Welcome to

Traffic Inspection and Firewalls

Communication and Network Security 2019

Henrik Lund Kramshøj hlk@zencurity.com

Slides are available as PDF, kramse@Github

3-Traffic-Inspection-and-Firewalls.tex in the repo security-courses

Plan for today

Subjects

- Traffic inspection and firewalls
- Generic IP Firewalls stateless filtering vs stateful inspection
- Next Generation firewalls, Deep Packet Inspection
- IEEE 802.1q VLAN
- Common countermeasures in firewalls

Exercises

- Nping
- Try pcap-diff
- Nmap scanning basics

firewalls

Indeholder typisk:

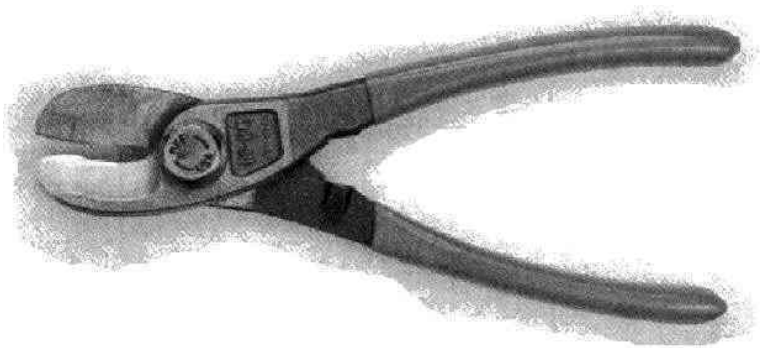
- Grafisk brugergrænseflade til konfiguration - er det en fordel?
- TCP/IP filtermuligheder - pakkernes afsender, modtager, retning ind/ud, porte, protokol, ...
- kun IPv4 for de kommercielle firewalls
- både IPv4 og IPv6 for Open Source firewalls: IPF, OpenBSD PF, Linux firewalls, ...
- foruddefinerede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

En router med Access Control Lists - 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
```

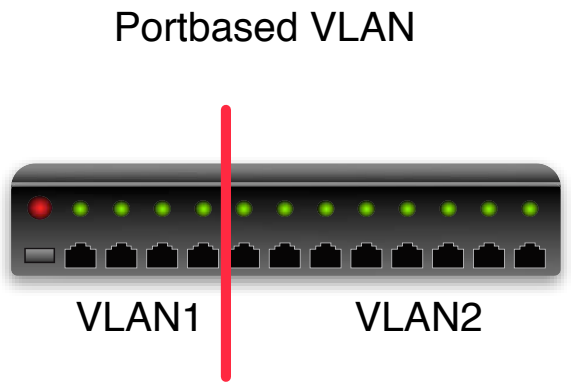
netdesign - med firewalls



- 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: `http://www.ranum.com/pubs/a1fwall/` The ULTIMATELY Secure Firewall

VLAN Virtual LAN



Nogle switche tillader at man opdeler portene

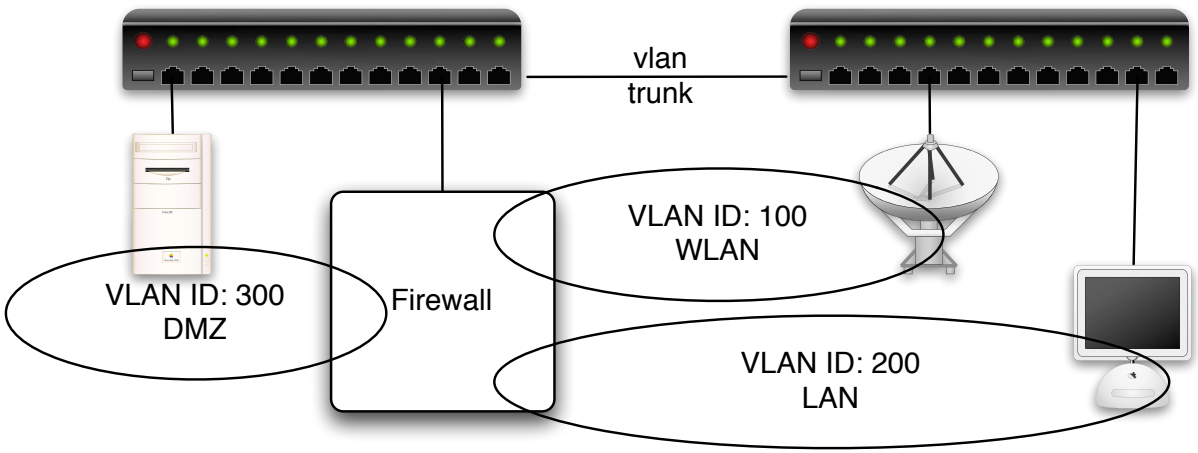
Denne opdeling kaldes VLAN og portbaseret er det mest simple

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



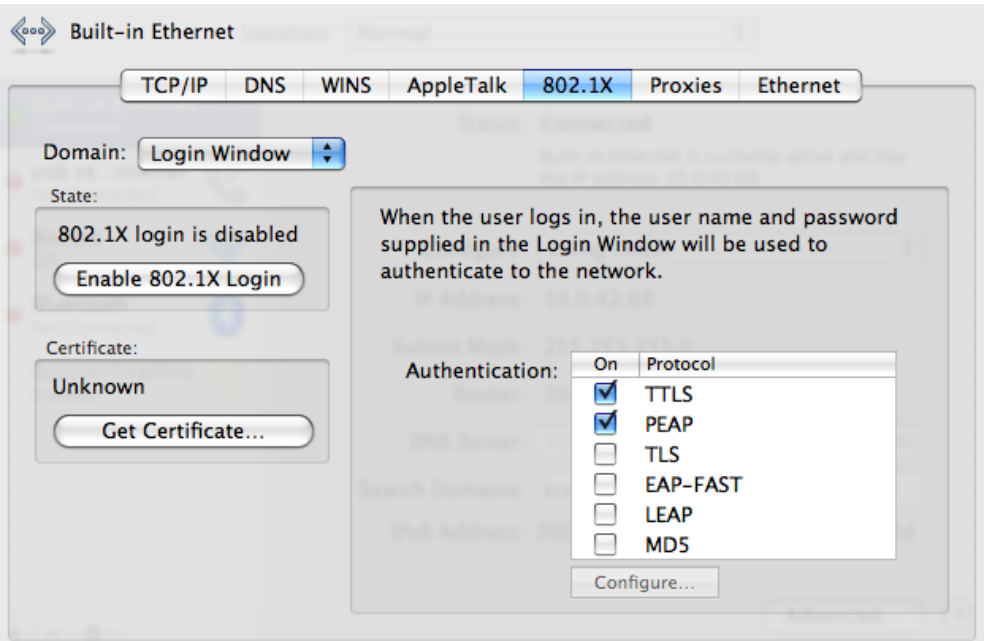
Med 802.1q tillades VLAN tagging på Ethernet niveau

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

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

IEEE 802.1x Port Based Network Access Control



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

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

802.1x og andre teknologier

802.1x i forhold til MAC filtrering giver væsentlige fordele

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

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

Hvad er en firewall

En firewall er noget som blokerer traffik på Internet

En firewall er noget som tillader traffik på Internet

Firewallrollen idag

Idag skal en firewall være med til at:

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

Det kræver overblik!

firewalls

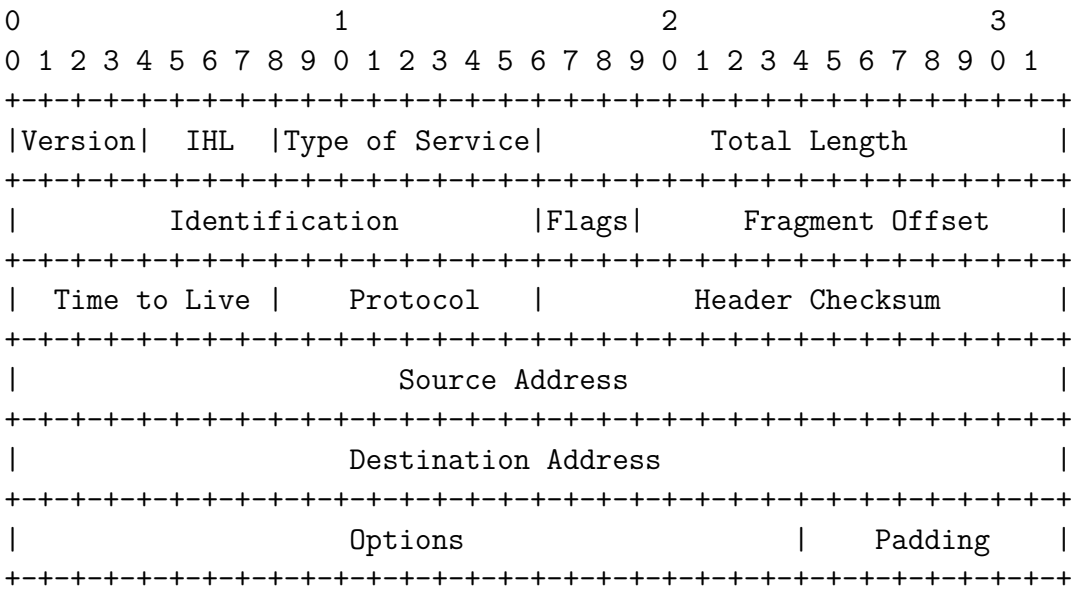
Basalt set et netværksfilter - det yderste fæstningsværk

Indeholder typisk:

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

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

Packet filtering



Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste statefull 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`

Ovenstående er dem som jeg oftest ser ude hos mine kunder

Open source baserede firewalls

- Linux firewalls - fra begyndelsen til det nuværende netfilter til kerner version 2.4 og 2.6 `http://www.netfilter.org`
- Firewall GUIs ovenpå Linux - mange! 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 Zyxel 10 en hardware firewall og en Soekris med OpenBSD for en software firewall!

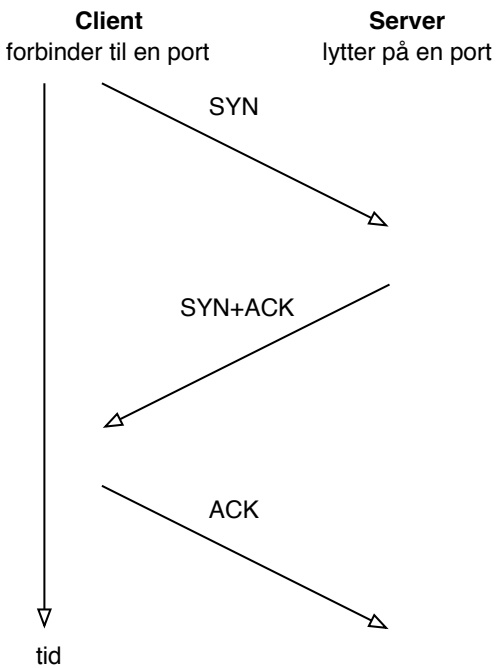
Det er efter min mening et marketingtrick

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

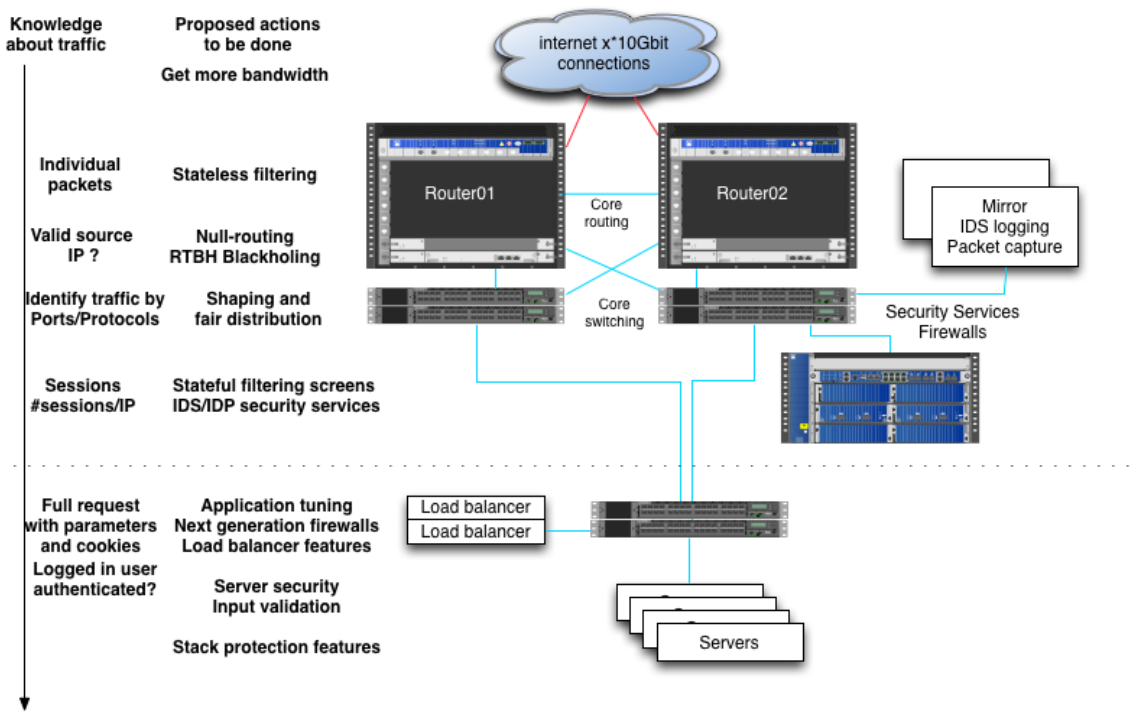
TCP three way handshake

.

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

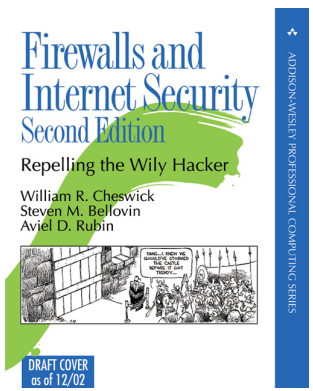


Firewall er ikke alene



Forsvaret er som altid - flere lag af sikkerhed!

Firewall historik



Firewalls har været kendt siden starten af 90'erne

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

Bogen udkom i 1994 men kan stadig anbefales

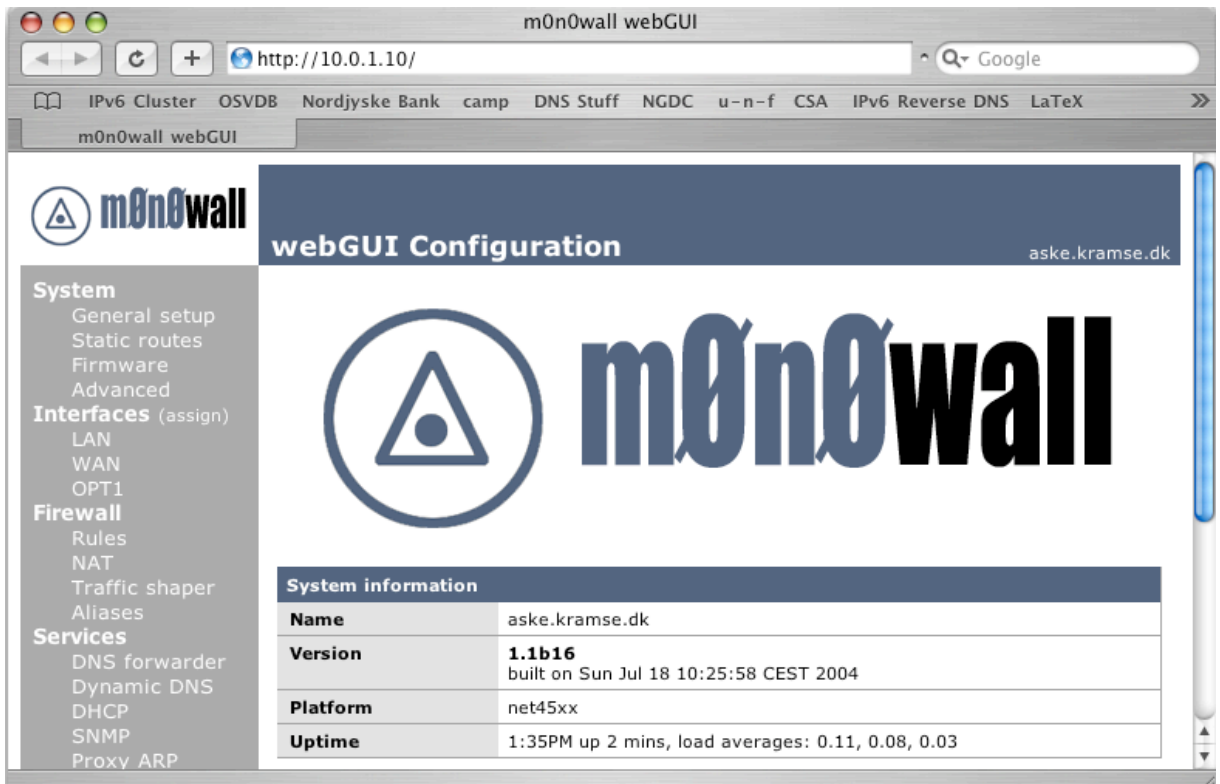
An Evening with Berferd

Artikel om en hacker der lokkes, vurderes, overvåges

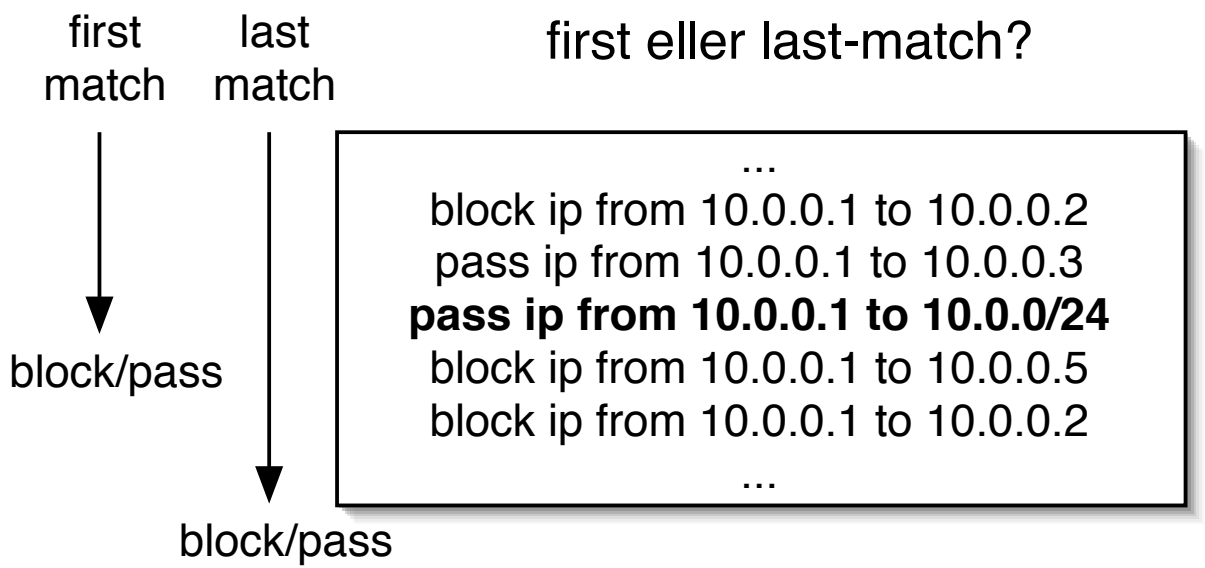
Et tidligt eksempel på en honeypot

Idag anbefales The Honeynet Project hvis man vil vide mere
`http://www.honeynet.org`

m0n0wall


m0n0wall webGUI
http://10.0.1.10/
Google
IPv6 Cluster
OSVDB
Nordjyske Bank
camp
DNS Stuff
NGDC
u-n-f
CSA
IPv6 Reverse DNS
LaTeX
m0n0wall webGUI
m0n0wall
webGUI Configuration
aske.kramse.dk
System
General setup
Static routes
Firmware
Advanced
Interfaces (assign)
LAN
WAN
OPT1
Firewall
Rules
NAT
Traffic shaper
Aliases
Services
DNS forwarder
Dynamic DNS
DHCP
SNMP
Proxy ARP
m0n0wall
System information
Name
aske.kramse.dk
Version
1.1b16
built on Sun Jul 18 10:25:58 CEST 2004
Platform
net45xx
Uptime
1:35PM up 2 mins, load averages: 0.11, 0.08, 0.03

First or Last match firewall?



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

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

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

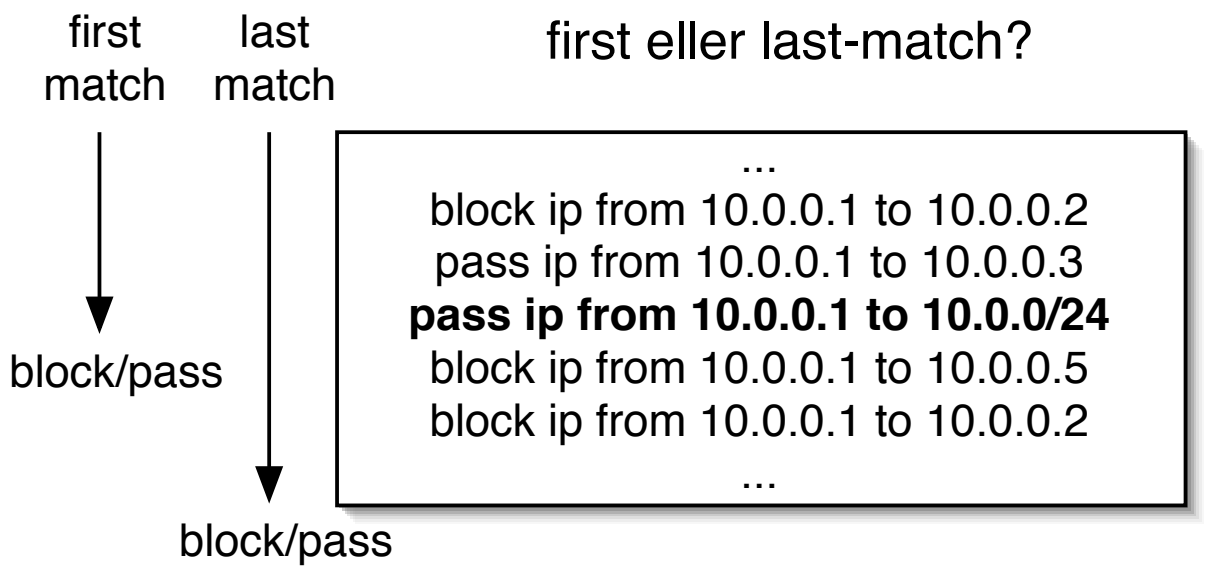
Det er ekstremt vigtigt at vide hvilken type firewall man bruger!

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

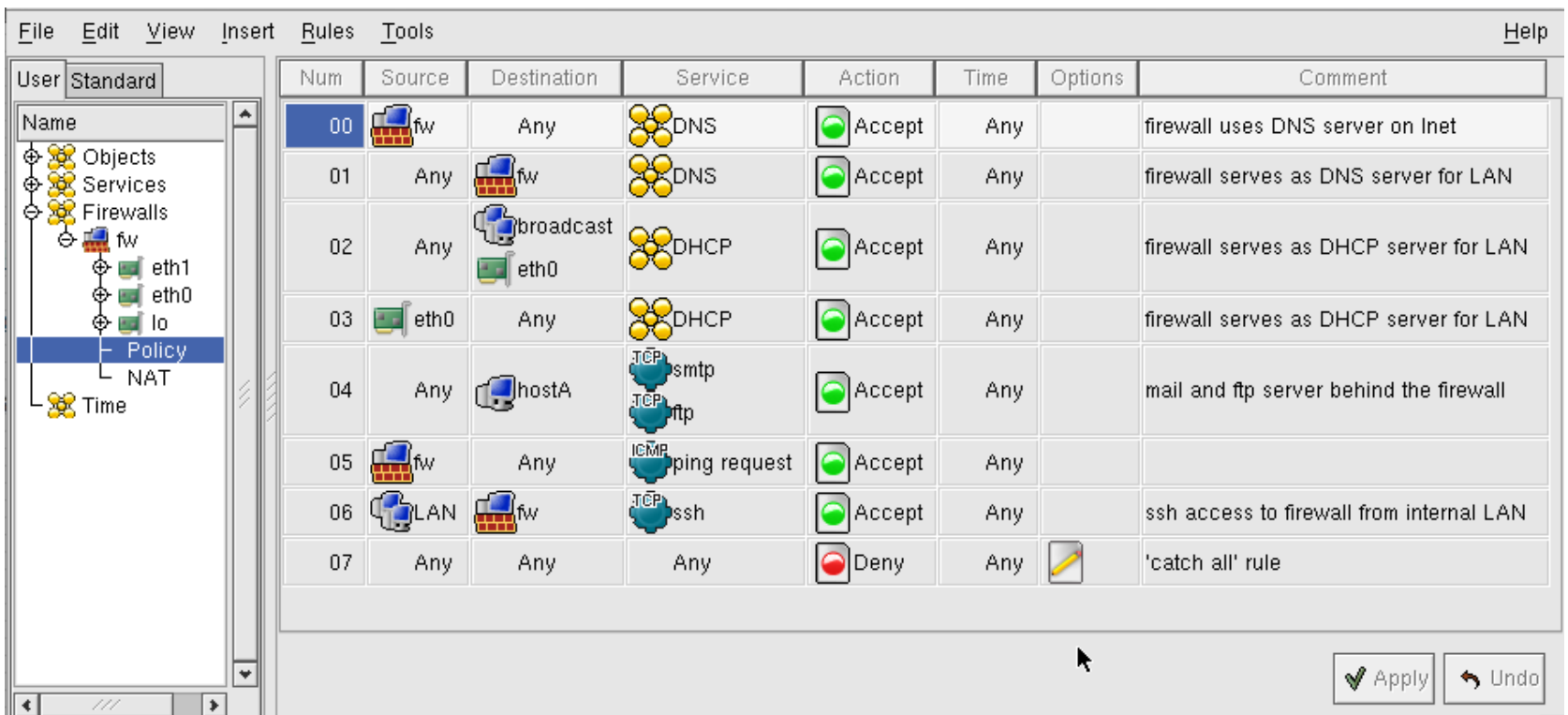
nøgleordet *quick* afslutter match - god til store regelsæt

Linux iptables/netfilter eksempel

```
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

NB: husk at aktivere IP forwarding

Firewall GUI



Kilde: billede fra `http://www.fwbuilder.org`

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

Firewall konfiguration

Den bedste firewall konfiguration starter med:

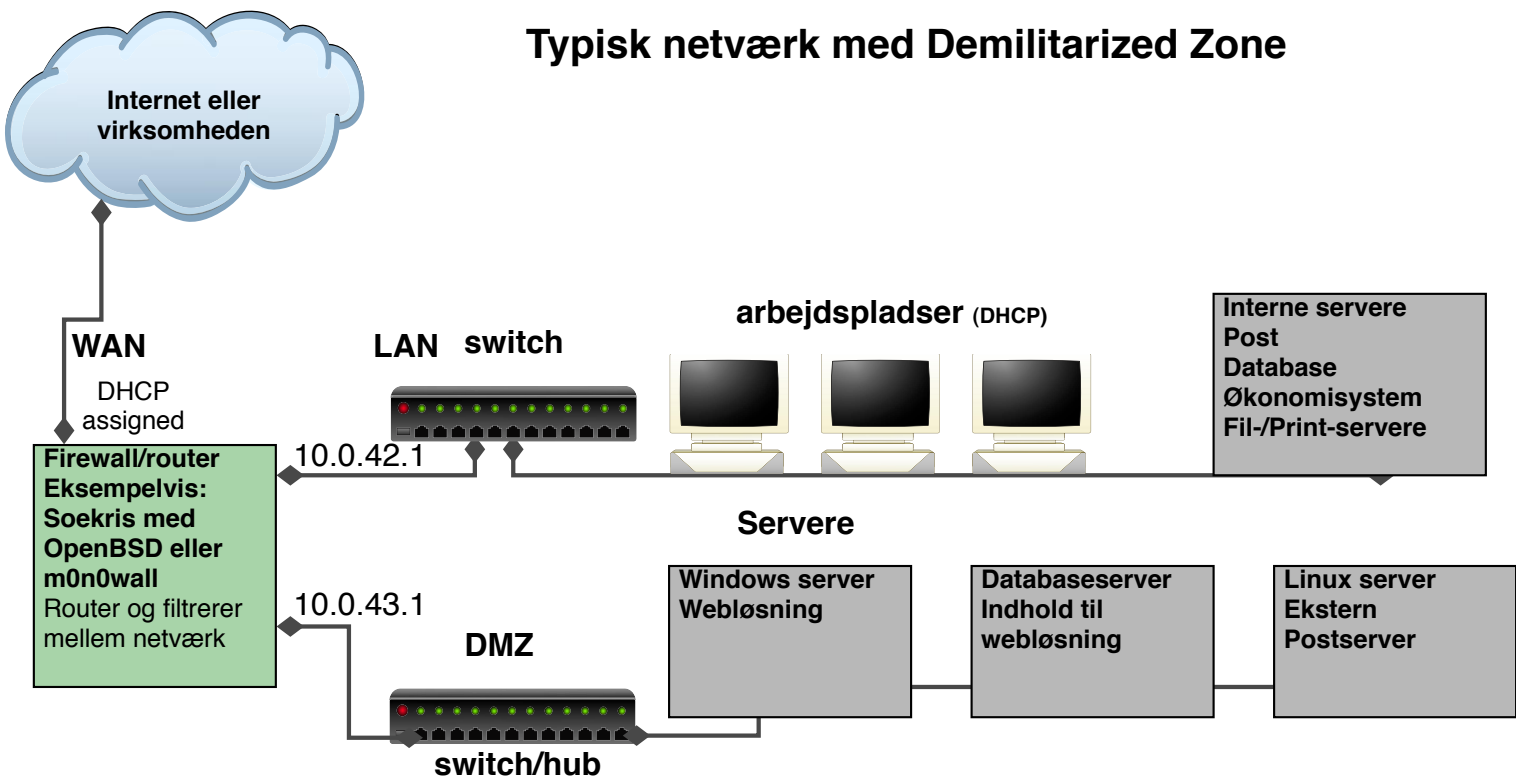
- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

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

En typisk firewall konfiguration



Opdeling i separate netværkssegmenter!

Firewallværktøjer

Der benyttes på kurset en del værktøjer:

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

Specielle features

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

Proxy servers

Filtrering på højere niveauer i OSI modellen er muligt

Idag findes proxy applikationer til de mest almindelige funktioner

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

NB: nogle protokoller egner sig ikke til proxy servere

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

IPsec og Andre VPN features

De fleste firewalls giver mulighed for at lave krypterede tunneler

Nyttigt til fjernkontorer der skal have usikker traffik henover usikre netværk som Internet

Konceptet kaldes Virtual Private Network VPN

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

Portscan, TCP, UDP og ICMP

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

Basal Portscanning

Hvad er portscanning

afprøvning af alle porte fra 0/1 og op til 65535

målet er at identificere åbne porte - sårbare services

typisk TCP og UDP scanning

TCP scanning er ofte mere pålidelig end UDP scanning

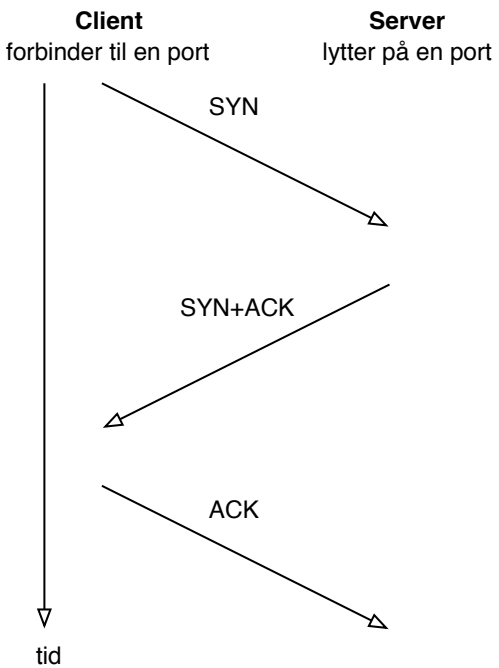
TCP handshake er nemmere at identificere

UDP applikationer svarer forskelligt - hvis overhovedet

TCP three way handshake

.

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



Ping og port sweep

scanninger på tværs af netværk kaldes for sweeps

Scan et netværk efter aktive systemer med PING

Scan et netværk efter systemer med en bestemt port åben

Er som regel nemt at opdage:

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

nmap port sweep efter port 80/TCP

Port 80 TCP er webservere

```
 #  nmap  -p 80 217.157.20.130/28

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

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

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

nmap port sweep efter port 161/UDP

Port 161 UDP er SNMP

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

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

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

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

The 1 scanned port on  (217.157.20.132) is: closed
```

OS detection

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

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

Top Security Tools

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

Den er samlet af Fyodor og findes på:
`https://www.sectools.org/`

Hvad skal der ske?

Tænk som en hacker

Rekognoscering

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

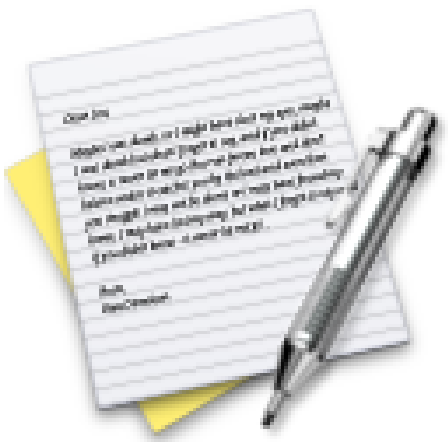
Udnyttelse/afprøvning: Nessus, nikto, exploit programs

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

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

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

Exercise

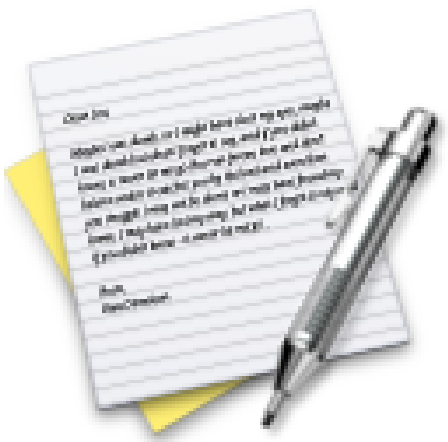


Now lets do the exercise

Nping check ports 5 min

which is number **11** in the exercise PDF.

Exercise

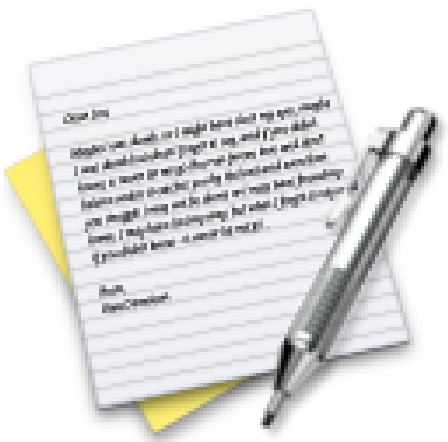


Now lets do the exercise

Try pcap-diff 10 min

which is number **12** in the exercise PDF.

Exercise

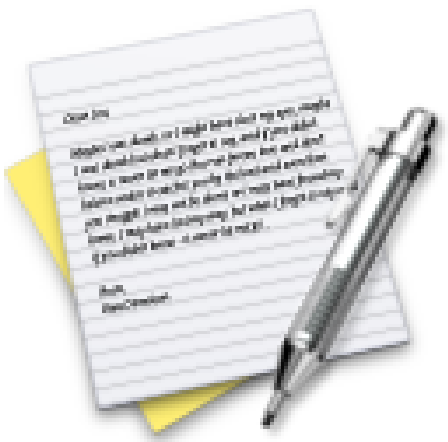


Now lets do the exercise

Discover active systems ping sweep 5 min

which is number **13** in the exercise PDF.

Exercise

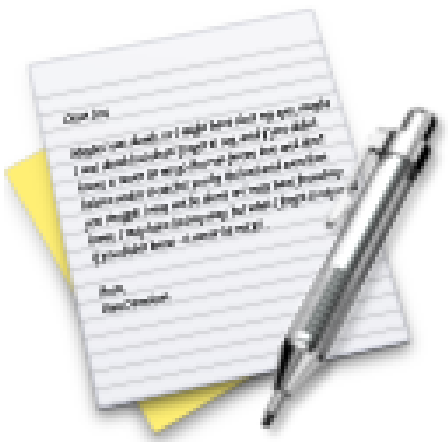


Now lets do the exercise

Execute nmap TCP and UDP port scan 10 min

which is number **14** in the exercise PDF.

Exercise

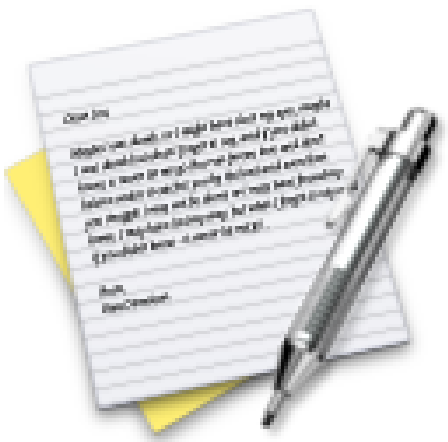


Now lets do the exercise

Perform nmap OS detection 10 min

which is number **15** in the exercise PDF.

Exercise

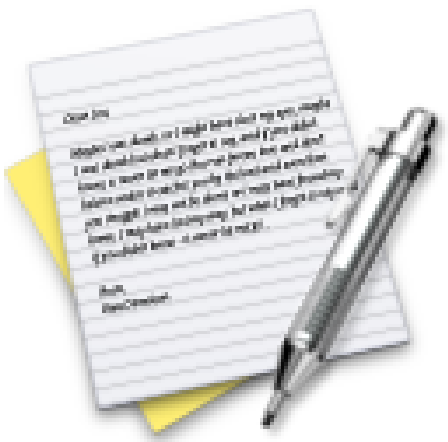


Now lets do the exercise

Perform nmap service scan 10 min

which is number **16** in the exercise PDF.

Exercise

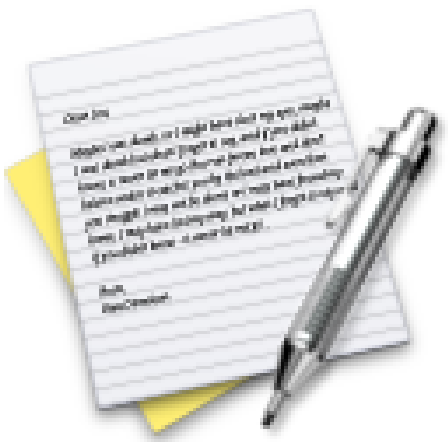


Now lets do the exercise

Nmap full scan 10min

which is number **17** in the exercise PDF.

Exercise



Now lets do the exercise

Reporting HTML 10 min

which is number **18** in the exercise PDF.

Firewalls og IPv6

Læg mærke til forskellen mellem ARP og ICMPv6

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

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

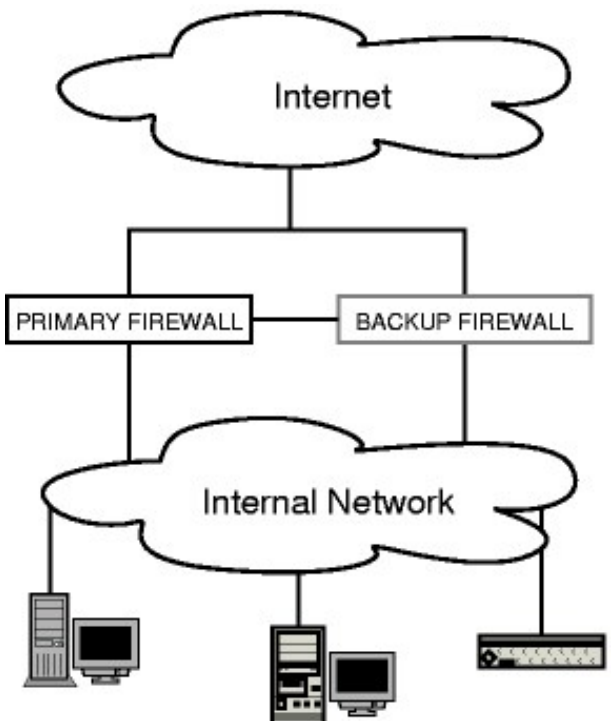
OpenBSD PF IPv6 NDP

```
# Macros: define common values, so they can be referenced and changed easily.
int_if=vr0
ext_if=vr2
tunnel_if=gif0
table <homenet6>  2001:16d8:ffd2:cf0f::/64
set skip on lo0
scrub in all
# Filtering: the implicit first two rules are
block in all

# allow ICMPv6 for NDP
# server with configured IP address and router advertisement daemon running
pass in inet6 proto ipv6-icmp all icmp6-type neighbradv keep state
pass out inet6 proto ipv6-icmp all icmp6-type routersol keep state

# client which uses autoconfiguration would use this instead
#pass in inet6 proto ipv6-icmp all icmp6-type routeradv keep state
#pass out inet6 proto ipv6-icmp all icmp6-type neighbrsol keep state

...  probably not working AS IS
```

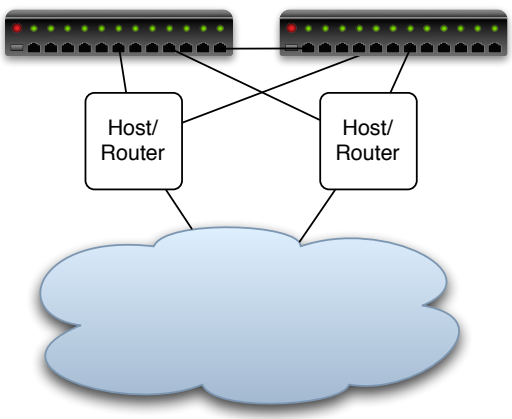
Redundante firewalls

.

- Mange producenter giver mulighed for redundante firewalls/routere
- Eksempler VRRP, CARP, HSRP Cisco, VARP Arista
- 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



Redundante forbindelser IP-niveau



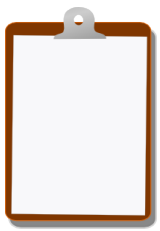
HSRP Hot Standby Router Protocol, Cisco protokol, RFC-2281

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

CARP Common Address Redundancy Protocol, findes på OpenBSD og FreeBSD

`http://en.wikipedia.org/wiki/Common_Address_Redundancy_Protocol`

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books
Most days have about 100 pages or less, but one day has 4 chapters to read!

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools