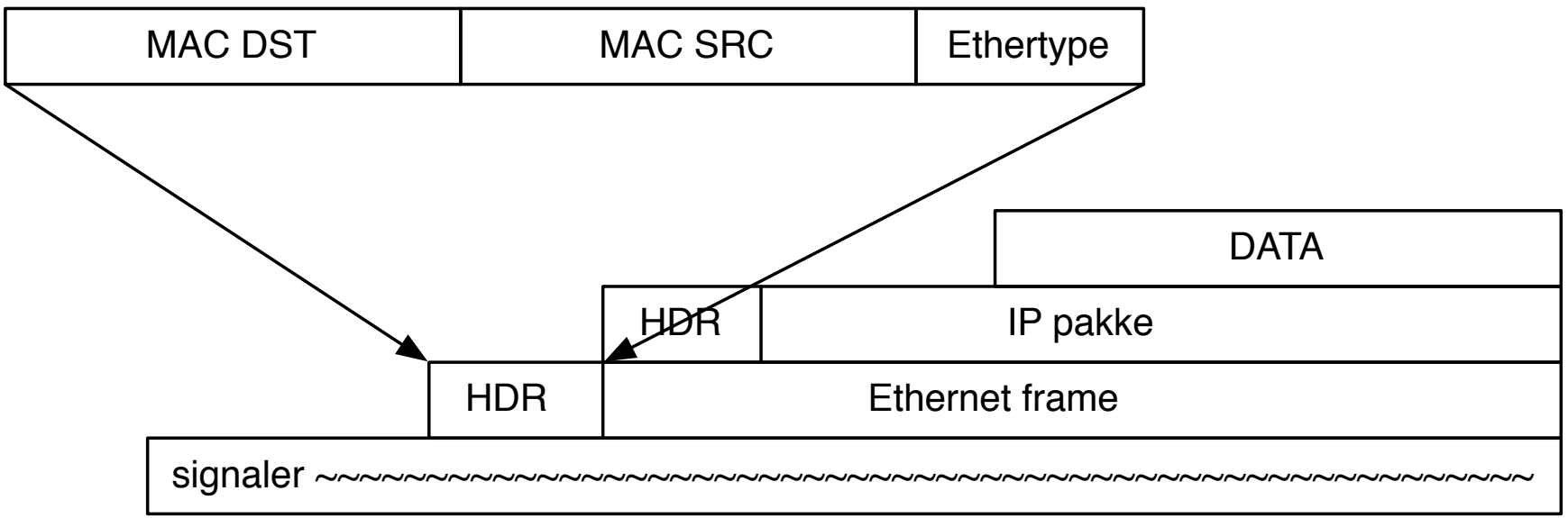
Se mere i bogen af Radia Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*

Core, Distribution og Access net



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

Pakker i en datastrøm

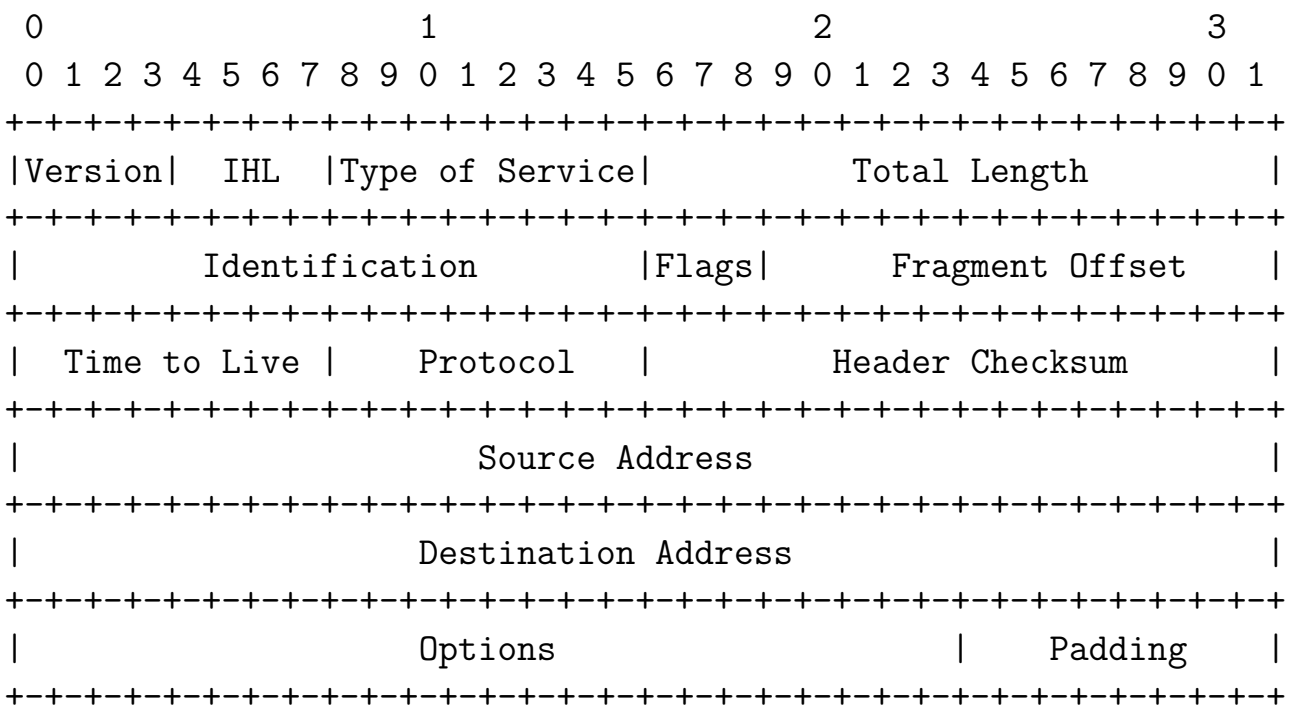


Ser vi data som en datastrøm er pakkerne blot et mønster lagt henover data

Netværksteknologien definerer start og slut på en frame

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

IPv4 pakken - header - RFC-791



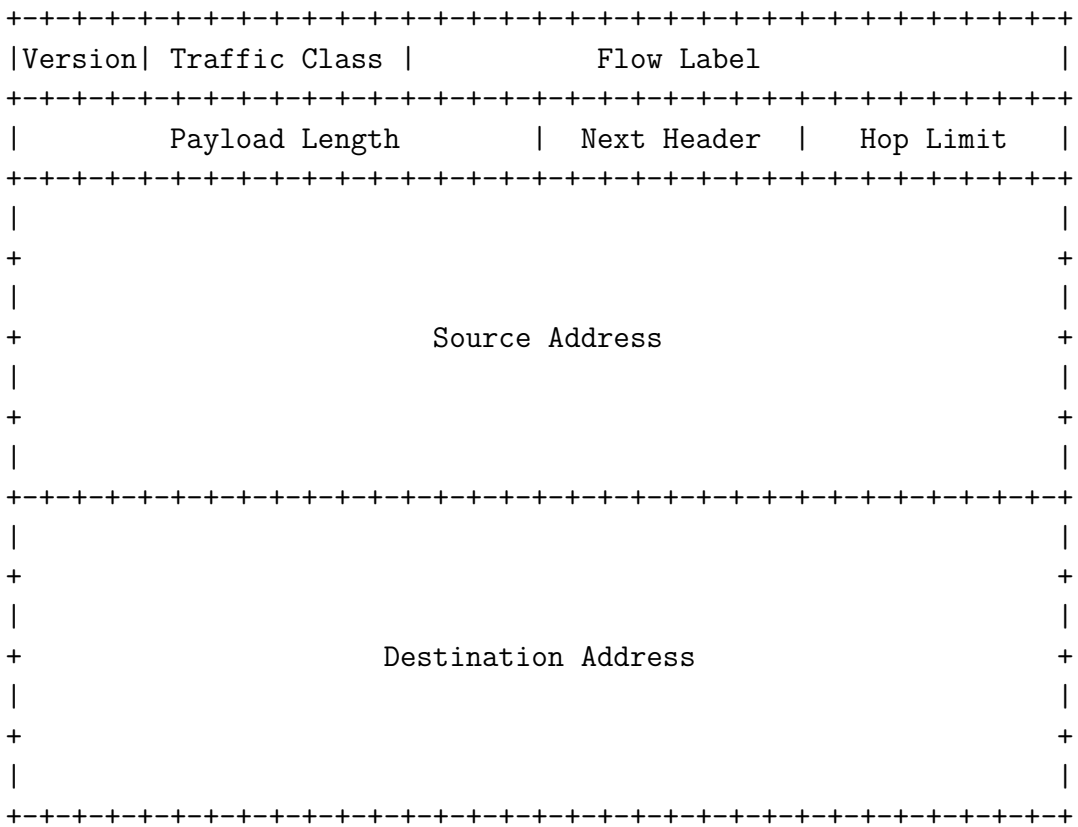
Example Internet Datagram Header

IPv6 pakken - header - RFC-2460

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

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

IPv6 pakken - header - RFC-2460



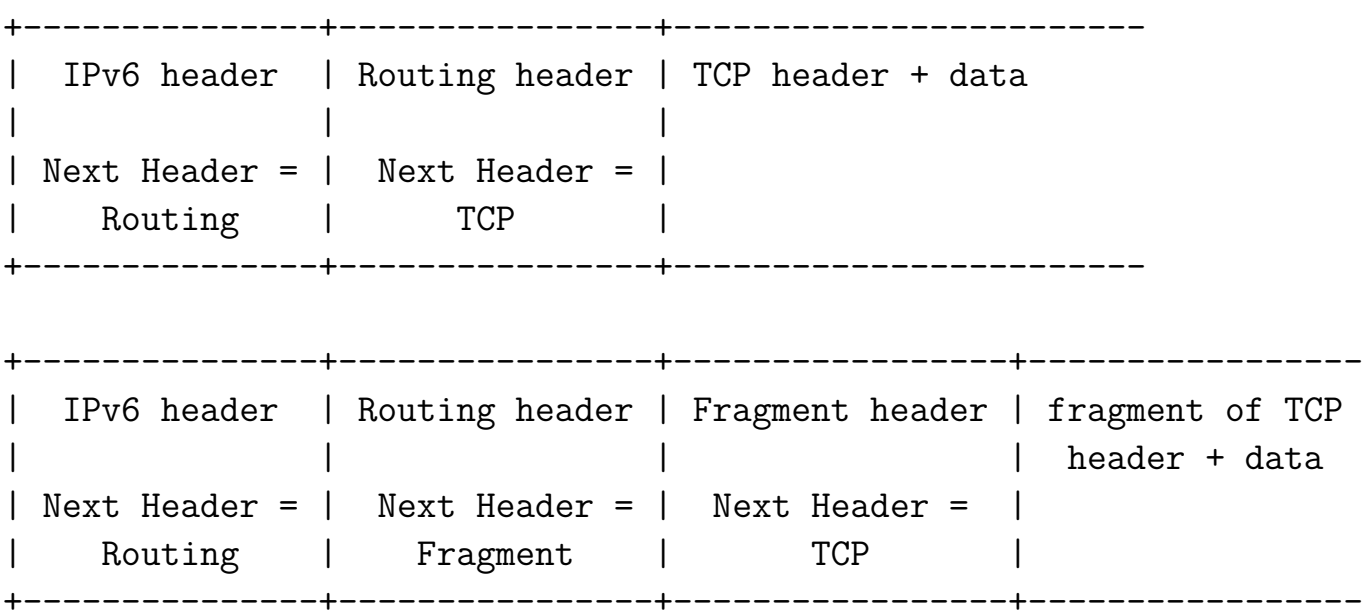
IPv6 pakken - extension headers RFC-2460

Fuld IPv6 implementation indeholder:

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

Ja, IPsec er en del af IPv6!

Placering af extension headers



Hvordan bruger man IPv6

www.zencurity.com

hlk@zencurity.com

DNS AAAA record tilføjes

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

IPv6 addresser og skrivemåde

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 0:0:0:0:0:0:0:0 er det samme som 0000:0000:0000:0000:0000:0000:0000:0000
- Dvs min webservers IPv6 adresse kan skrives som: 2001:16d8:ff00:12f::2
- Specielle adresser: ::1 localhost/loopback og :: default route
- Læs mere i RFC-3513

IPv6 addresser - prefix notation

CIDR Classless Inter-Domain Routing RFC-1519

Aggregatable Global Unicast

2001::/16 RIR subTLA space

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

2002::/16 6to4 prefix

3ffe::/16 6bone allocation

link-local unicast addresses
fe80::/10 genereres udfra MAC addresserne EUI-64

IP karakteristik

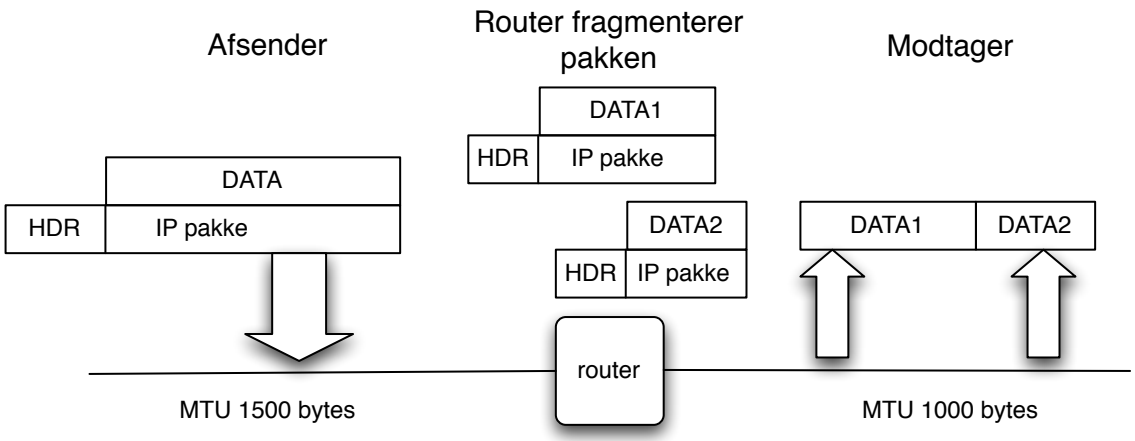
Fælles adresserum, dog version 4 for sig, og IPv6 for sig.

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

Kræver ikke mange services fra underliggende teknologi *dumt netværk*

Defineret gennem åben standardiseringsprocess og RFC-dokumenter

Fragmentering og PMTU



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

Pakkestørrelsen kaldes MTU Maximum Transmission Unit

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

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

ICMP Internet Control Message Protocol

Kontrolprotokol og fejlmeldinger

Nogle af de mest almindelige beskedtyper

- echo
- netmask
- info

Bruges generelt til *signalering*

Defineret i RFC-792

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

ICMP beskedtyper

Type

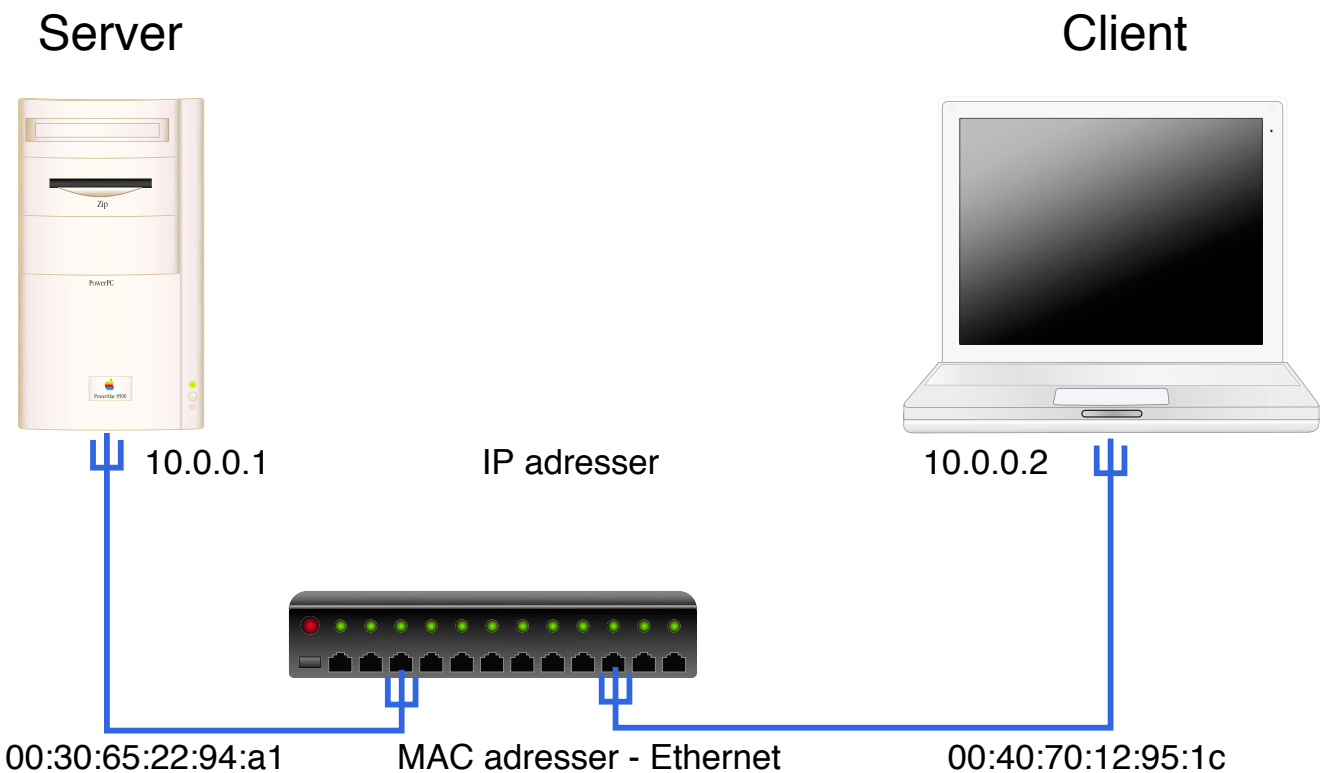
- 0 = net unreachable;
- 1 = host unreachable;
- 2 = protocol unreachable;
- 3 = port unreachable;
- 4 = fragmentation needed and DF set;
- 5 = source route failed.

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

Tillad ICMP types:

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

Hvordan virker ARP?



Hvordan virker ARP? - 2

ping 10.0.0.2 udført på server medfører

ARP Address Resolution Protocol request/reply:

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

IP ICMP request/reply:

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

ARP udføres altid på Ethernet før der kan sendes IP trafik

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

ARP cache

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

ARP cache kan vises med kommandoen `arp -an`

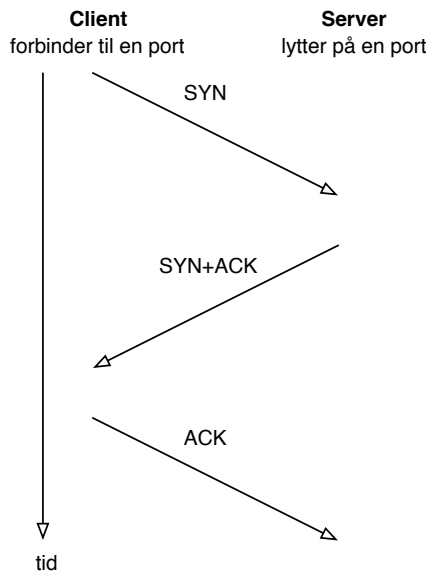
-a viser alle

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

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

Læs mere med `man 4 arp`

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

Well-known port numbers

IANA vedligeholder en liste over magiske konstanter i IP

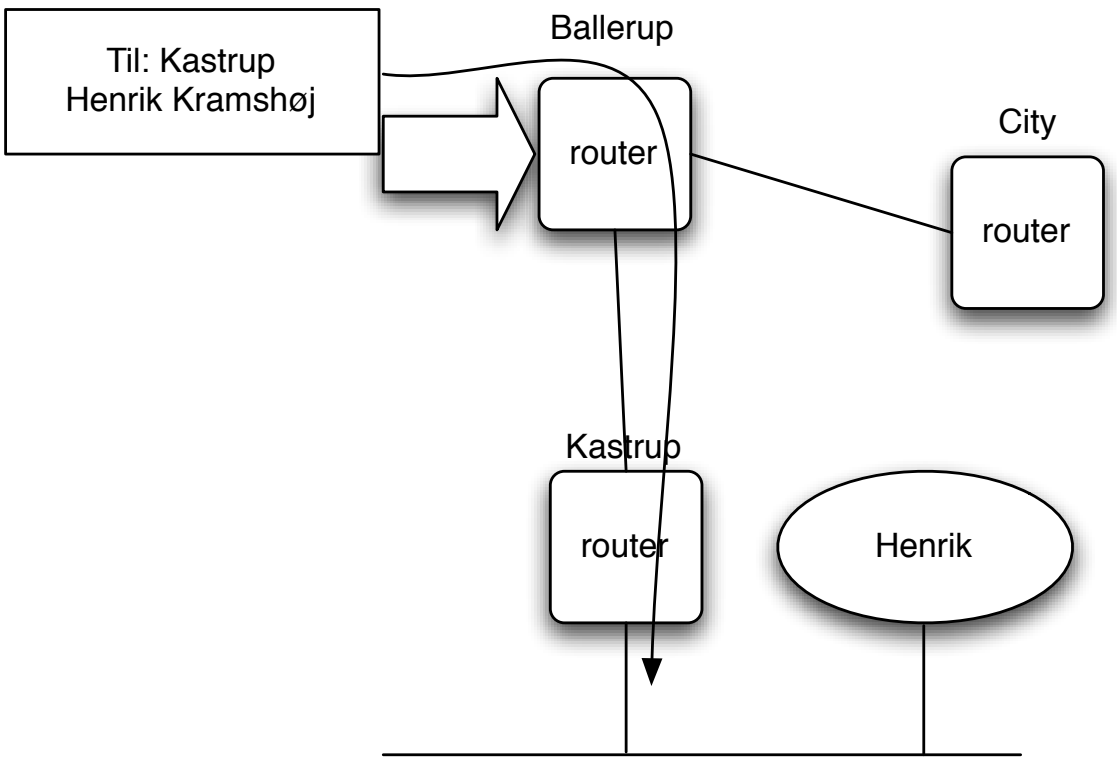
De har lister med hvilke protokoller har hvilke protokol ID m.v.

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

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

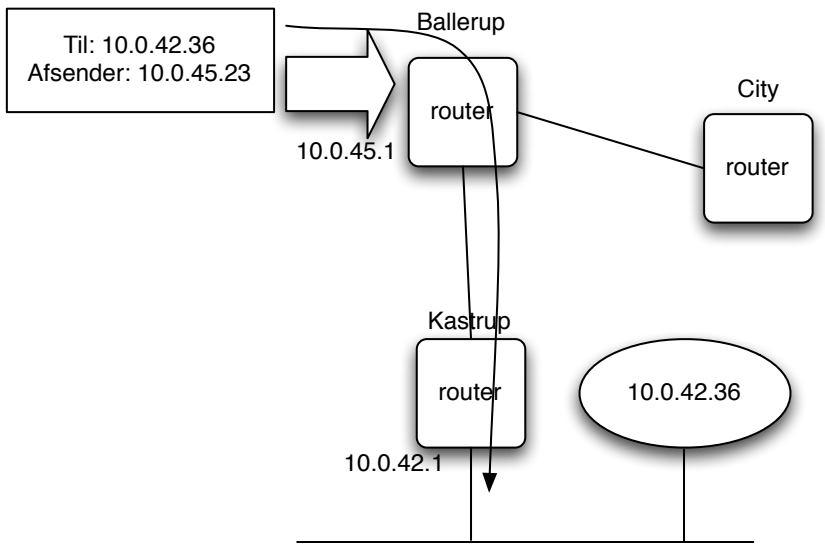
Se flere på http://www.iana.org

Hierarkisk routing



Hvordan kommer pakkerne frem til modtageren

IP default gateway



IP routing er nemt, longest match

En host kender typisk en default gateway i nærheden

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

Routing

routing table - tabel over netværkskort og tilhørende adresser

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

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

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

de lidt ældre routing protokoller har ingen sikkerhedsmekanismer

IP benytter longest match i routing tabeller!

Den mest specifikke route gælder for forward af en pakke!

Routing forståelse

```
$ netstat -rn
Routing tables

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

Start med kun at se på Destination, Gateway og Netinterface

Ping

ICMP - Internet Control Message Protocol

Benyttes til fejlbeskeder og til diagnosticering af forbindelser

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

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

traceroute

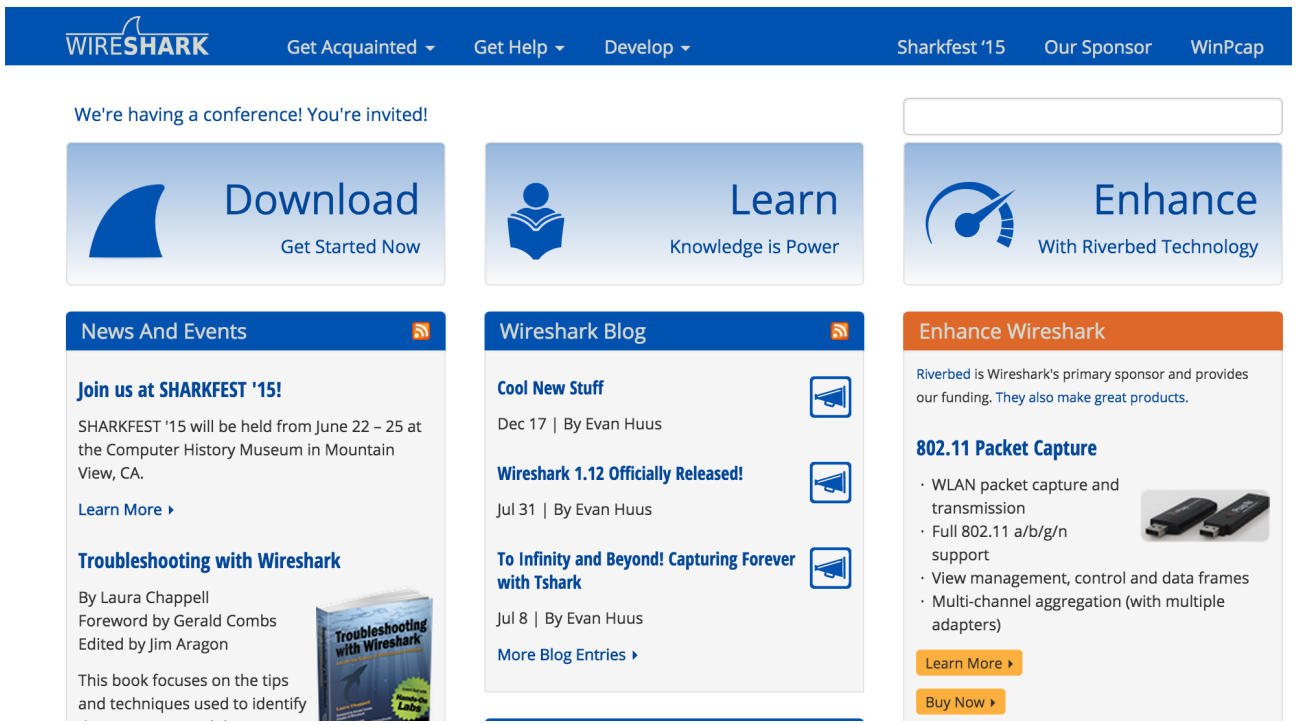
traceroute programmet virker ved hjælp af TTL

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

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

```
$ traceroute 185.129.60.129
traceroute to 185.129.60.129 (185.129.60.129),
30 hops max, 40 byte packets
 1  safri (10.0.0.11)  3.577 ms  0.565 ms  0.323 ms
 2  router (185.129.60.129)  1.481 ms  1.374 ms  1.261 ms
```

Wireshark - grafisk pakkesniffer



http://www.wireshark.org
både til Windows og UNIX

Firewalls and related issues

> In computing, a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.[1] A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.[2]

Source: Wikipedia

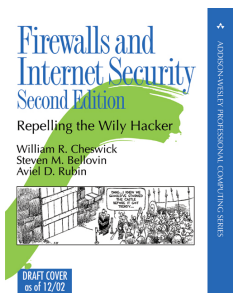
https://en.wikipedia.org/wiki/Firewall_(computing)

http://www.wilyhacker.com/ Cheswick chapter 2 PDF *A Security Review of Protocols: Lower Layers*

- Network layer, packet filters, application level, stateless, stateful

Firewalls are by design a choke point, natural place to do network security monitoring!

Firewall historik



Firewalls har været kendt siden starten af 90'erne

Første bog *Firewalls and Internet Security* udkom i 1994 men kan stadig anbefales, læs den på http://www.wilyhacker.com/

2003 kom den i anden udgave *Firewalls and Internet Security* William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, Addison-Wesley, 2nd edition

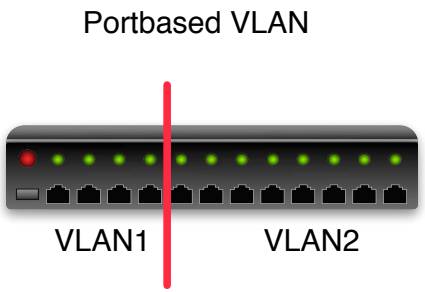
Reading about firewalls

http://www.wilyhacker.com/ Cheswick chapter 3 PDF *Security Review: The Upper Layers*

- How to configure firewalls often boil down to, should we allow protocol X
- If we allow SMB through an internet firewall, we are asking for trouble

Skim chapters from 1st edition:
http://www.wilyhacker.com/1e/chap03.pdf
http://www.wilyhacker.com/1e/chap04.pdf

Together with Firewalls - VLAN Virtual LAN



Nogle switche tillader at man opdeler portene

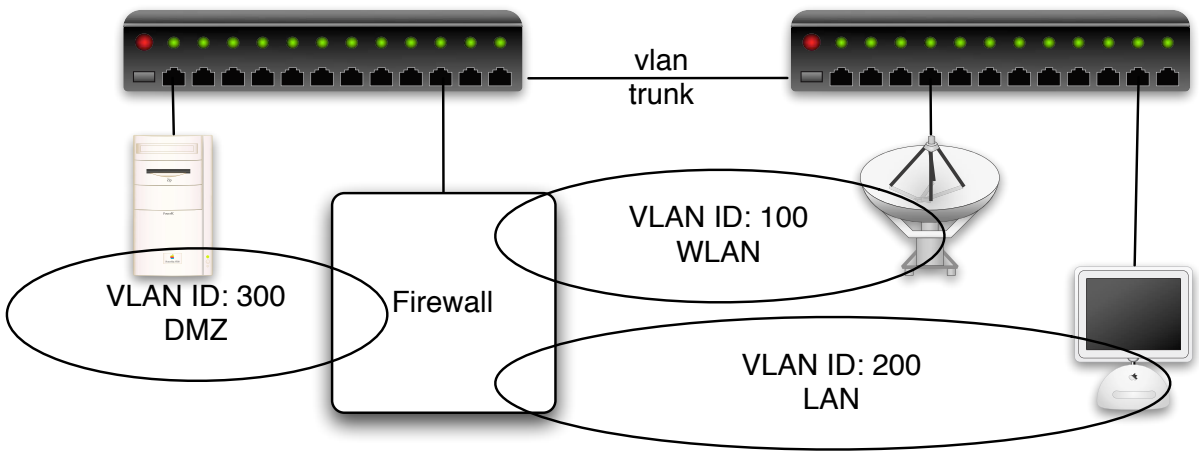
Denne opdeling kaldes VLAN og portbaseret er det mest simple

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

IEEE 802.1q



Med 802.1q tillades VLAN tagging på Ethernet niveau

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

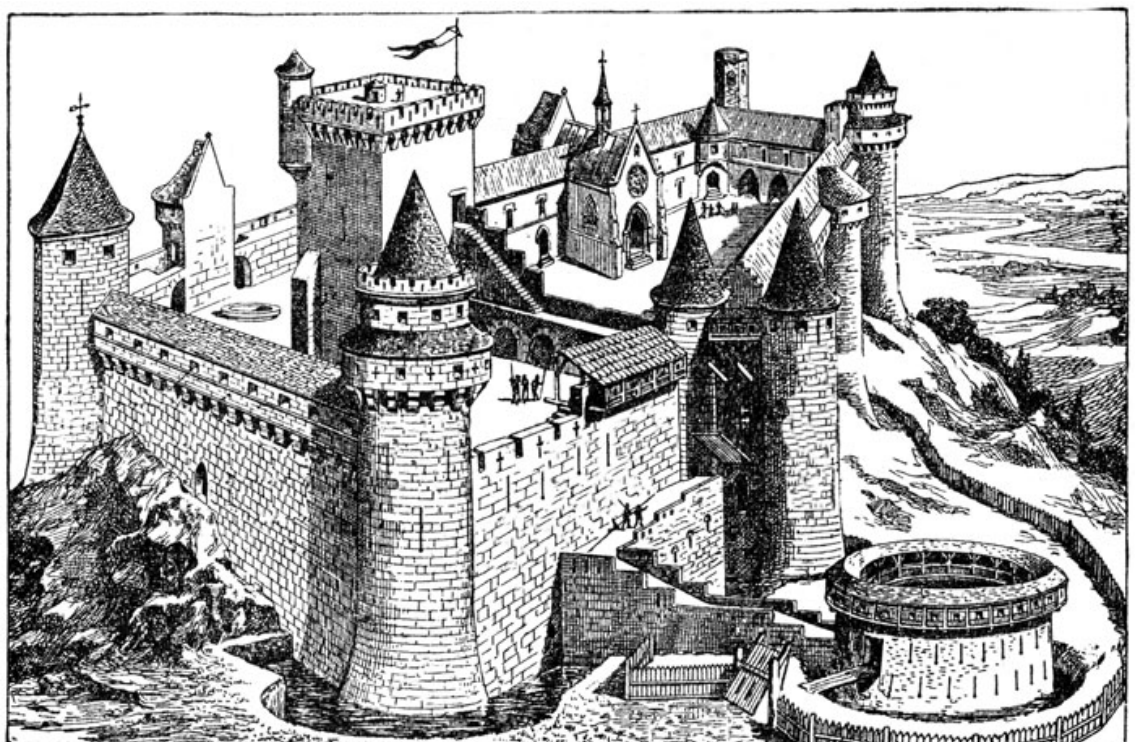
VLAN trunking giver mulighed for at dele VLANs ud på flere switches

Generic IP Firewalls

En firewall er noget som blokerer traffik på Internet

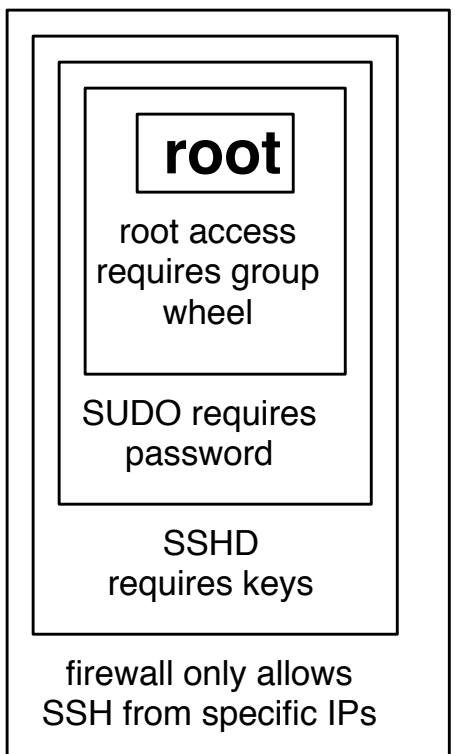
En firewall er noget som tillader traffik på Internet

Defense in depth



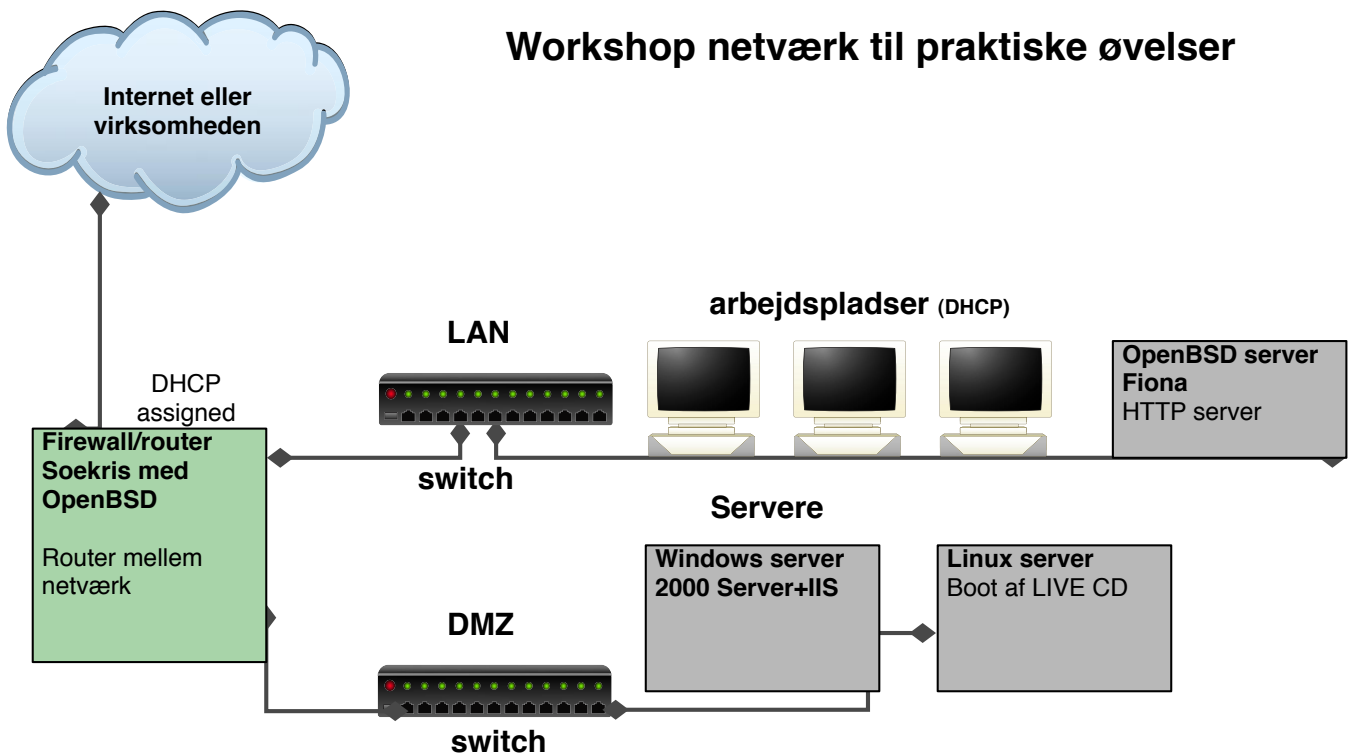
Picture originally from: http://karenswhimsy.com/public-domain-images

Defense in depth - layered security

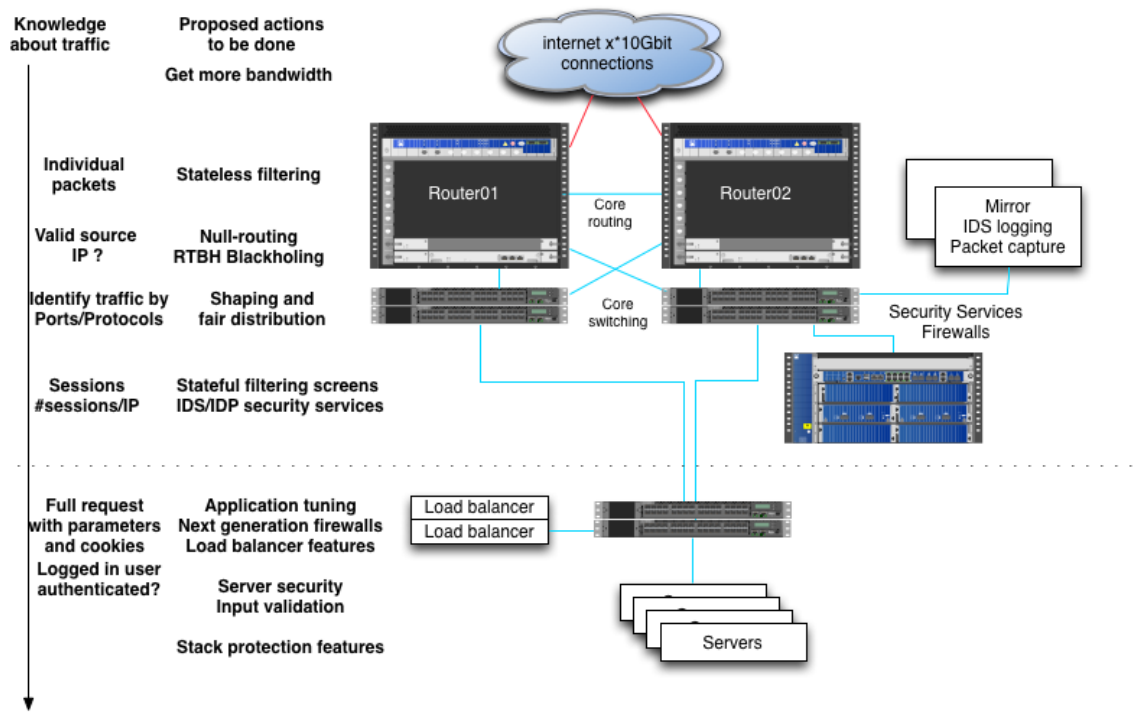


Multiple layers of security! Isolation!

Small networks

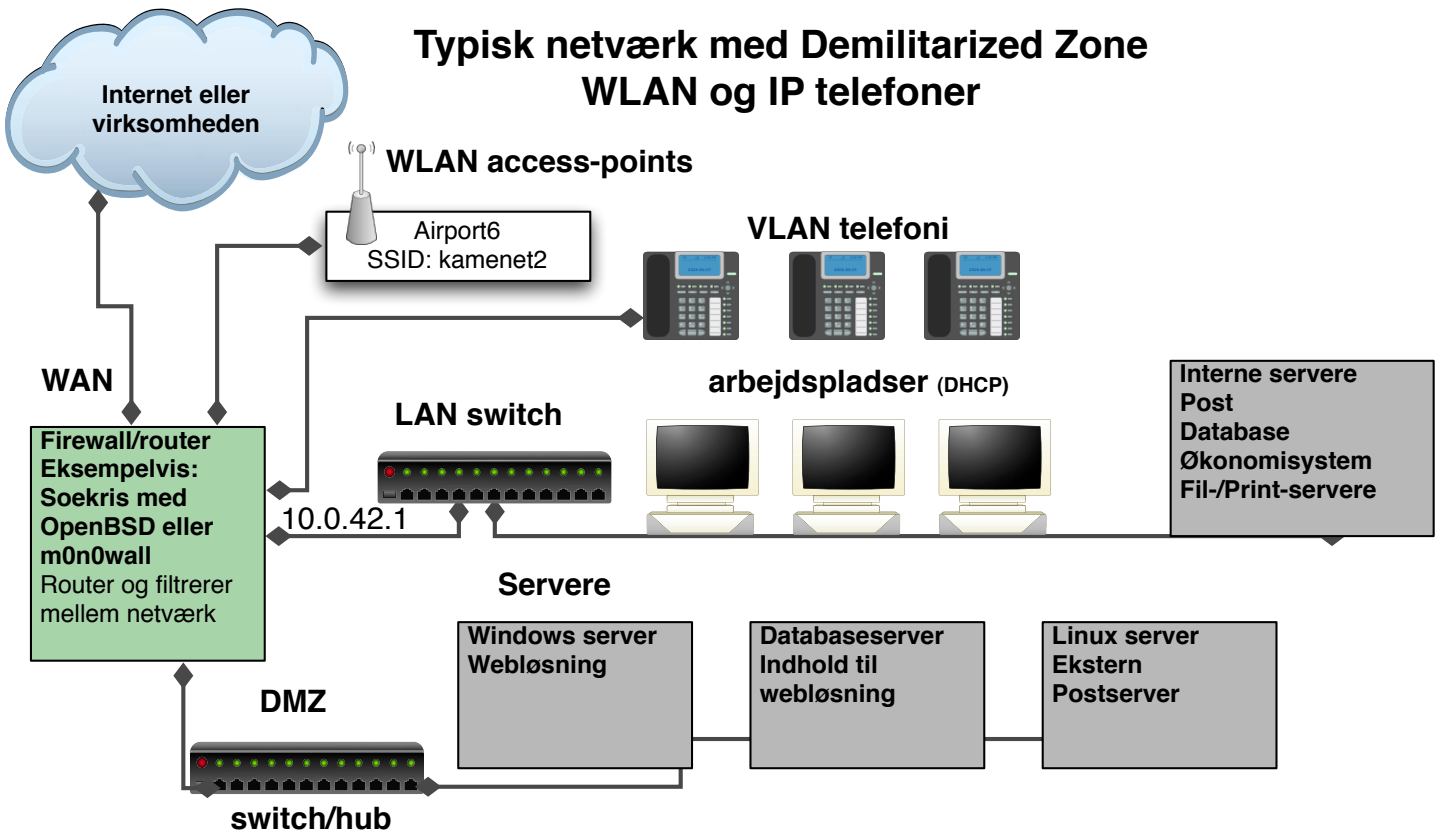


Firewall er ikke alene



Forsvaret er som altid - flere lag af sikkerhed!

Unified communications



Firewallrollen idag

Idag skal en firewall være med til at:

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

Det kræver overblik!

Modern Firewalls

Basalt set et netværksfilter - det yderste fæstningsværk

Indeholder typisk:

- Grafisk brugergrænseflade til konfiguration - er det en fordel?
- TCP/IP filtermuligheder - pakkernes afsender, modtager, retning ind/ud, porte, protokol, ...
- både IPv4 og IPv6
- foruddefinerede regler/eksempler - er det godt hvis det er nemt at tilføje/åbne en usikker protokol?
- typisk NAT funktionalitet indbygget
- typisk mulighed for nogle serverfunktioner: kan agere DHCP-server, DNS caching server og lignende

En router med Access Control Lists - kaldes ofte netværksfilter, mens en dedikeret maskine kaldes firewall

Sample rules from OpenBSD PF

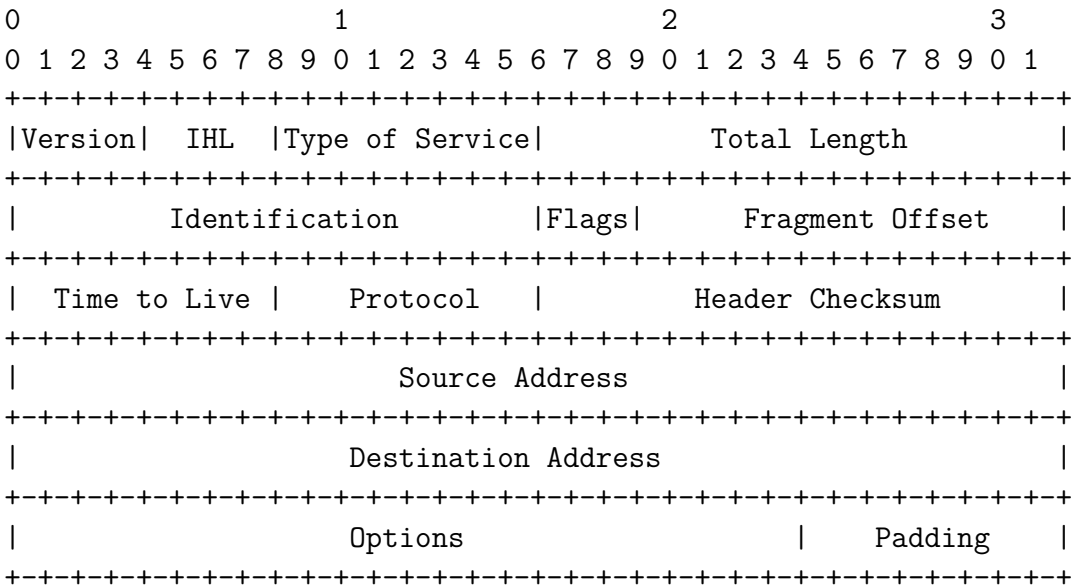
```
# hosts and networks
router="217.157.20.129"
webserver="217.157.20.131"
homenet=" 192.168.1.0/24, 1.2.3.4/24 "
wlan="10.0.42.0/24"
wireless=wi0
set skip lo0
# things not used
spoofed=" 127.0.0.0/8, 172.16.0.0/12, 10.0.0.0/16, 255.255.255.255/32 "

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

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

pass out
```

Packet filtering



Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste stateful inspection

Kommercielle firewalls

- Checkpoint Firewall-1 http://www.checkpoint.com
- Cisco ASA http://www.cisco.com
- Clavister firewalls http://www.clavister.com
- Juniper SRX http://www.juniper.net
- Palo Alto https://www.paloaltonetworks.com/
- Fortinet https://www.fortinet.com/

Ovenstående er dem som jeg oftest ser ude hos mine kunder i Danmark

Open source baserede firewalls

- Linux firewalls IP tables, use command line tool ufw Uncomplicated Firewall!
- Firewall GUIs ovenpå Linux - mange! nogle er kommercielle produkter
- OpenBSD PF http://www.openbsd.org
- FreeBSD IPFW og IPFW2 http://www.freebsd.org
- Mac OS X benytter OpenBSD PF
- FreeBSD inkluderer også OpenBSD PF

NB: kun eksempler og dem jeg selv har brugt

Hardware eller software

Man hører indimellem begrebet *hardware firewall*

Det er dog et faktum at en firewall består af:

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

Det giver ikke mening at kalde en ASA 5501 en hardware firewall
og en APU2C4 med OpenBSD for en software firewall!

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

Det er også fint at tale om host-firewalls, altså at servere og laptops har firewall slået til

firewall koncepter

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

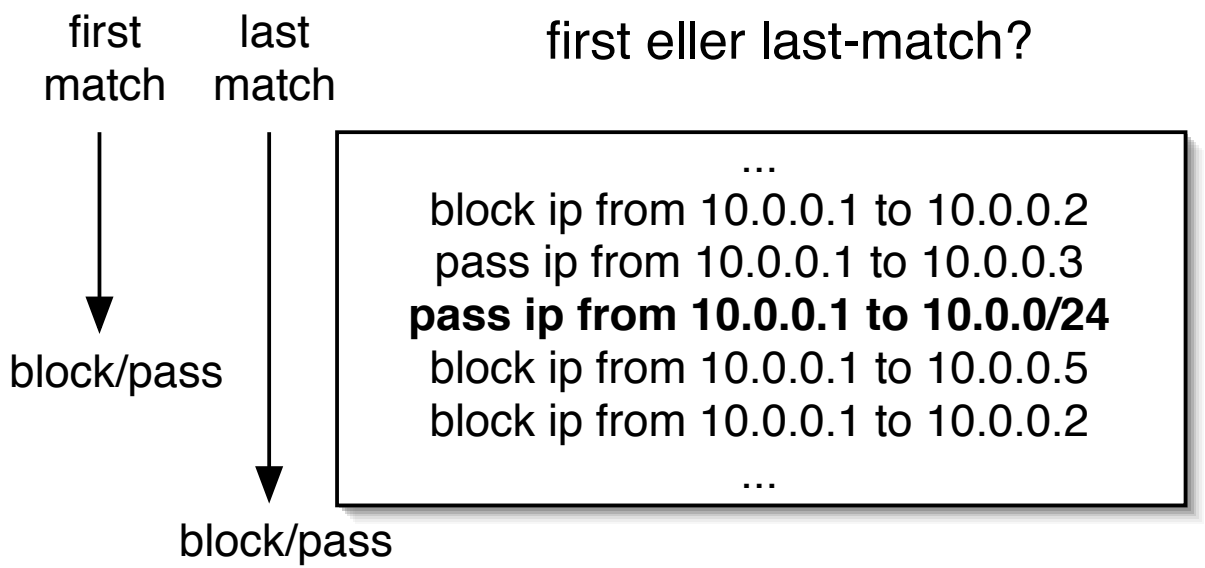
Det er ekstremt vigtigt at vide hvilken type firewall man bruger!

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

First match - IPFW

```
00100 16389  1551541 allow ip from any to any via lo0
00200     0        0 deny log ip from any to 127.0.0.0/8
00300     0        0 check-state
...

65435    36     5697 deny log ip from any to any
65535   865    54964 allow ip from any to any
```

Den sidste regel nås aldrig!

Last match - OpenBSD PF

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

block in
pass out keep state

pass quick on { lo $int_if }

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

Pakkerne markeres med block eller pass indtil sidste regel

nøgleordet *quick* afslutter match - god til store regelsæt

Firewalls og ICMP

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

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

Tillad ICMP types:

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

Firewall konfiguration

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

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

Bloker indefra og ud

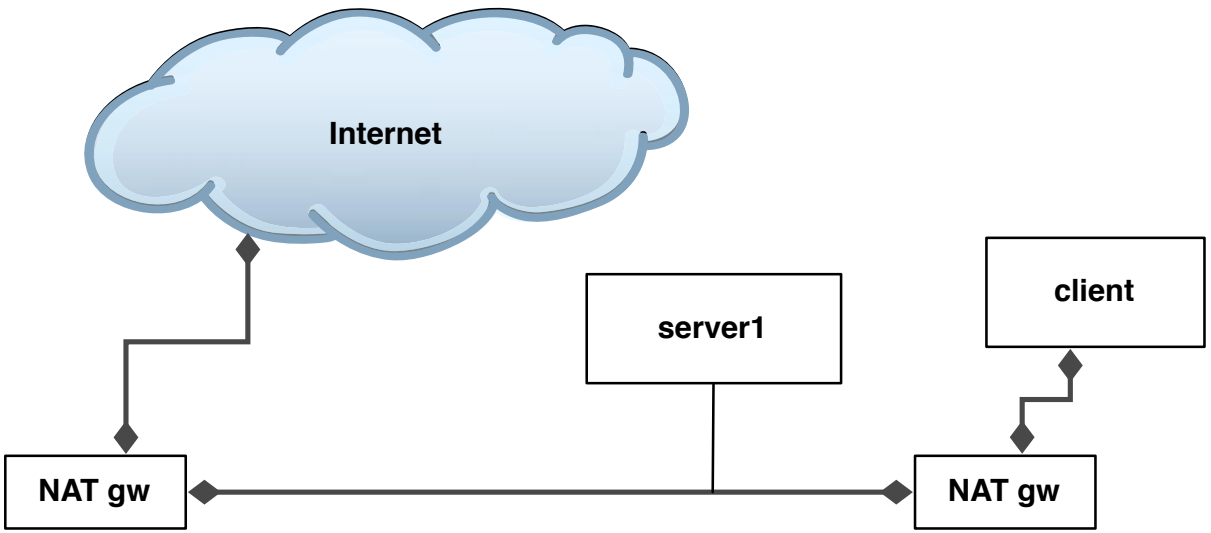
Der er porte og services som altid bør blokeres

Det kan være kendte sårbare services

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

Kendte problemer som minimum

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!

IPsec og Andre VPN features

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

Linux TCP SACK PANIC - CVE-2019-11477 et al

Kernel vulnerabilities, CVE-2019-11477, CVE-2019-11478 and CVE-2019-11479

Executive Summary
Three related flaws were found in the Linux kernel's handling of TCP networking. The most severe vulnerability could allow a remote attacker to trigger a kernel panic in systems running the affected software and, as a result, impact the system's availability.

The issues have been assigned multiple CVEs: CVE-2019-11477 is considered an Important severity, whereas CVE-2019-11478 and CVE-2019-11479 are considered a Moderate severity.

The first two are related to the Selective Acknowledgement (SACK) packets combined with Maximum Segment Size (MSS), the third solely with the Maximum Segment Size (MSS).

These issues are corrected either through applying mitigations or kernel patches. Mitigation details and links to RHSA advsories can be found on the RESOLVE tab of this article.

Source: https://access.redhat.com/security/vulnerabilities/tcpsack

Intrusion Detection

- networkbased intrusion detection systems (NIDS)
- item host based intrusion detection systems (HIDS)

Indicators of Compromise and Signatures

An IOC is any piece of information that can be used to objectively describe a network intrusion, expressed in a platform-independent manner. This could include a simple indicator such as the IP address of a command and control (C2) server or a complex set of behaviors that indicate that a mail server is being used as a malicious SMTP relay.

When an IOC is taken and used in a platform-specific language or format, such as a Snort Rule or a Bro-formatted file, it becomes part of a signature. A signature can contain one or more IOCs.

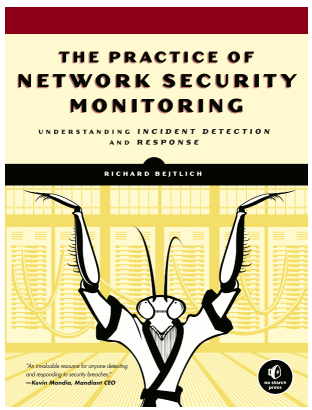
Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders

Reading Summary, False Positives

- True Positive (TP). An alert that has correctly identified a specific activity. If a signature was designed to detect a certain type of malware, and an alert is generated when that malware is launched on a system, this would be a true positive, which is what we strive for with every deployed signature.Indicators of Compromise and Signatures
- False Positive (FP). An alert has incorrectly identified a specific activity. If a signature was designed to detect a specific type of malware, and an alert is generated for an instance in which that malware was not present, this would be a false positive.
- True Negative (TN). An alert has correctly not been generated when a specific activity has not occurred. If a signature was designed to detect a certain type of malware, and no alert is generated without that malware being launched, then this is a true negative, which is also desirable. This is difficult, if not impossible, to quantify in terms of NSM detection.
- False Negative (FN). An alert has incorrectly not been generated when a specific activity has occurred.

Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders

Network Security Monitoring

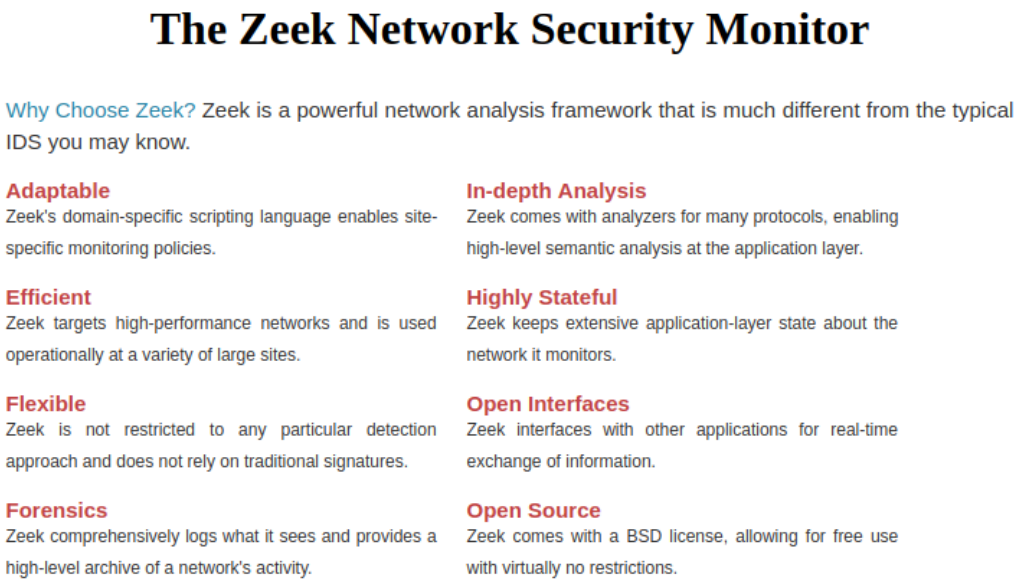


Network Security Monitoring (NSM) - monitoring networks for intrusions, and reacting to those

Recommend the book *The Practice of Network Security Monitoring Understanding Incident Detection and Response* by Richard Bejtlich July 2013

Example systems are Security Onion https://securityonion.net/ or
SELKS https://www.stamus-networks.com/open-source/

The Zeek Network Security Monitor



The Zeek Network Security Monitor is not a single tool, more of a powerful network analysis framework

Zeek is the tool formerly known as Bro, changed name in 2018. https://www.zeek.org/

Zeek IDS is

The Zeek Network Security Monitor

While focusing on network security monitoring, Zeek provides a comprehensive platform for more general network traffic analysis as well. Well grounded in more than 15 years of research, Zeek has successfully bridged the traditional gap between academia and operations since its inception.

https://www.Zeek.org/

Suricata IDS/IPS/NSM

Suricata is a high performance Network IDS, IPS and Network Security Monitoring engine.

http://suricata-ids.org/ http://openinfosecfoundation.org

I have a whole 4-hour workshop - feel free to look at it:
Suricata, Zeek og DNS Capture
https://github.com/kramse/security-courses/tree/master/courses/networking/suricatazeek-workshop

Kali Linux the new backtrack

Kali http://www.kali.org/

Use the documentation on Kali website for installation

Use tools web sites for documentation about tools, like https://nmap.org/ and http://www.hping.org/

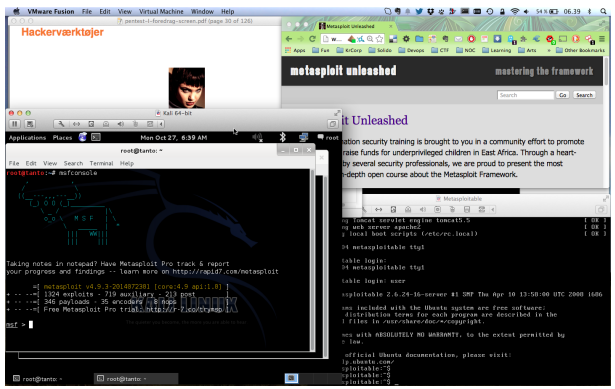
Kali

Almost 200.000 youtube videos about "kali hack"

You can learn these tools from their respective home pages:
Like http://nmap.org, http://aircrack-ng.org

The main site helps with install and VM tools Kali http://www.kali.org/

Hackerlab setup



- I recommend getting a hackerlab running on your laptop
- Hardware: modern laptop which has CPU virtualization
  Dont forget to check BIOS settings for virtualization
- Software: your favorite OS: Windows, Mac, Linux
- Virtualization software: VMware, Virtual box, HyperV
- Hacker software: Kali as a Virtual Machine https://www.kali.org/

Availability and Network flooding attacks

- Our book spends some time on SYN and other flooding attacks
- SYN flood is the most basic and very common on the internet towards 80/tcp and 443/tcp
- ICMP and UDP flooding are the next targets
- Supporting litterature is TCP Synfloods - an old yet current problem, and improving pf's response to it, Henning Brauer, BSDCan 2017
- All of them try to use up some resources
- Memory space in specific sections of the kernel, TCP state, firewalls state, number of concurrent sessions/connections
- interrupt processing of packets - packets per second
- CPU processing in firewalls, pps
- CPU processing in server software
- Bandwidth - megabits per second mbps

There is a presentation about DDoS protection with low level technical measures to implement at
https://github.com/kramse/security-courses/tree/master/presentations/network/introduction-ddos-testing

Simulating DDoS packets

A minimal introduction workshop teaching people how to produce DDoS simulation traffic - usefull for testing their own infrastructures.

We will have a server connected on a switch with multiple 1Gbit port for attackers. Attackers will be connected through 1Gbit ports using USB Ethernet - we have loaners.

Work together to produce enough to take down this server!

WHILE attack is ongoing there will be both the possibility to monitor traffic, monitor port, and decide on changes to prevent the attacks from working.

Common DDoS attack types

We will work through common attack types, like:

- TCP SYN flooding
- TCP other flooding
- UDP flooding NTP, etc.
- ICMP flooding
- Misc - stranger attacks and illegal combinations of flags etc.

then we will discuss which changes to environment could be implemented.

You will go away from this with tools for producing packets, hping3 and some configurations for protecting - PF rules, switch rules, server firewall rules.

hping3 packet generator

```
usage: hping3 host [options]
  -i  --interval  wait (uX for X microseconds, for example -i u1000)
      --fast      alias for -i u10000 (10 packets for second)
      --faster    alias for -i u1000 (100 packets for second)
      --flood     sent packets as fast as possible. Don't show replies.
...
hping3 is fully scriptable using the TCL language, and packets
can be received and sent via a binary or string representation
describing the packets.
```

- Hping3 packet generator is a very flexible tool to produce simulated DDoS traffic with specific charateristics
- Home page: http://www.hping.org/hping3.html
- Source repository https://github.com/antirez/hping

My primary DDoS testing tool, easy to get specific rate pps

t50 packet generator

```
root@cornerstone03:~# t50 -?
T50 Experimental Mixed Packet Injector Tool 5.4.1
Originally created by Nelson Brito <nbrito@sekure.org>
Maintained by Fernando Mercês <fernando@mentebinaria.com.br>

Usage: T50 <host> [/CIDR] [options]

Common Options:
    --threshold NUM        Threshold of packets to send     (default 1000)
    --flood                This option supersedes the 'threshold'
...
6. Running T50 with '--protocol T50' option, sends ALL protocols sequentially.
root@cornerstone03:~# t50 -? | wc -l
264
```

- T50 packet generator, another high speed packet generator can easily overload most firewalls by producing a randomized traffic with multiple protocols like IPsec, GRE, MIX
  home page: http://t50.sourceforge.net/resources.html

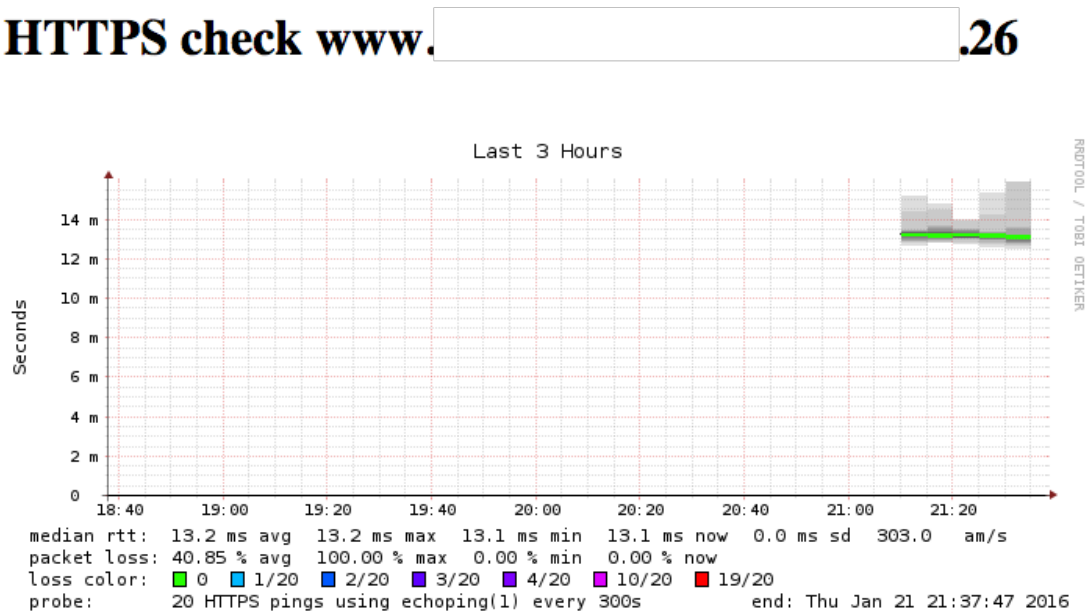
Extremely fast and breaks most firewalls when flooding, easy 800k pps/400Mbps

Process: monitor, attack, break, repeat

- Pre-test: Monitoring setup - from multiple points
- Pre-test: Perform full Nmap scan of network and ports
- Start small, run with delays between packets
- Turn up until it breaks, decrease delay - until using `--flood`
- Monitor speed of attack on your router interface pps/bandwidth
- Give it maximum speed
  `hping3 --flood -1` and `hping3 --flood -2`
- Have a common chat with network operators/customer to talk about symptoms and things observed
- Any information resulting from testing is good information

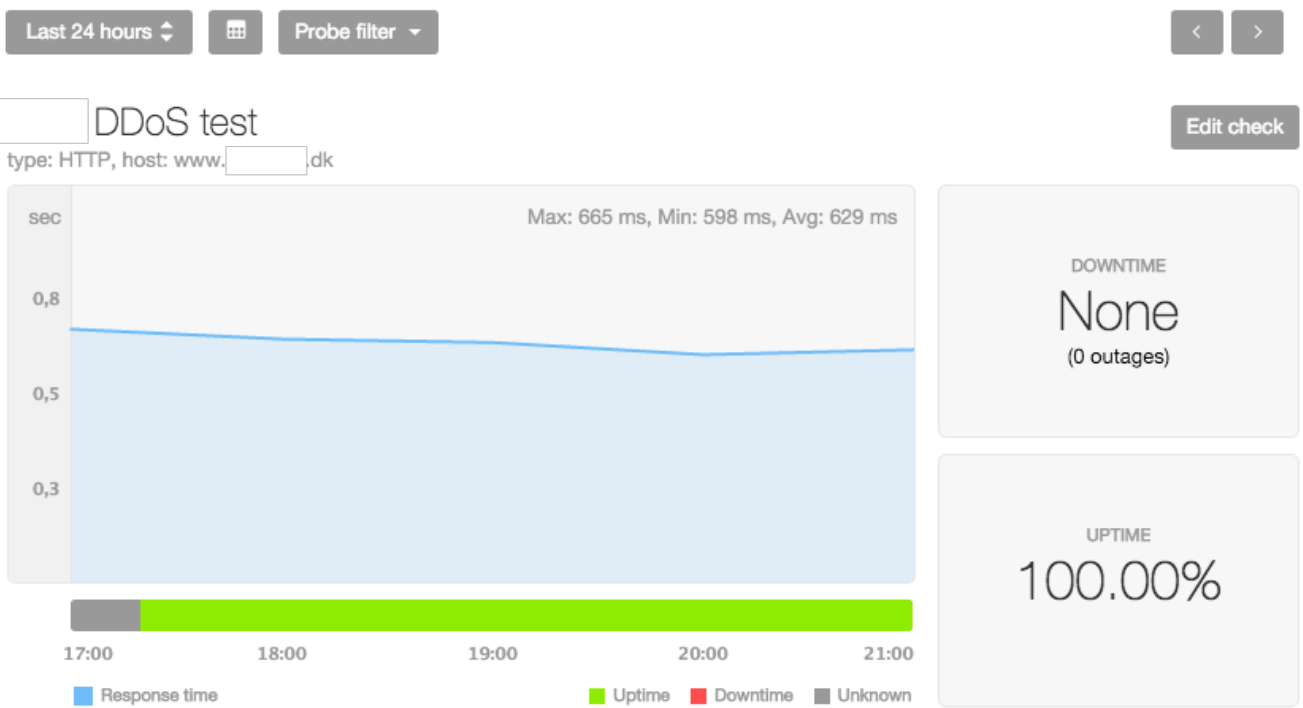
Ohh we lost our VPN into the environment, ohh the fw console is dead

Before testing: Smokeping



Before DDoS testing use Smokeping software

Before testing: Pingdom



Another external monitoring from Pingdom.com

Running full port scan on network

```
# export CUST_NET="192.0.2.0/24"
# nmap -p 1-65535 -A -oA full-scan $CUST_NET
```

Performs a full port scan of the network, all ports

Saves output in "all formats"normal, XML, and grepable formats

Goal is to enumerate the ports that are allowed through the network.

Note: This command is pretty harmless, if something dies, then it is *vulnerable to normal traffic* - and should be fixed!

Running Attacks with hping3

```
# export CUST_IP=192.0.2.1
# date;time hping3 -q -c 1000000  -i u60 -S -p 80  $CUST_IP


# date;time hping3 -q -c 1000000  -i u60 -S -p 80  $CUST_IP
Thu Jan 21 22:37:06 CET 2016
HPING 192.0.2.1 (eth0 192.0.2.1): S set, 40 headers + 0 data bytes

--- 192.0.2.1 hping statistic ---
1000000 packets transmitted, 999996 packets received, 1% packet loss
round-trip min/avg/max = 0.9/7.0/1005.5 ms

real 1m7.438s
user 0m1.200s
sys 0m5.444s
```

Dont forget to do a killall hping3 when done ☺

Recommendations During Test

Run each test for at least 5 minutes, or even 15 minutes
Some attacks require some build-up before resource run out

Take note of any change in response, higher latency, lost probes

If you see a change, then re-test using the same parameters, or a little less first

We want to know the approximate level where it breaks

If you want to change environment, then wait until all scenarios tested

Comparable to real DDoS?

Tools are simple and widely available but are they actually producing same result as high-powered and advanced criminal botnets. We can confirm that the attack delivered in this test is, in fact, producing the traffic patterns very close to criminal attacks in real-life scenarios.

- We can also monitor logs when running a single test-case
- Gain knowledge about supporting infrastructure
- Can your syslog infrastructure handle 800.000 events in $<$ 1 hour?

Running the tools

A basic test would be:

- TCP SYN flooding
- TCP other flags, PUSH-ACK, RST, ACK, FIN
- ICMP flooding
- UDP flooding
- Spoofed packets src=dst=target ☺
- Small fragments
- Bad fragment offset
- Bad checksum
- Be creative
- Mixed packets - like `t50 --protocol T50`
- Perhaps esoteric or unused protocols, GRE, IPSec

Test-cases / Scenarios

The minimal run contains at least these:

- SYN flood: `hping3 -q -c 1000000 -i u60 -S -p 80 $CUST_IP &`
- SYN+ACK: `hping3 -q -c 1000000 -i u60 -S -A -p 80 $CUST_IP &`
- ICMP flood: `hping3 -q -c --flood -1 $CUST_IP &`
- UDP flood: `hping3 -q -c --flood -1 $CUST_IP &`

Vary the speed using the packet interval `-i u60` up/down

Use flooding with caution, runs max speeeeeeeeeeeed ☺

TCP testing use a port which is allowed through the network, often 80/443

Focus on attacks which are hard to block, example TCP SYN must be allowed in

Also if you found devices like routers in front of environment

```
hping3 -q -c 1000000 -i u60 -S -p 22 $ROUTER_IP
hping3 -q -c 1000000 -i u60 -S -p 179 $ROUTER_IP
```

Test-cases / Scenarios, continued Spoof Source

Spoofed packets src=dst=target ☺

Flooding with spoofed packet source, within customer range

```
-a --spoof hostname
    Use this option in order to set a fake IP  source  address,  this
    option ensures that target will not gain your real address.

hping3 -q --flood -p 80 -S -a $CUST_IP $CUST_IP
```

Preferably using a test-case you know fails, to see effect

Still amazed how often this works

BCP38 anyone!

Test-cases / Scenarios, continued Small Fragments

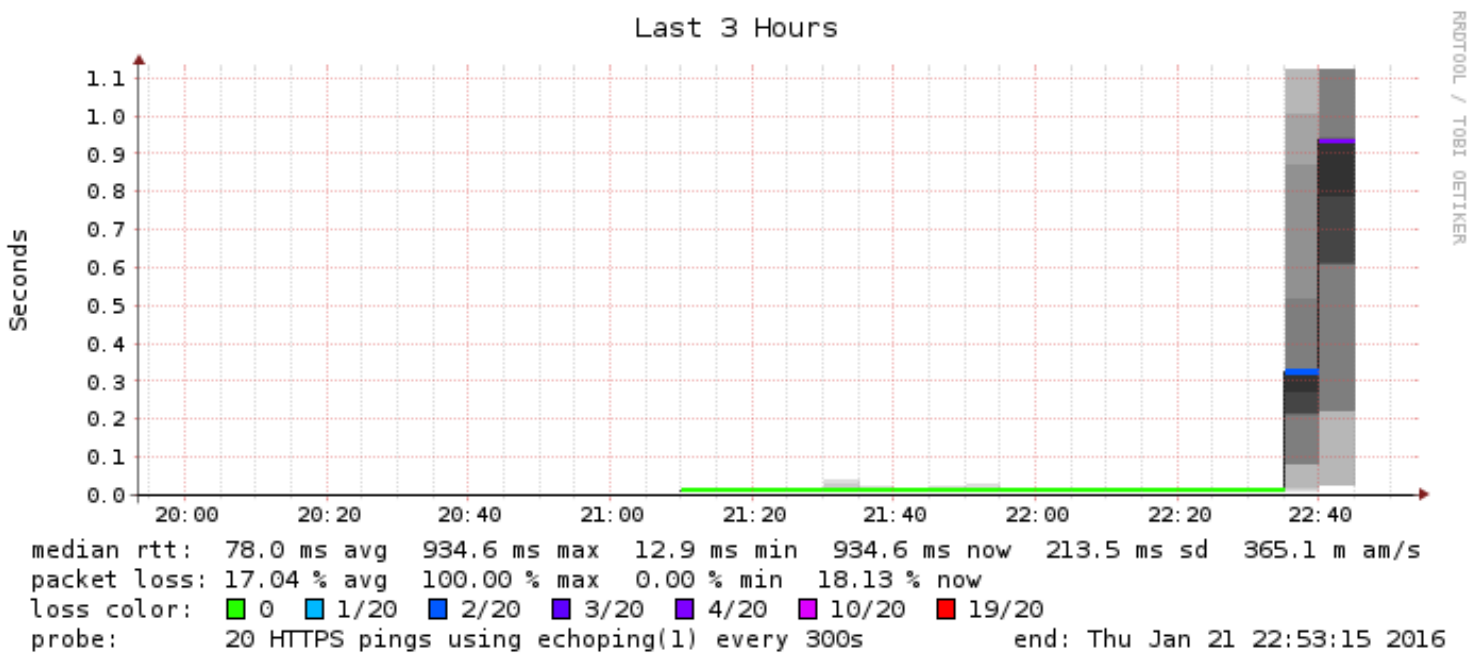
Using the built-in option -f for hping

```
-f --frag
    Split  packets  in more fragments, this may be useful in order to test IP
    stacks fragmentation performance and to test if some packet filter is  so
    weak  that  can  be  passed using tiny fragments (anachronistic). Default
     'virtual mtu' is 16 bytes. see also --mtu option.
```

```
hping3 -q --flood -p 80 -S -f $CUST_IP
```

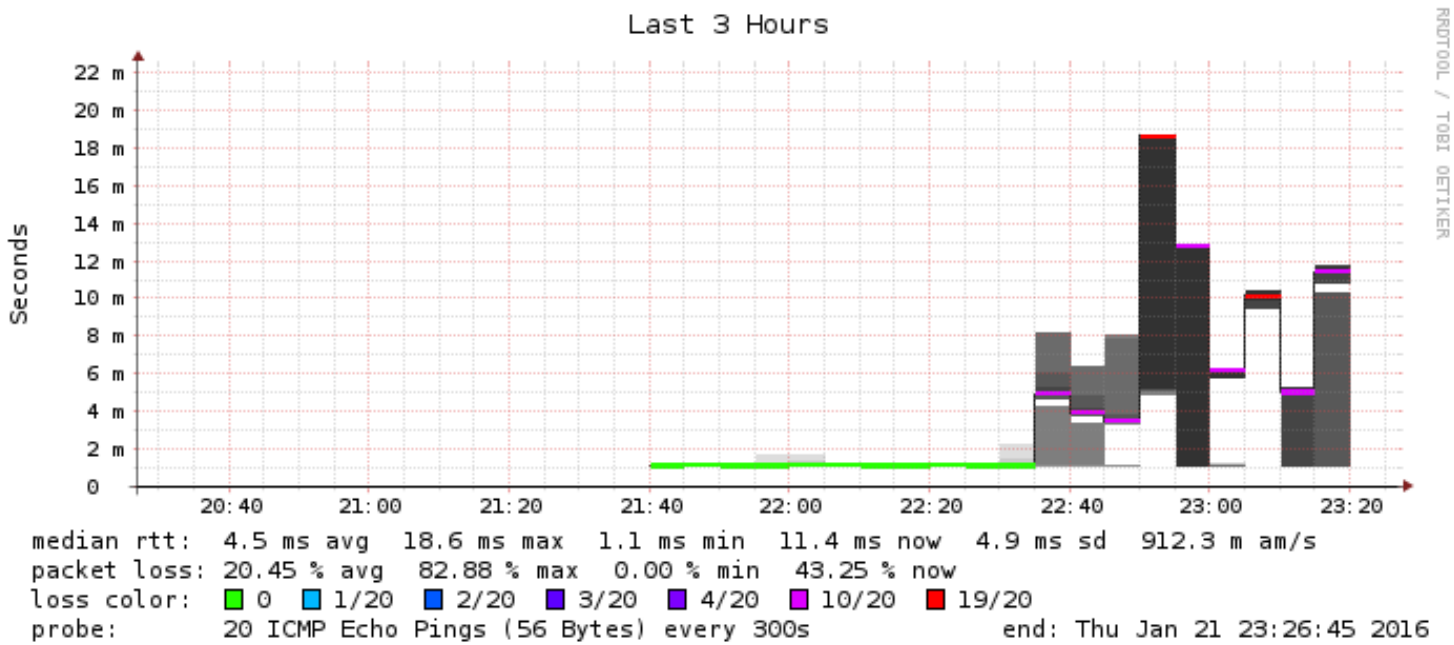
Similar process with bad checksum and Bad fragment offset

Rocky Horror Picture Show - 1



Really does it break from 50.000 pps SYN attack?

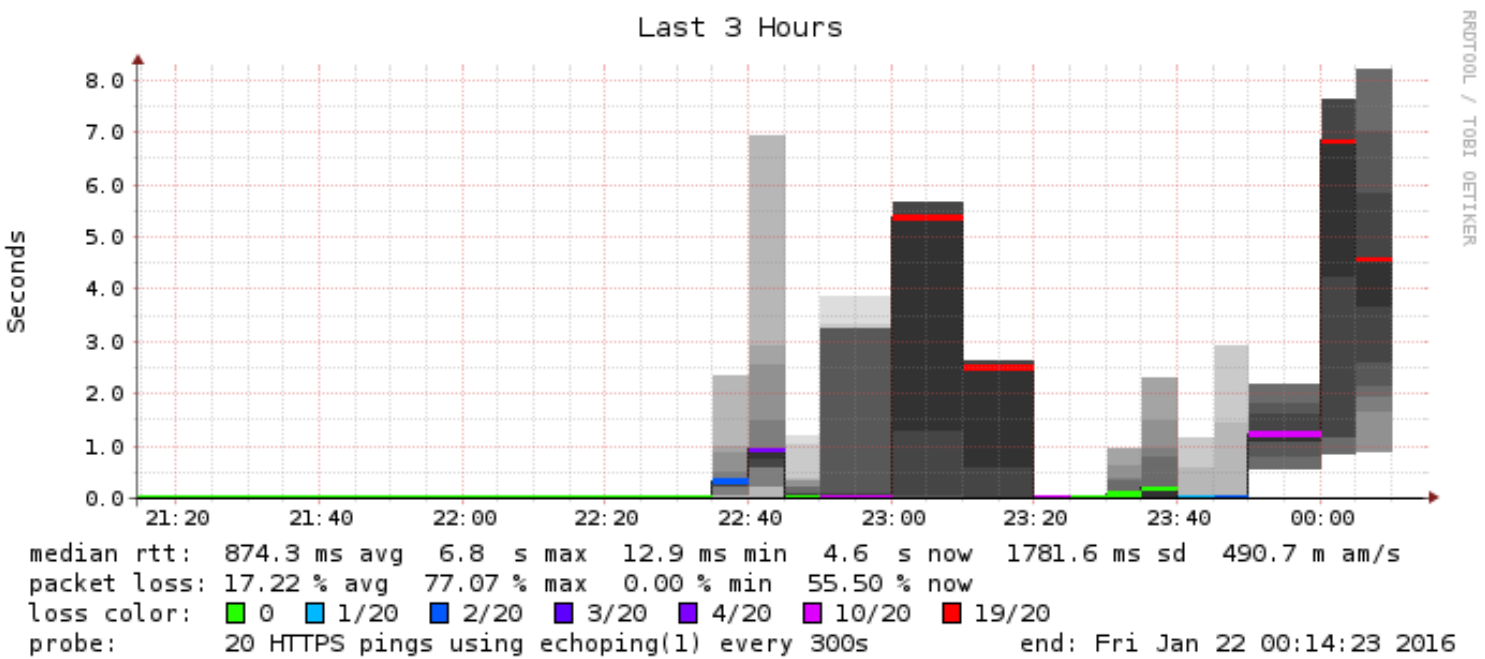
Rocky Horror Picture Show - 2



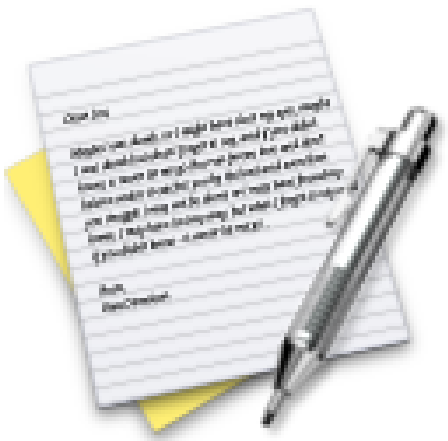
Oh no 500.000 pps UDP attacks work?

Rocky Horror Picture Show - 3

Oh no spoofing attacks work?



Exercise

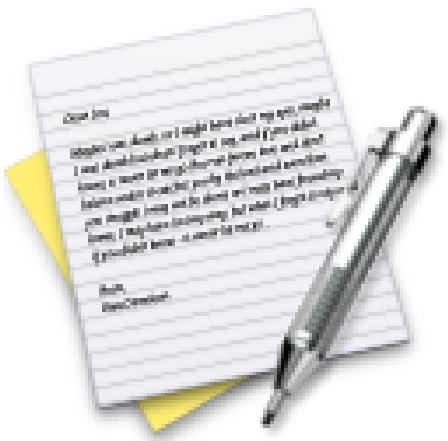


Now lets do the exercise

Execute nmap TCP and UDP port scan 20 min

which is number **34** in the exercise PDF.

Exercise

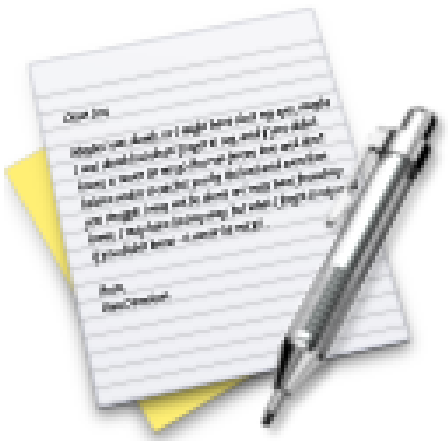


Now lets do the exercise

Discover active systems ping and port sweep 15 min

which is number **35** in the exercise PDF.

Exercise



Now lets do the exercise

TCP SYN flooding 30min

which is number **36** in the exercise PDF.

Exercise booklet contains some bonus exercises, feel free to try them at home

Improvements seen after testing

Turning off unneeded features - free up resources

Tuning sesions, max sessions src / dst

Tuning firewalls, max sessions in half-open state, enabling services

Tuning network, drop spoofed src from inside net ☺

Tuning network, can follow logs, manage network during attacks

...

And organisation has better understanding of DDoS challenges

Including vendors, firewall consultants, ISPs etc.

After tuning of **existing devices/network** improves results 10-100 times

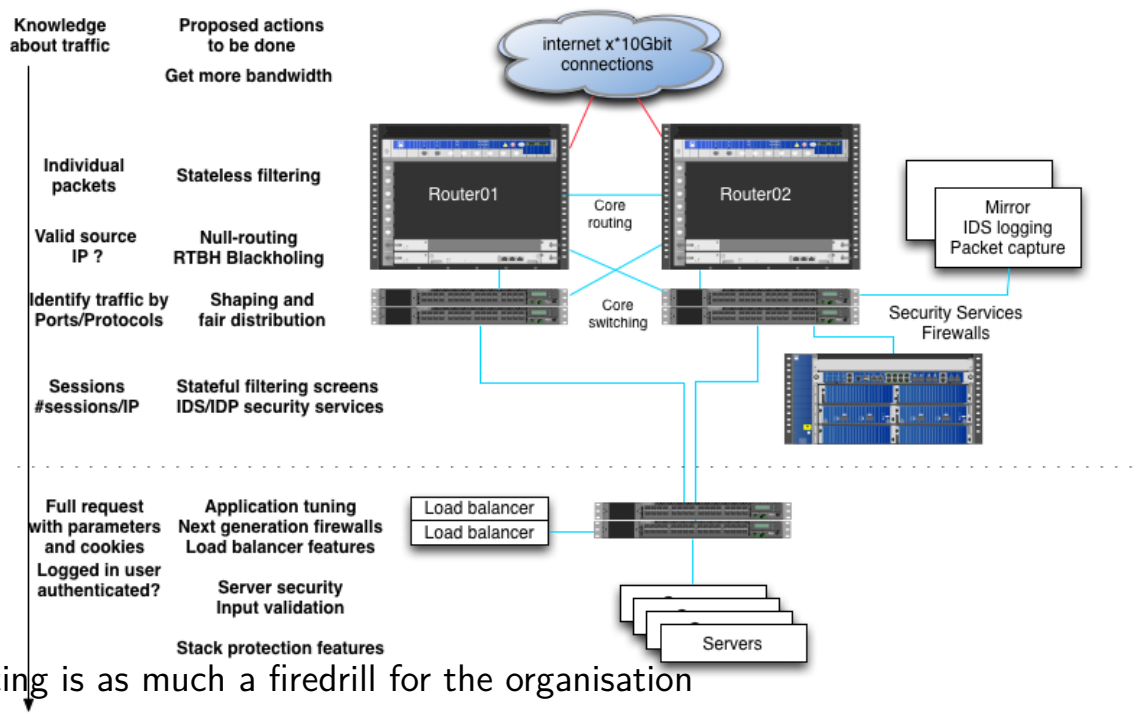
Conclusion

.

You really should try testing

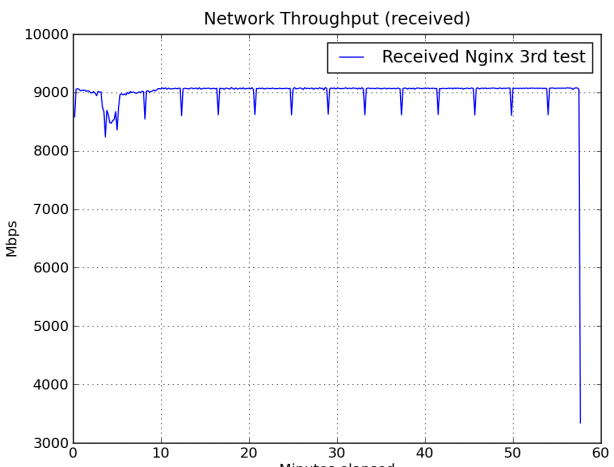
Investigate your existing devices
all of them, RTFM, upgrade firmware

Choose which devices does which
part - discard early to free resources
for later devices to dig deeper



And dont forget that DDoS testing is as much a firedrill for the organisation

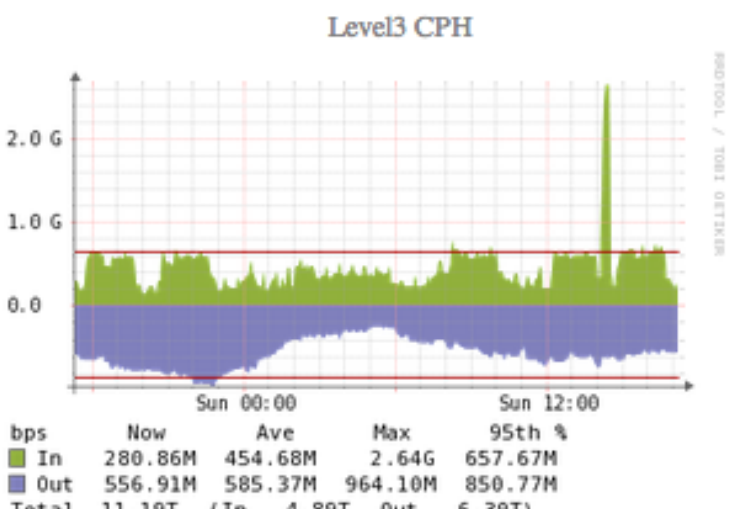
More application testing

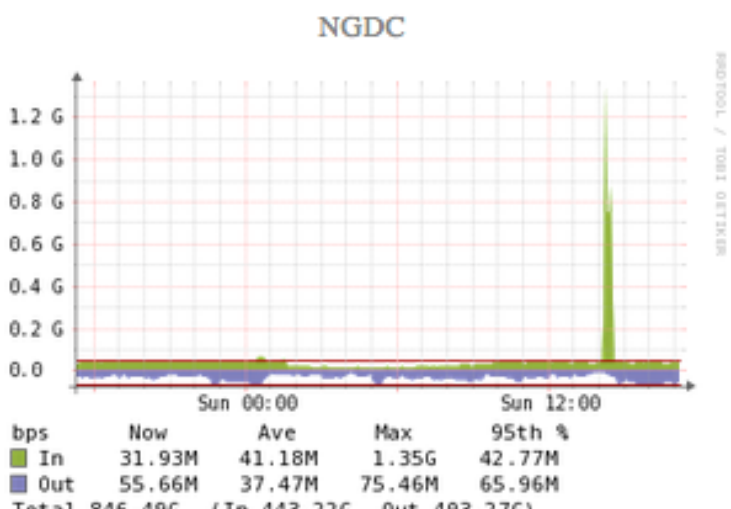


We covered only lower layers - but helpful layer 7 testing programs exist

Tsung can be used to stress HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP and Jabber/XMPP servers http://tsung.erlang-projects.org/

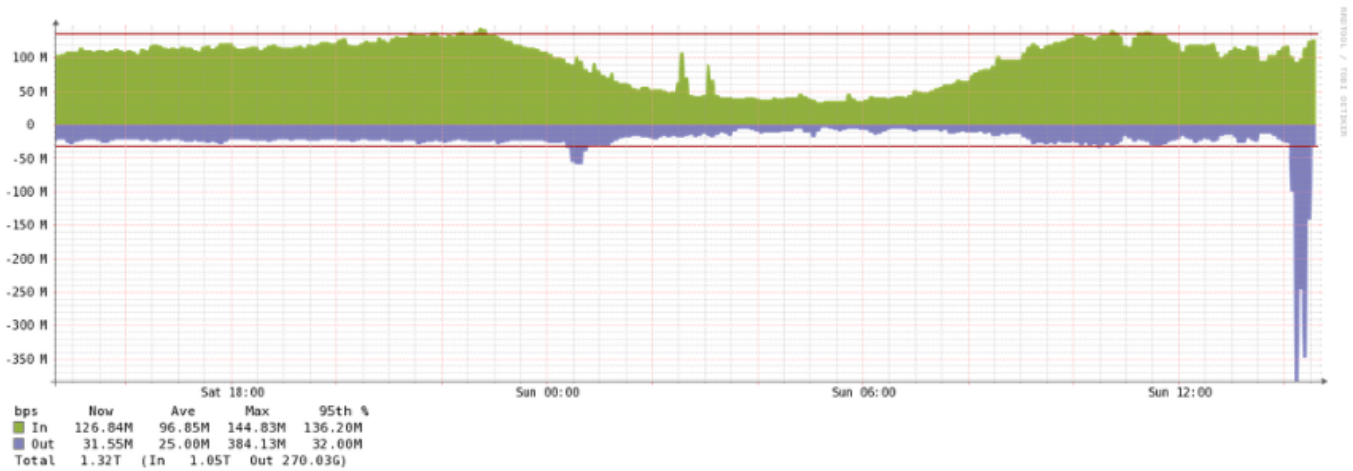
DDoS traffic before filtering





Only two links shown, at least 3Gbit incoming for this single IP

DDoS traffic after filtering



Link toward server (next level firewall actually) about 350Mbit outgoing

Stateless firewall filter throw stuff away

```
hlk@MX-CPH-02> show configuration firewall filter all | no-more
/* This is a static sample, perhaps better to use BGP flowspec and RTBH */
term edgeblocker {
    from {
        source-address {
            84.180.xxx.173/32;
...
            87.245.xxx.171/32;
        }
        destination-address {
            91.102.91.16/28;
        }
        protocol [ tcp udp icmp ];
    }
    then {
        count edge-block;
        discard;
    }
}
```

Hint: can also leave out protocol and then it will match all protocols

Stateless firewall filter limit protocols

```
term limit-icmp {
    from {
        protocol icmp;
    }
    then {
        policer ICMP-100M;
        accept;
    }
}
term limit-udp {
    from {
        protocol udp;
    }
    then {
        policer UDP-1000M;
        accept;
    }
}
```

Routers have extensive Class-of-Service (CoS) tools today

Strict filtering for some servers, still stateless!

```
term some-server-allow {
    from {
        destination-address {
            109.238.xx.0/xx;
        }
        protocol tcp;
        destination-port [ 80 443 ];
    }
    then accept;
}
term some-server-block-unneeded {
    from {
        destination-address {
            109.238.xx.0/xx;
        }
        protocol-except icmp;
    }
    then discard;
}
```

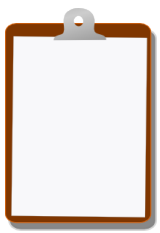
Wut - no UDP, yes UDP service is not used on these servers

Firewalls - screens, IDS like features

When you know regular traffic you can decide *normal settings*:

```
hlk@srx-kas-05# show security screen ids-option untrust-screen
icmp {
    ping-death;
}
ip {
    source-route-option;
    tear-drop;
}
tcp {    Note: UDP flood setting also exist
    syn-flood {
        alarm-threshold 10024;
        attack-threshold 200;
        source-threshold 10024;
        destination-threshold 2048;
        timeout 20;
    }
    land;
}
```

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools