



Welcome to

# Firewall Overview: The Modern Firewall Infrastructure PROSA

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

Slides are available as PDF, kramse@Github   
firewall-overview.tex in the repo security-courses

# Contact information



- Henrik Kramselund, he/him internet samurai mostly networks and infosec
- Network and security consultant Zencurity, teach at KEA and activist
- Master from the Computer Science Department at the University of Copenhagen (DIKU)
- Email: [hlk@zencurity.com](mailto:hlk@zencurity.com)      Mobile: +45 2026 6000
- Run a small network for fun AS57860

You are welcome to drop me an email

# Goals for today



- Introduce a network tool for isolation, firewalls and discuss what they *filter on* – network packets
- See how they relate to VLANs – logical/virtual separation for hosts  
Wi-Fi is also related to VLAN, but not much about wireless today
- The Modern Firewall Infrastructure: Pruning networks without breaking them
- Building a firewall infrastructure, talk about parts, show and tell

hfill Photo by Thomas Galler on Unsplash

# Plan for today: Firewall Overview



## Subjects

### A) Introduction and firewall basics

- Traffic inspection and firewalls
- Generic IP Firewalls stateless filtering vs stateful inspection
- Common features in firewalls

### B) The Modern Firewall Infrastructure

- Problems in firewalls, and network security
- Firewall pruning – handling complexity
- Making changes, administration

### C) Examples of firewalls, more hands-on show it

## Time schedule



- 17:00 - 18:15 Introduction and basics
- 30min break Eat and mingle, hang around, get coffee/tea  
Go be with your family
- 18:45 - 19:30 Walking through the technologies and protection
- 15min break
- 19:45 -20:30 More about components used for building secure and robust networks
- 20:30 - 21:00 playtime, questions, demo, discussion

# Introducing firewalls



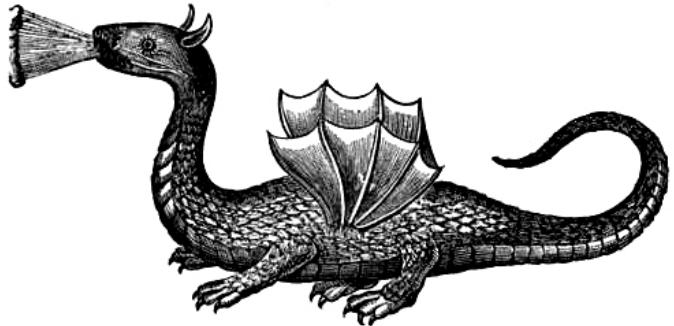
Some of these slides are part of the course:

*Communication and Network Security* at KEA

specifically the slideshow

*3. Traffic Inspection and Firewalls*

# Networks are trouble



Internet here be dragons

- Networks are constantly evolving
- Increased threat landscape, World wide networks, attacks from everywhere
- Vulnerabilities are found daily, Software quality - even security and firewall software has flaws
- Even more vulnerabilities are *developed* and *installed*  
Sorry developers, but some of you don't care, and it shows!

# Best Current Practice



Lets get this out of the way immediately, you should already be doing

- Network segmentation and filtering – we could write a book about this! 🏴
- Monitor your network – both bandwidth, error, netflow etc. 🏴
- Take control of your network, no more admin/admin logins on core devices 🏴
- Turn on authentication for protocols – routing protocols but also any http service within your org 🏴
- Configure host-based firewalls 🏴
- Control DNS – internally and externally, recursive, authoritative etc. 🏴

This goes for IPv4-only, IPv6-only, and mixed networks!

# Internet in a Box



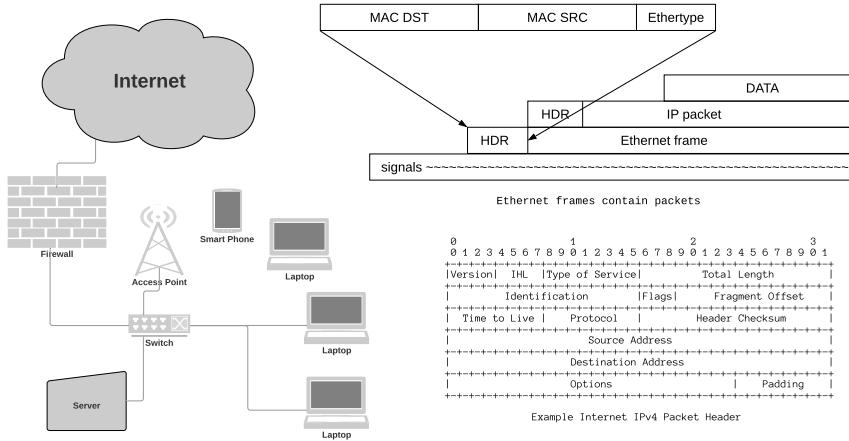
The main purposes for bringing this box, is to show

- These are standard devices, Juniper EX3300 cheap oldish, works great
- Managed switches are a must! You can learn by buying cheap ones, like the TP-Link T1500G-10PS shown, VLAN, SNMP, Syslog ...
- Multiple systems created using PC Engines APU2C4 (really D4) running OpenBSD, Unbound, Suricata, Zeek, DHCP, router advertisement, PF firewall - explicit and nice ICMPv6 filtering ...
- Attack systems compact PCs or laptop
- Creating a home lab is not expensive, bought the Arista 7150 24-port 10G used on ebay



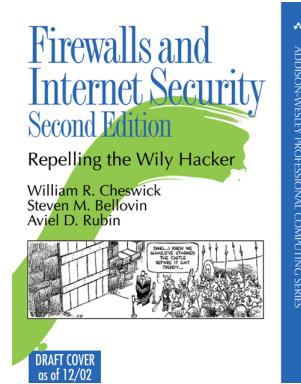
You should have similar (or better) devices in your production network, and they can be configured to do a LOT more than you use them for right now

# Protection, building secure and robust networks



- We should prefer security mechanisms that does NOT require us to keep patching every month
- Can we change our networks to avoid this? Yes!

# Back in the day: Firewalls and Internet Security, 1994



- *Firewalls and Internet Security: Repelling the Wily Hacker*, Second Edition 2003, William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin, **2003** <http://www.wilyhacker.com/>
- The full PDF—and the full LaTeX source of the book. Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
- How to configure firewalls often boil down to, should we allow protocol X
- If we allow certain protocols through a firewall, we are asking for trouble



## Firewalls defined, multiple definitions

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

Source: Wikipedia [https://en.wikipedia.org/wiki/Firewall\\_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))

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

Source: Cheswick 1994 book and 2nd ed 2003 *Firewalls and Internet Security: Repelling the Wily Hacker* , Second Edition, William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin, **2003**

# Network Segmentation – Firewalls



\$ firewall

1. (I) **An internetwork gateway that restricts data communication traffic to and from one of the connected networks** (the one said to be "inside" the firewall) and thus protects that network's system resources against threats from the other network (the one that is said to be "outside" the firewall). (See: guard, security gateway.)

2. (O) **A device or system that controls the flow of traffic between networks using differing security postures.**  
Wack, J. et al (NIST), "Guidelines on Firewalls and Firewall Policy", Special Publication 800-41, January 2002.

Tutorial: A firewall typically protects a smaller, secure network (such as a corporate LAN, or even just one host) from a larger network (such as the Internet). The firewall is installed at the point where the networks connect, and the firewall applies policy rules to control traffic that flows in and out of the protected network.

Source: RFC4949 *Internet Security Glossary, Version 2*

<https://datatracker.ietf.org/doc/html/rfc4949> 2007

## Continued



**A firewall is not always a single computer.** For example, a firewall may consist of a pair of filtering routers and one or more proxy servers running on one or more bastion hosts, all connected to a small, dedicated LAN (see: buffer zone) between the two routers. The external router blocks attacks that use IP to break security (IP address spoofing, source routing, packet fragments), while proxy servers block attacks that would exploit a vulnerability in a higher-layer protocol or service. The internal router blocks traffic from leaving the protected network except through the proxy servers. The difficult part is defining criteria by which packets are denied passage through the firewall, because a firewall not only needs to keep unauthorized traffic (i.e., intruders) out, but usually also needs to let authorized traffic pass both in and out.

Source: RFC4949 *Internet Security Glossary, Version 2*

<https://datatracker.ietf.org/doc/html/rfc4949> 2007

- Network layer, we often call them packet filters, stateless
- application level, we often call them, stateful filtering and gateways
- Firewalls are by design a choke point, natural place to do network security monitoring!

## Definition of firewalls – Wikipedia



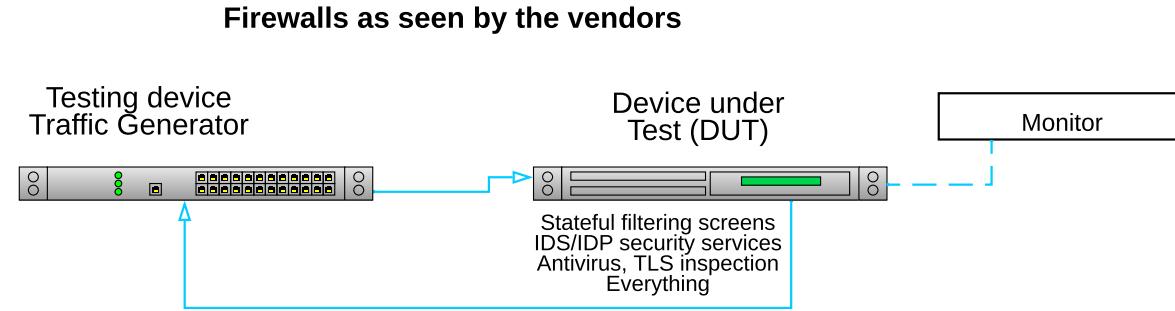
Another short definition that encapsulates this is found on Wikipedia, and may suffice in many situations. Again there will typically be multiple networks, zones or areas of the networks with varying degrees of trust.

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

Source: Wikipedia [https://en.wikipedia.org/wiki/Firewall\\_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))

**TL;DR Not necessarily a single device**

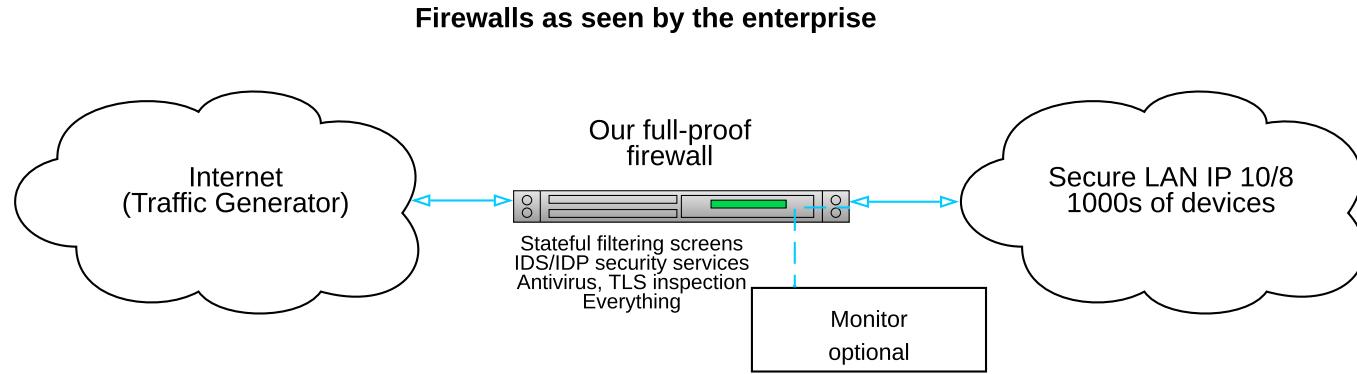
# A firewall – in the vendor eyes



**"Can your firewall flex in the face of change?  
Does it harmonize your network, workload, and application security? Does it protect apps and employees in your hybrid or multicloud environment? Make sure you're covered."**

Source: not shown to protect the audience from further marketing speak

# A firewall – in the enterprise mindset



- Even though some vendors suggest they can do everything in a single box, I don't believe them!
- Truth – yes, we can do almost anything in software
- Realization **Your infrastructure is based on multiple components and or devices**

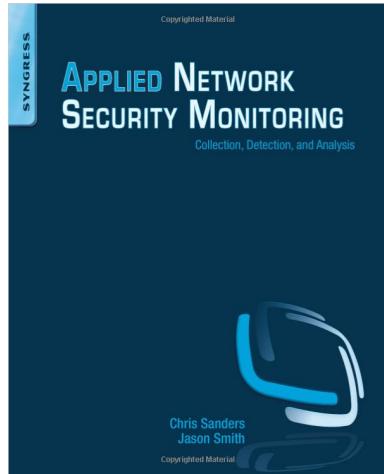
# Baseline Skills for Network Security



- Threat-Centric Security, NSM, and the NSM Cycle
- TCP/IP Protocols
- Common Application Layer Protocols
- Packet Analysis
- Windows Architecture
- Linux Architecture
- Basic Data Parsing (BASH, Grep, SED, AWK, Python, PowerShell etc)
- IDS Usage (Snort, Suricata, etc.)
- Indicators of Compromise and IDS Signature Tuning
- Open Source Intelligence Gathering
- Basic Analytic Diagnostic Methods
- Basic Malware Analysis

Source: *Applied Network Security Monitoring Collection, Detection, and Analysis*,  
Chris Sanders and Jason Smith

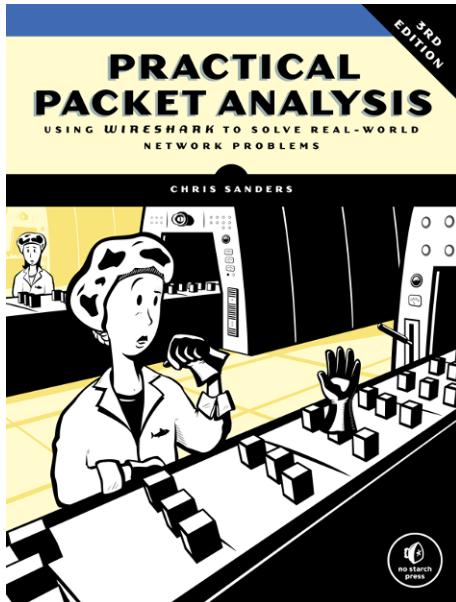
# Book: Applied Network Security Monitoring (ANSM)



*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

# 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>

# Network Protocol Knowledge Needed for Network Security

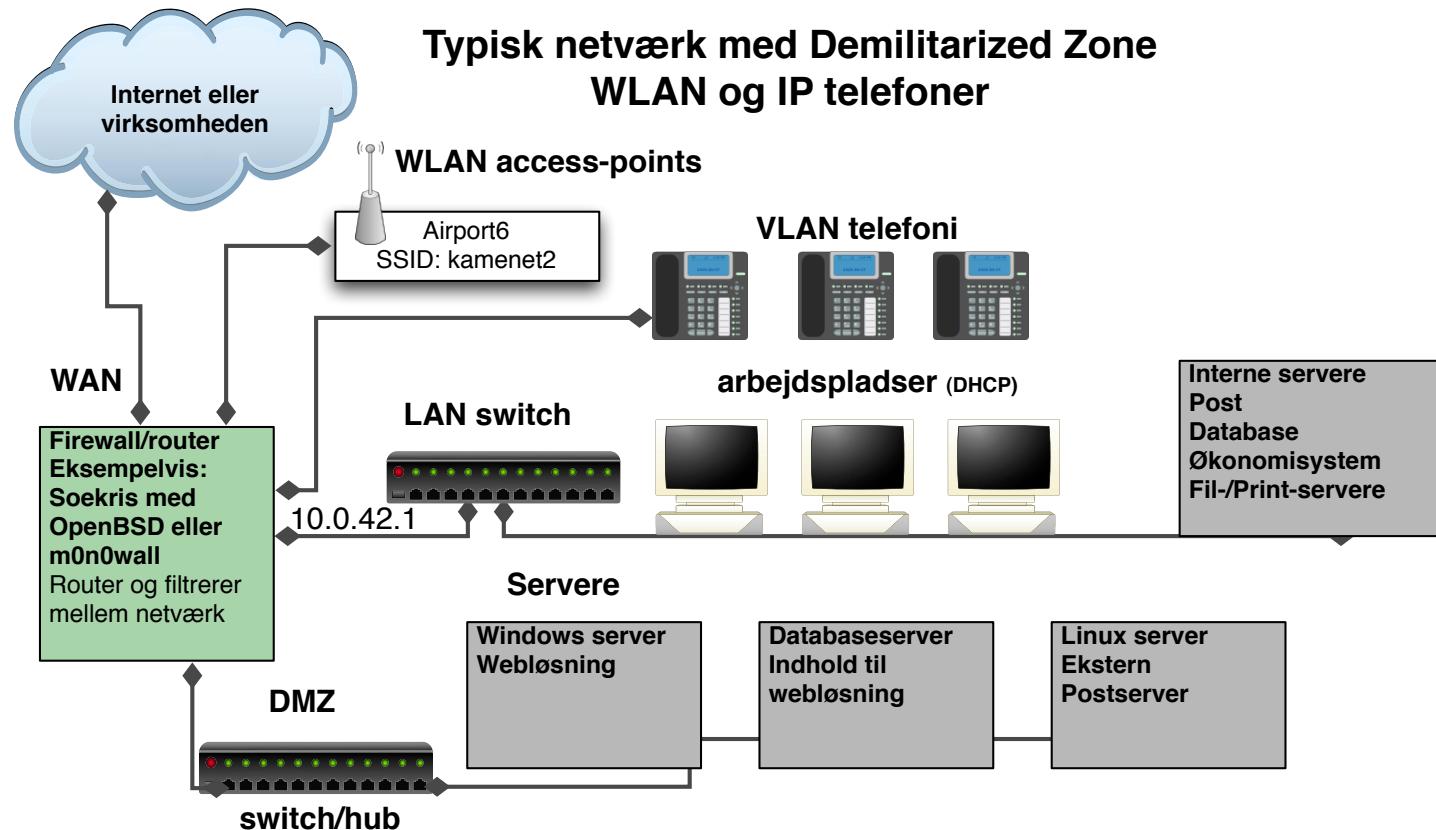


To work with network security the following protocols are the bare minimum to know about.

- ARP Address Resolution Protocol for IPv4
- NDP Neighbor Discovery Protocol for IPv6
- IPv4 & IPv6 – the basic packet fields source, destination,
- ICMPv4 & ICMPv6 Internet Control Message Protocol
- UDP User Datagram Protocol
- TCP Transmission Control Protocol
- DHCP Dynamic Host Configuration Protocol
- DNS Domain Name System

These protocols are part of the Internet Protocol suite, or TCP/IP for short. The canonical document describing this is from 1981 RFC-0791 *Internet Protocol*. The protocols were deployed on the internet around 1983.

# Unified communications



# Modern Firewall Infrastructures



A firewall **blocks** traffic on a network

A firewall **allows** traffic on a network

The interesting part is typically what it allows!

A firewall infrastructure must:

- Prevent attackers from entering
- Prevent data exfiltration
- Prevent worms, malware, virus from spreading in networks
- Be part of an overall solution with ISP, routers, other firewalls, switched infrastructures, intrusion detection systems and the rest of the infrastructure

Difficult – and requires design and secure operations

# Packet Filtering



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

Packet filtering are firewall devices filtering on single packet

Most *specialized firewall* devices do stateful filtering and more

Don't forget IPv6 – even though you haven't turned it on, it is there



Basalt set et netværksfilter - det yderste fæstningsværk

Indholder typisk:

- Grafisk brugergrænseflade til konfiguration - er det en fordel?
- TCP/IP filtermuligheder - pakernes afsender, modtager, retning ind/ud, porte, protokol, ...
- 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 "

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

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

pass out
```



## Example Firewall Products

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

Those listed are the most popular commercial ones I see in Denmark

# Open source based firewalls



- Linux firewalls IP tables, use command line tool ufw Uncomplicated Firewall!
- Firewall GUIs on top of Linux – lots! Some are also available as commercial ones
- OpenBSD PF <http://www.openbsd.org>
- FreeBSD IPFW og IPFW2 <http://www.freebsd.org>
- Mac OS X uses OpenBSD PF
- FreeBSD has an older version of the OpenBSD PF, should really be renamed now

# OPNsense GUI based and easy to install



A screenshot of the OPNsense web-based management interface. The main title bar says "Firewall: Rules: LAN". Below it, there's a table with columns: Evaluations, States, Packets, Bytes, and Description. The table lists several rules: "allow access to DHCP server" (196 evaluations, 0 states, 0 packets, 0 bytes), "allow access to DHCP server" (0 evaluations, 0 states, 0 packets, 0 bytes), "allow access to DHCP server" (1408 evaluations, 0 states, 0 packets, 0 bytes), "anti-lock rule" (1582 evaluations, 24 states, 837 packets, 430 KB bytes), and "Default LAN" (171 evaluations, 187 states, 160940 packets, 132.94 MB bytes). The "Description" column contains links to edit each rule. At the bottom of the interface, there are legends for actions (pass, block, reject, log) and directions (in, out), and a note about rule evaluation order.

OPNsense <https://opnsense.org/>

Firewall bygget på FreeBSD med grafisk web interface

Oprindeligt videreudvikling af m0n0wall over <https://www.pfsense.org/>

De har været brugt i produktion i danske firmaer igennem mange år



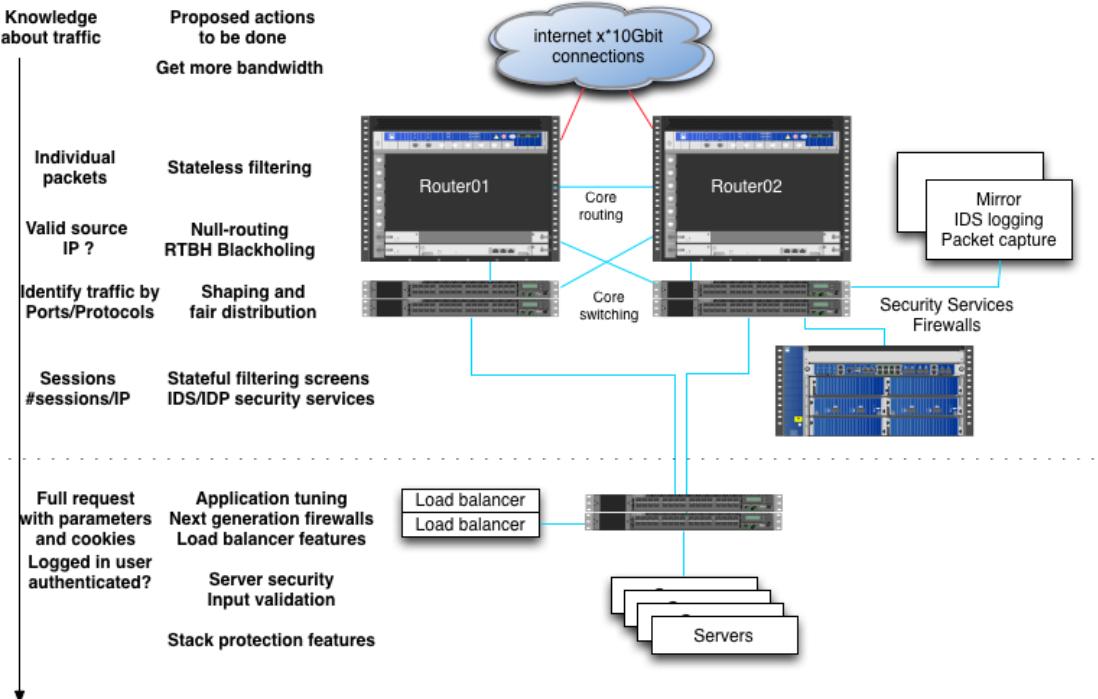
# Uncomplicated Firewall (UFW)

```
root@debian01:~# apt install ufw
...
root@debian01:~# ufw allow 22/tcp
Rules updated
Rules updated (v6)
root@debian01:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@debian01:~# ufw status numbered
Status: active
```

To	Action	From
--	-----	----
[ 1] 22/tcp	ALLOW IN	Anywhere
[ 2] 22/tcp (v6)	ALLOW IN	Anywhere (v6)

- Extremely easy to use – I recommend and use the (Uncomplicated Firewall) UFW

# Specialized Firewall devices are NOT Alone

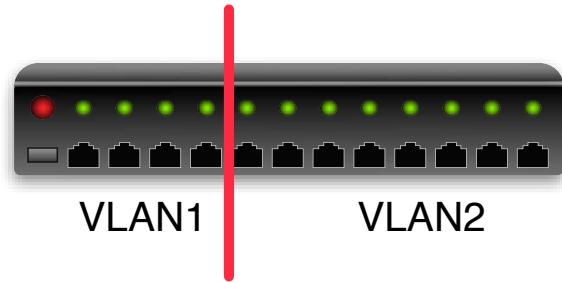


Use Defense in Depth – all layers have features

## Together with Firewalls - Virtual LAN (VLAN)



Portbased VLAN



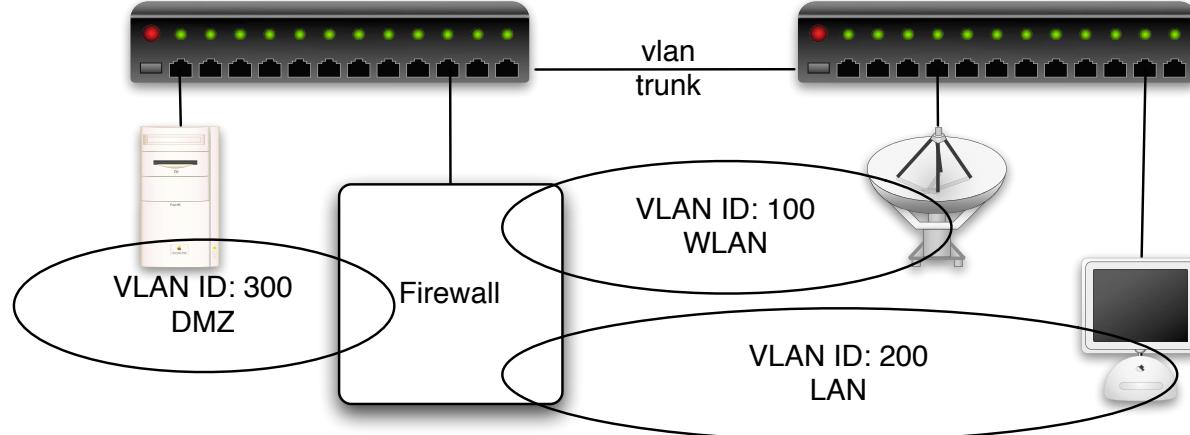
Managed switches often allow splitting into zones called virtual LANs

Most simple version is port based

Like putting ports 1-4 into one LAN and remaining in another LAN

Packets must traverse a router or firewall to cross between VLANs

# Virtual LAN (VLAN) IEEE 802.1q



Using IEEE 802.1q VLAN tagging on Ethernet frames

Virtual LAN, to pass from one to another, must use a router/firewall

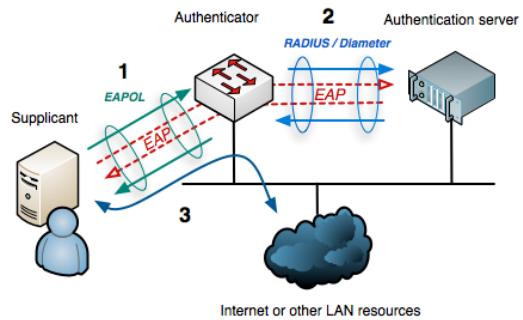
Allows separation/segmentation and protects traffic from many security issues

# Network Access Control – Connecting clients more securely



Talking about standard, another useful one:

IEEE 802.1x – Port Based Network Access Control



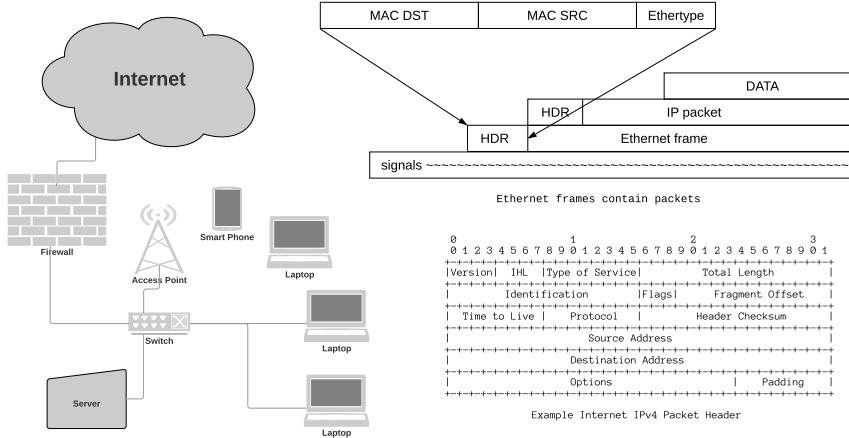
Authentication protocol ensures user validation before port access

Can authenticate using username and then password or certificate

Typically RADIUS and 802.1x which can use LDAP or Active Directory

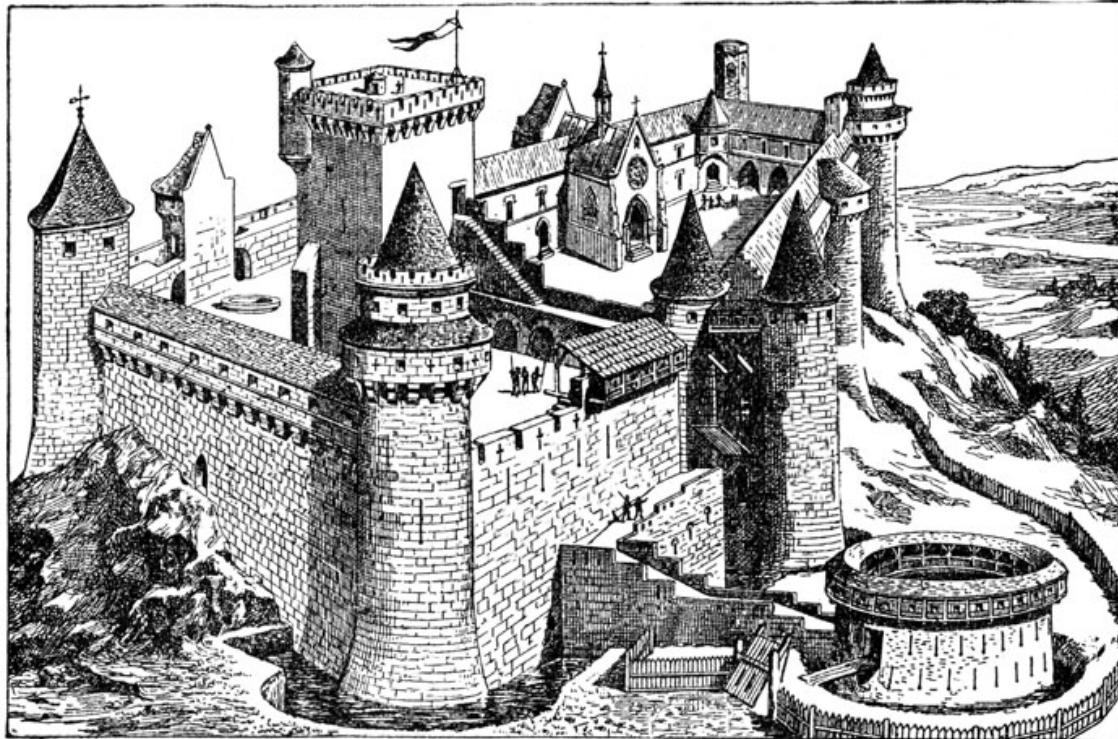
Already used in Wi-Fi networks, so can be turned on for wired Ethernet ports

# Protection, building secure and robust networks



- We should prefer security mechanisms that does NOT require us to keep patching every month
- Can we change our networks to avoid this? Yes!

# Defense in depth



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



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!**

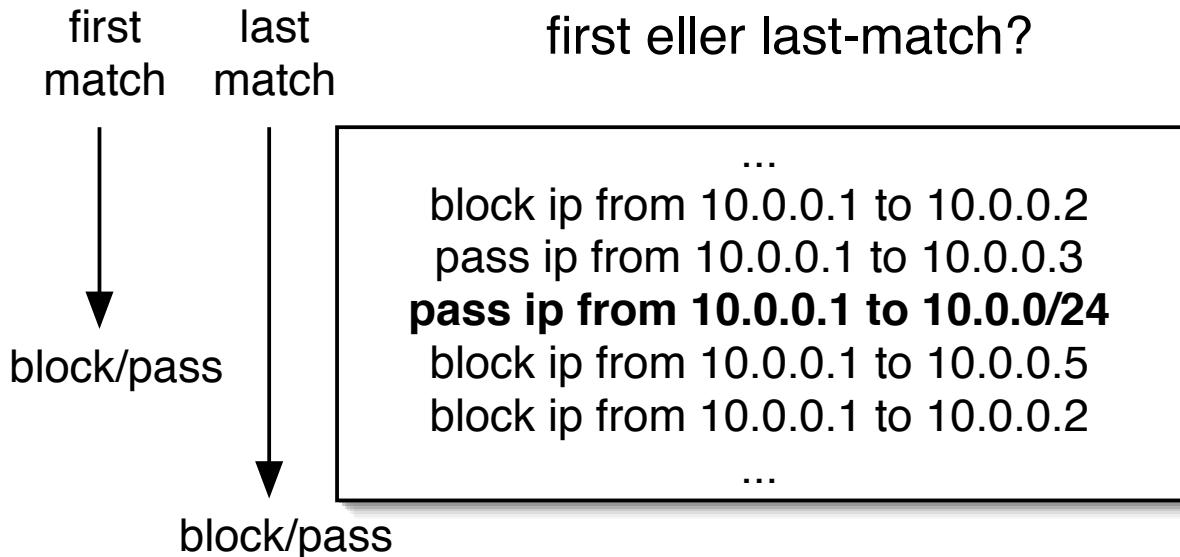
OpenBSD PF er last match

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

## Firewalls og ICMP



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

Ovenstående er IPFW syntaks for at tillade de interessante ICMP beskeder igennem  
Tillad visse ICMP typer – og tilsvarende på IPv6:

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

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



# Firewall konfiguration

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

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



## Blokér indefra og ud

Der er porte og services som altid bør blokeres

Det kan være kendte sårbare services

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

Kendte problemer som minimum

Demo time? Nmap and Nping?



## Special 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
- inspection - diverse muligheder for at lave deep inspection i protokoller
- Eksempelvis DNS inspection

## 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

Reverse proxy benyttes også ofte til at terminere TLS/HTTPS og lave loadbalancing

Web Application Firewall (WAF) benyttes også

NB: nogle protokoller egner sig ikke til proxy servere

## 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

# Netflow and Session Logging



- Netflow is getting more important, more data share the same links
- Accounting is important
- Detecting DoS/DDoS and problems is essential
- Netflow sampling is vital information - 123Mbit, but what kind of traffic
- NFSen is an old but free application <http://nfsen.sourceforge.net/>
- Currently also investigating sFlow - hopefully more fine grained
- sFlow, short for "sampled flow", is an industry standard for packet export at Layer 2 of the OSI model, <https://en.wikipedia.org/wiki/SFlow>

Netflow is often from routers, we dont have any here

Also look into Elastiflow! <https://github.com/robcowart/elastiflow>

# Collect Network Evidence from the network



## Network Flows

Cisco standard NetFlow version 5 defines a flow as a unidirectional sequence of packets that all share the following 7 values:

- Ingress interface (SNMP ifIndex)
- IP protocol, Source IP address and Destination IP address
- Source port for UDP or TCP, 0 for other protocols
- Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
- IP Type of Service

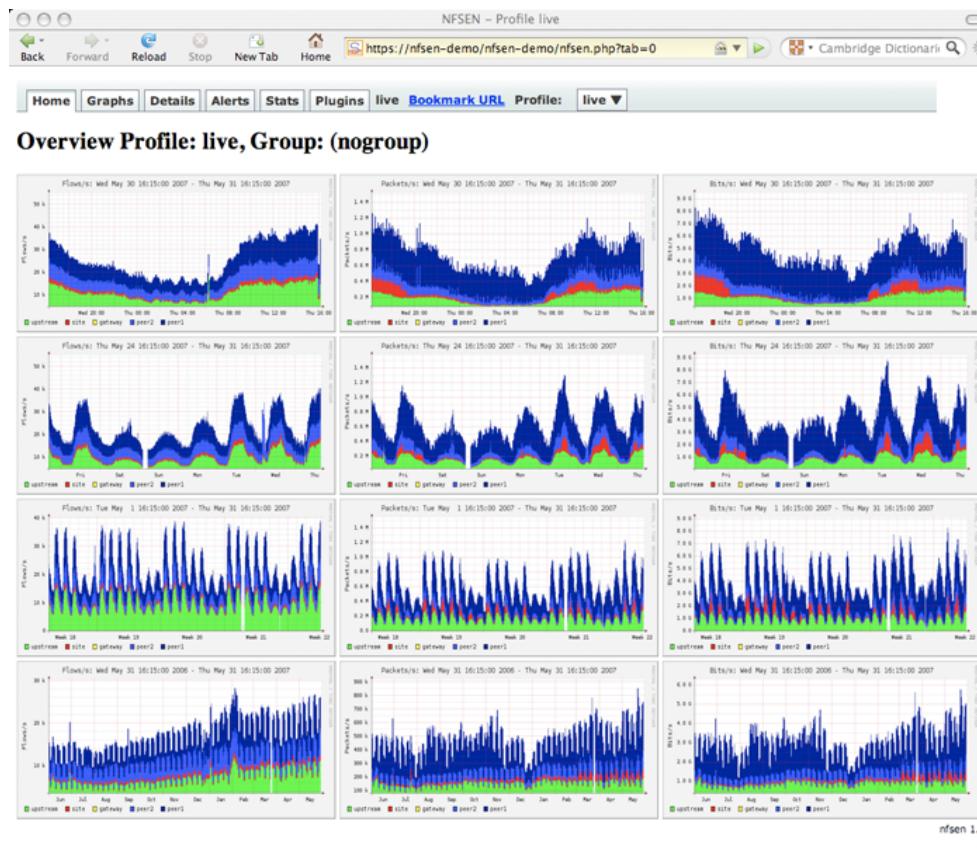
today Netflow version 9 or IPFIX

Source:

<https://en.wikipedia.org/wiki/NetFlow>

[https://en.wikipedia.org/wiki/IP\\_Flow\\_Information\\_Export](https://en.wikipedia.org/wiki/IP_Flow_Information_Export)

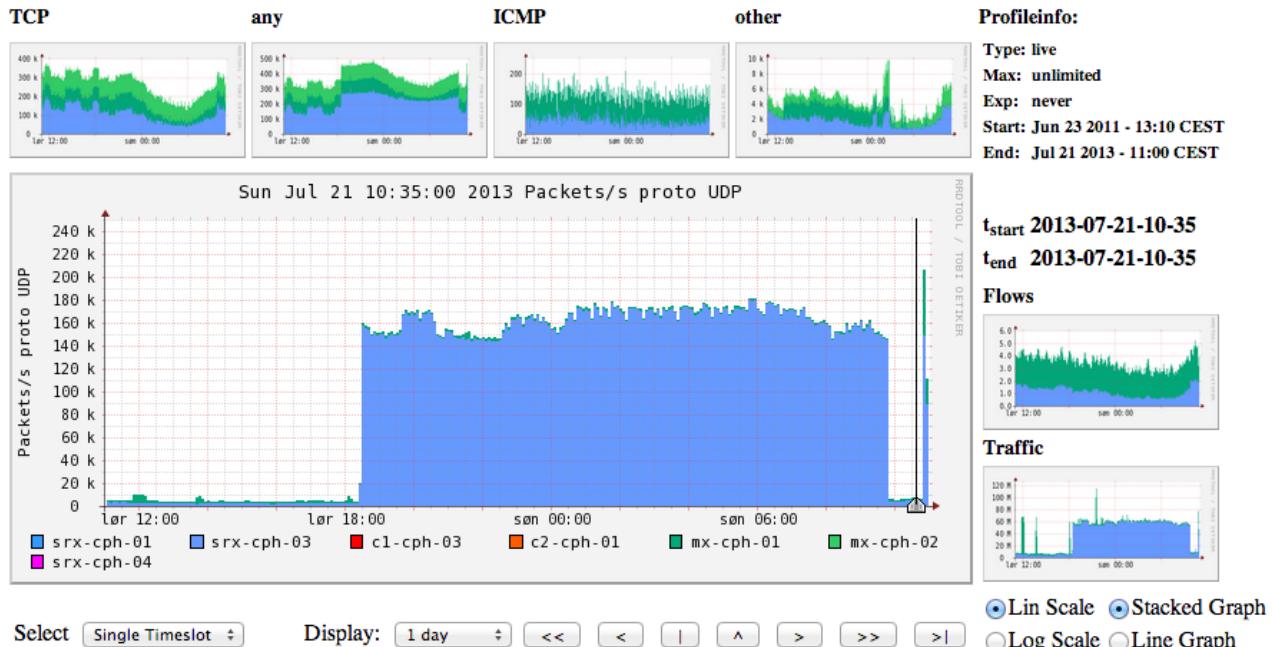
# Netflow using NfSen



# Netflow NFSen



## Profile: live



An extra 100k packets per second from this netflow source (source is a router)

# Netflow processing from the web interface



NFSEN - Profile live May 31 2007 - 04:40

Back Forward Reload Stop New Tab Home https://nfsen-demo/nfsen-demo/nfsen.php?processing

peer2 3.3 k/s 76.2 k/s 66.9 k/s 7.0 k/s 621.0 /s 1.7 k/s 484.6 Mb/s 459.9 Mb/s 12.5 Mb/s 437.3 kb/s 11.7 Mb/s  
gateway 1.0 /s 651.0 /s 600.8 /s 46.6 /s 0 /s 3.7 /s 6.2 Mb/s 6.1 Mb/s 36.4 kb/s 0 b/s 4.4 kb/s  
site 467.1 /s 8.9 k/s 6.1 k/s 2.0 k/s 181.7 /s 613.3 /s 38.8 Mb/s 28.3 Mb/s 7.4 Mb/s 104.0 kb/s 2.9 Mb/s  
upstream 6.4 k/s 94.2 k/s 84.3 k/s 8.2 k/s 896.4 /s 766.7 /s 588.4 Mb/s 568.2 Mb/s 16.7 Mb/s 685.1 kb/s 2.8 Mb/s

All | None Display:  Sum  Rate

**Netflow Processing**

Source: peer1 Filter:

peer1 peer2 gateway site upstream

All Sources and <none>

Options:

List Flows  Stat TopN  
Top: 10  
Stat: Flow Records order by flows  
proto  
srcPort  
dstPort  
srcIP  
dstIP  
Aggregate  
Limit: Packets > 0  
Output: line / IPv6 long

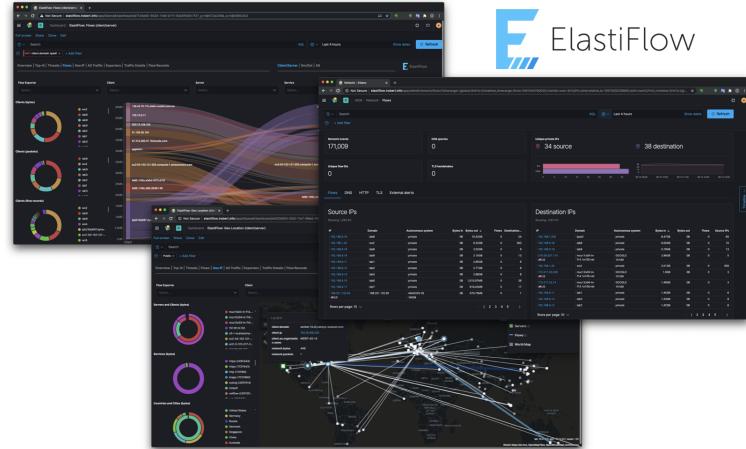
Clear Form process

```
** nfdump -M /netflow0/nfsen-demo/profile-data/live/peer1:peer2:gateway:site:upstream -T -r 2007/05/31/04:nfcapd.200705310440
nfdump filter:
any
Aggregated flows 2797250
Top 10 flows ordered by flows:
Date flow start Duration Proto Src IP Addr:Port Dst IP Addr:Port Packets Bytes Flows
2007-05-31 04:39:54.045 299.034 UDP 116.147.95.88:1110 -> 188.142.64.162:27014 68 5508 68
2007-05-31 04:39:56.282 298.174 UDP 116.147.249.27:1478 -> 188.142.64.163:27014 67 5427 67
2007-05-31 04:39:57.530 298.206 UDP 117.196.44.62:1031 -> 188.142.64.166:27014 67 5427 67
2007-05-31 04:39:57.819 298.112 UDP 117.196.75.134:1146 -> 188.142.64.167:27014 67 5427 67
2007-05-31 04:39:53.187 297.216 UDP 61.191.235.132:4121 -> 60.9.138.37:4121 62 3720 62
2007-05-31 04:39:53.234 303.588 UDP 60.9.138.37:2121 -> 118.25.93.95:2121 61 3660 61
2007-05-31 04:39:58.921 298.977 UDP 60.9.138.36:2121 -> 121.135.4.186:2121 61 3660 61
2007-05-31 04:39:54.329 303.585 UDP 120.150.194.76:2121 -> 60.9.138.37:2121 61 3660 61
2007-05-31 04:39:53.916 300.734 UDP 60.9.138.37:2121 -> 125.167.25.128:2121 61 3660 61
2007-05-31 04:39:57.946 300.353 UDP 60.9.138.36:2121 -> 121.135.4.186:2121 61 3660 61

IP addresses anonymized
Summary: total flows: 4616424, total bytes: 156.6 G, total packets: 172.6 M, avg bps: 644.8 M, avg pps: 90946, avg bpp: 929
Time: 2007-05-31 04:11:45 - 2007-05-31 04:44:55
Total flows processed: 4616424, skipped: 0, Bytes read: 240064932
Sys: 6.184s flows/second: 746464.4 Wall: 6.185s flows/second: 746361.3
```

Bringing the power of the command line forward

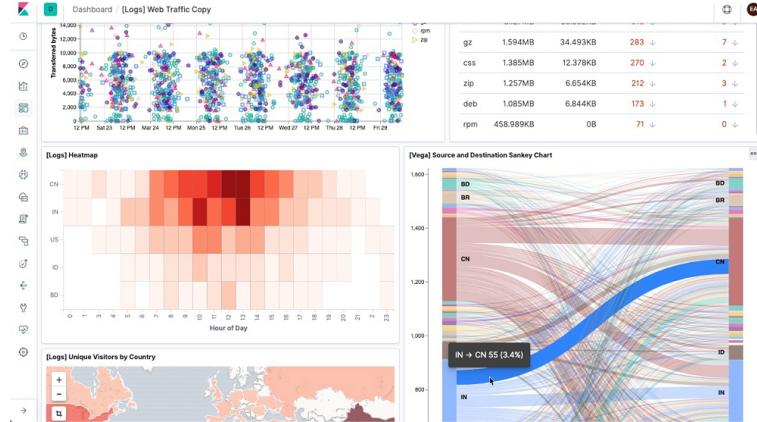
## ElastiFlow – Elasticsearch based



ElastiFlow™ provides network flow data collection and visualization using the Elastic Stack (Elasticsearch, Logstash and Kibana). It supports Netflow v5/v9, sFlow and IPFIX flow types (1.x versions support only Netflow v5/v9).

Source: Picture and text from <https://github.com/robcowart/elastiflow>

# Big Data tools: Elasticsearch



Elasticsearch is an open source distributed, RESTful search and analytics engine capable of solving a growing number of use cases.

<https://www.elastic.co>

We are all Devops now, even security people!

# Kibana

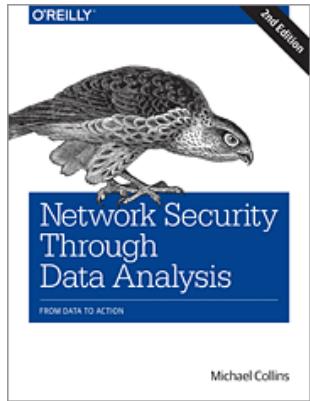


Highly recommended for a lot of data visualisation as non-programmers can create, save, and share dashboards

Source: <https://www.elastic.co/products/kibana>

I have a set of Ansible files that can install a mostly complete Elastic stack on Debian:  
<https://github.com/kramse/kramse-labs/tree/master/suricatazeek>

# Network Security Through Data Analysis



Low page count, but high value! Recommended.

Network Security through Data Analysis, 2nd Edition By Michael S Collins Publisher: O'Reilly Media 2015-05-01:  
Second release, 348 Pages

New Release Date: August 2017

# Installing Cilium



What about cloud platforms and *firewalls*

I use Cilium

```
helm repo add cilium https://helm.cilium.io/  
  
API_SERVER_IP=<your_api_server_ip>  
# Kubeadm default is 6443  
API_SERVER_PORT=<your_api_server_port>  
helm install cilium cilium/cilium --version 1.12.5 \  
  --namespace kube-system \  
  --set kubeProxyReplacement=strict \  
  --set k8sServiceHost=${API_SERVER_IP} \  
  --set k8sServicePort=${API_SERVER_PORT}
```

- Note: I ended up deciding to run without the kube-proxy, only Cilium
- Document!



# Document what you did, what it did

```
cilium install --version=1.13.0-rc4 \
    --helm-set ipam.mode=kubernetes --helm-set tunnel=disabled \
    --helm-set ipv4NativeRoutingCIDR="10.0.0.0/8" --helm-set bgpControlPlane.enabled=true \
    --helm-set k8s.requireIPv4PodCIDR=true --helm-set kube-proxy-replacement=strict

Using Cilium version 1.13.0-rc4
Auto-detected cluster name: kubernetes
Auto-detected datapath mode: tunnel
Auto-detected kube-proxy has not been installed
Cilium will fully replace all functionalities of kube-proxy
helm template --namespace kube-system cilium cilium/cilium --version 1.13.0-rc4 --set bgpControlPlane.enabled=true,
cluster.id=0,cluster.name=kubernetes,encryption.nodeEncryption=false,ipam.mode=kubernetes,
ipv4NativeRoutingCIDR=10.0.0.0/8,k8s.requireIPv4PodCIDR=true,k8sServiceHost=10.137.0.26,
k8sServicePort=6443,kube-proxyreplacement=strict,kubeProxyReplacement=strict,operator.replicas=1,
serviceAccounts.cilium.name=cilium,serviceAccounts.operator.name=cilium-operator,tunnel=disabled
...
Waiting for Cilium to be installed and ready...
Cilium was successfully installed! Run 'cilium status' to view installation health
```

How I ran the kubeadm for building cluster - note the skip:

```
sudo kubeadm init --pod-network-cidr=10.50.0.0/16 --skip-phases=addon/kube-proxy
```



## After install Cilium

```
root@k8s-1:~# kubectl get po -n kube-system
NAME                  READY   STATUS    RESTARTS   AGE
cilium-2287c          1/1     Running   1 (161m ago)   3d
cilium-9kjhv          1/1     Running   1 (160m ago)   3d
cilium-operator-5589744cf4-7mwqx 1/1     Running   1 (160m ago)   3d
coredns-f74b98ccc-4hg4n      1/1     Running   1 (161m ago)   3d
coredns-f74b98ccc-ts55j      1/1     Running   1 (160m ago)   3d
etcd-k8s-1             1/1     Running   4 (161m ago)   5d
kube-apiserver-k8s-1       1/1     Running   4 (161m ago)   5d
kube-controller-manager-k8s-1 1/1     Running   4 (161m ago)   5d
kube-scheduler-k8s-1        1/1     Running   4 (161m ago)   5d
```

# Cilium CLI tool



```
[Projects] Terminal - hlk@k8s-1: ~
File Edit View Terminal Tabs Help
root@k8s-1:~# cilium status
      /--\      Cilium:      OK
     /_ \_/_\_\  Operator:    OK
    \_/\_/\_/\_/ Hubble:    disabled
   /_/\_/\_/\_/\_ ClusterMesh: disabled

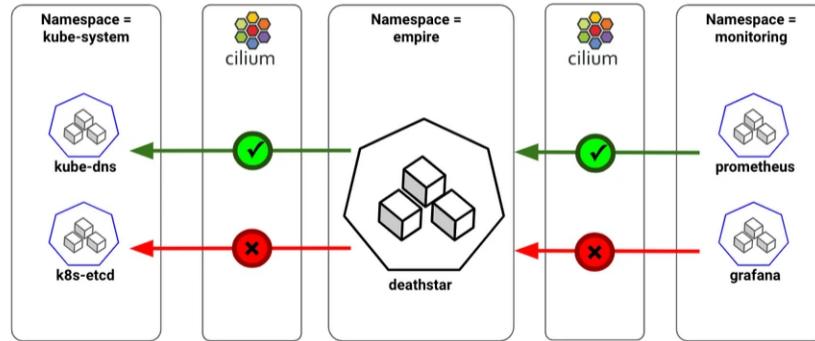
Deployment      cilium-operator  Desired: 1, Ready: 1/1, Available: 1/1
DaemonSet       cilium          Desired: 2, Ready: 2/2, Available: 2/2
Containers:     cilium          Running: 2
                cilium-operator  Running: 1
Cluster Pods:  8/8 managed by Cilium
Image versions  cilium          quay.io/cilium/cilium:v1.13.0-rc4@sha256:32acd47fd9bea9c0045222ba5d27f5fe9ad06dabd572a80b870b1f0e68c0e928: 2
                cilium-operator  quay.io/cilium/operator-generic:v1.13.0-rc4@sha256:19f612d4f1052e26edf33e26f60d64d8fb6caed9f03692b85b429a4ef5d175b2: 1
root@k8s-1:~#
```

- Many tools are executed via kubectl
- Others have their own command
- This can be very confusing, and again – document which tools you use!
- Having a jump host with updated tools installed might help – helps me!

# Cilium overview



## Controlling Ingress/Egress from Namespaces



Kubernetes provides Network Policies for controlling traffic going in and out of the pods. Cilium implements the Kubernetes Network Policies for L3/L4 level and extends with L7 policies for granular API-level security for common protocols such as HTTP, Kafka, gRPC, etc

Source: picture and text from <https://cilium.io/blog/2018/09/19/kubernetes-network-policies/>

# Security is more than blocking!



## Networking

Service Load Balancing

Scalable Kubernetes CNI

Multi-cluster Connectivity

## Observability

Identity-aware Visibility

Advanced Self Service Observability

Network Metrics + Policy Troubleshooting

## Security

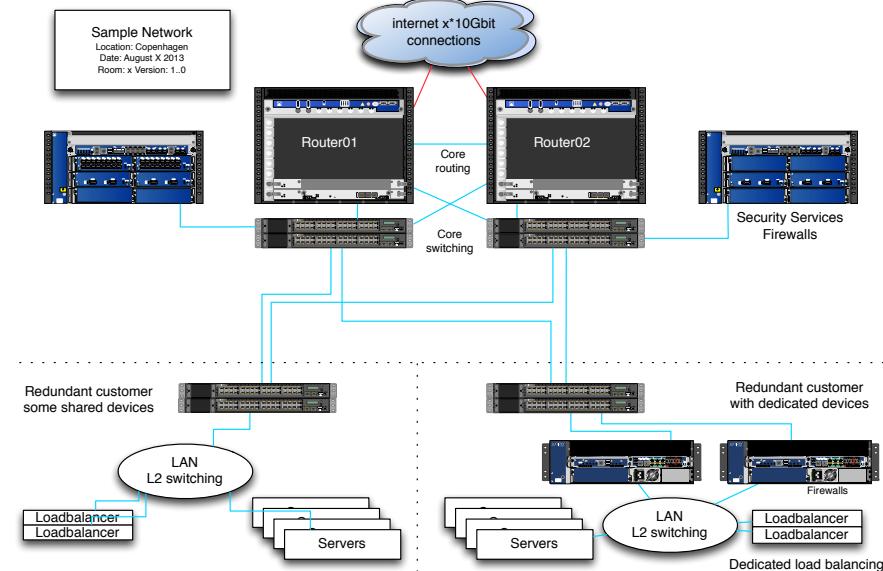
Transparent Encryption

Security Forensics + Audit

Advanced Network Policy

- A lot of features relate to *security*

# Bottlenecks exist, but where



- Lower layer attacks Transport Layer Attacks TCP SYN flood – packet based
- Higher layer attacks like Slowloris and web attacks – keep sessions running
- Protect everything without loosing functionality or creating administrative nightmare

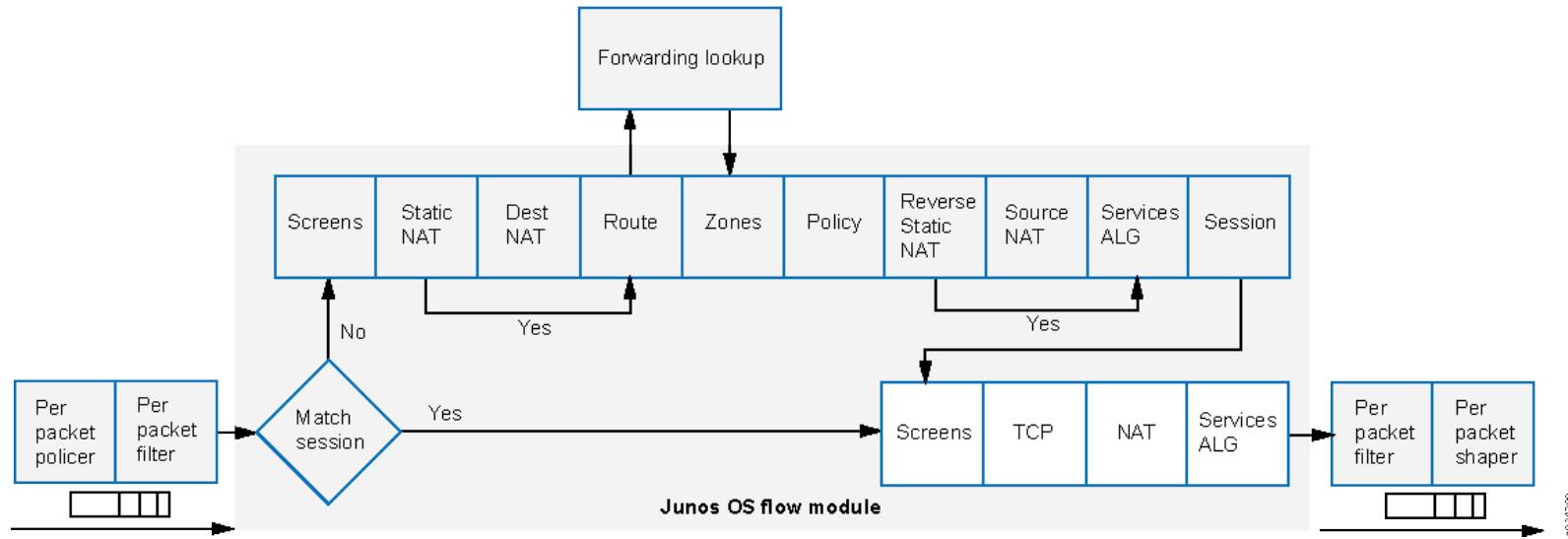
# Availability and Network flooding attacks



The attacks we are discussing today are:

- **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 popular targets – more similar ones exist
- Special packets and protocols – anything that can create *load on systems* work
- 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 (pps)
- **CPU processing** in firewalls, pps
- CPU processing in server software
- **Bandwidth** - megabits per second (mbps)
- Typically source is spoofed or amplification attacks abusing devices on the Internet

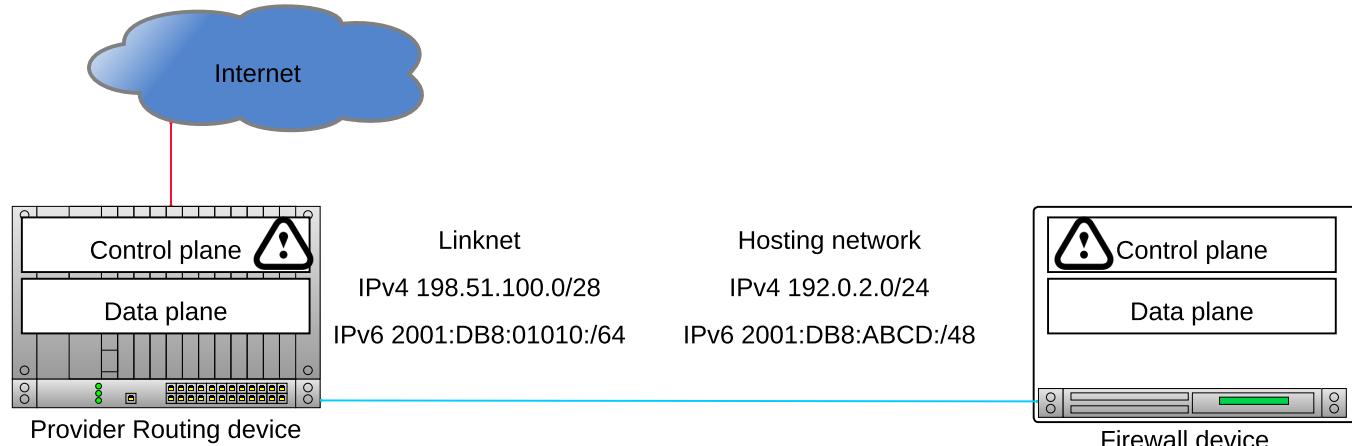
# Packet processing in firewalls – detailed view



## Traffic Processing on SRX Series Devices Overview

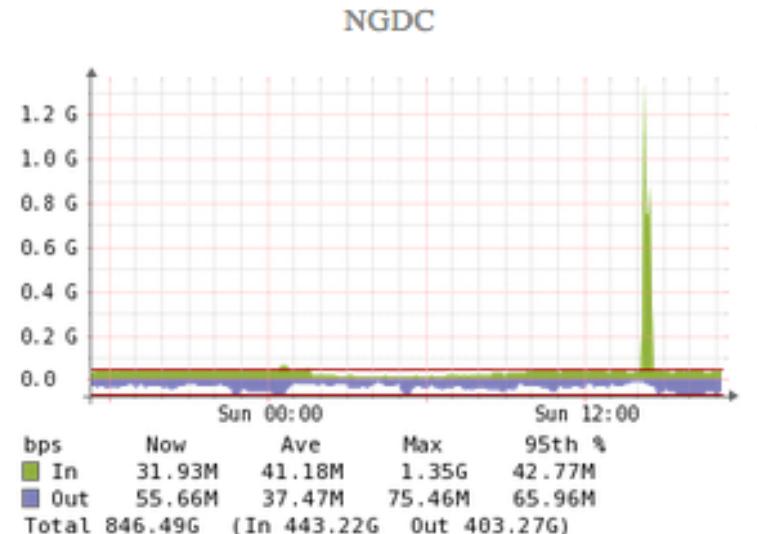
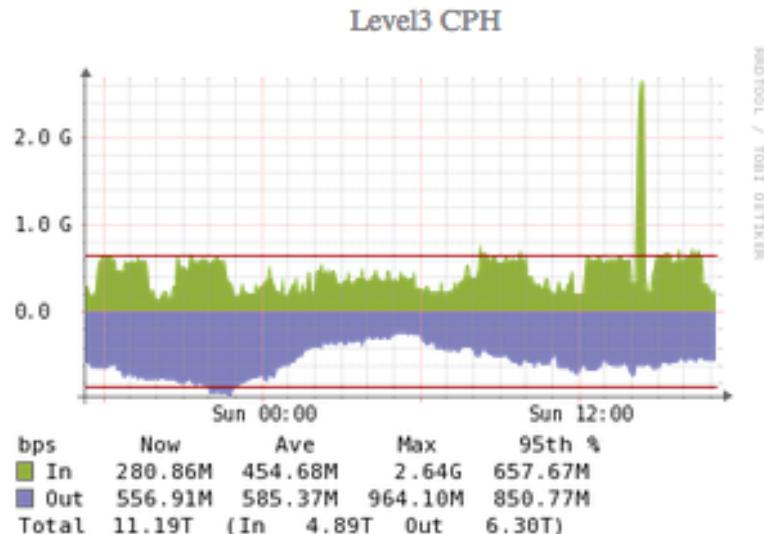
<https://www.juniper.net/documentation/us/en/software/junos/flow-packet-processing/topics/topic-map/security-srx-devices-processing-overview.html>

# Scanning and Attacking – Pressure Points and Scope



- In scope for me is everything that could adversely affect the network
- Common scope IPv4: Link network /28 or /26 and a hosting network /24
- Common scope IPv6: Link network /64 (bad) or /127 (RFC9099) and a hosting network /48 with subnets

# DDoS traffic before filtering



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

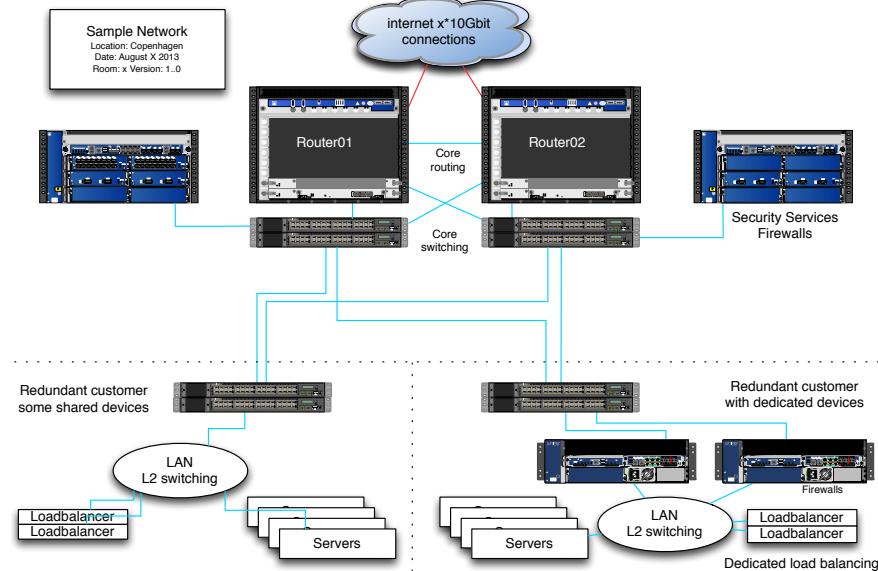
# DDoS traffic after filtering



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

Better to filter stateless before traffic reaches firewall, less work!

# DDoS protection and flooding



- Transport Layer Attacks TCP SYN flood TCP sequence numbers
- High level attacks like Slowloris - keep TCP/HTTP connection for a long time.

# Båndbreddestyring og policy based routing



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

Specielt relevant for DDoS beskyttelse

Findes på F5 BigIP, Cisco, Junos osv.

Mest kendte er i Open Source:

- OpenBSD - integreret i PF
- FreeBSD har dumynet
- Linux Traffic Control

Det kaldes også traffic shaping



## Designing the protection – bandwidth and rate limit

Protocol	Mbps	Prefix
TCP	Up to full bandwidth 10Gbps	192.0.2.0/25
UDP	Less than 1Gbps	192.0.2.128/25
ICMP	Less than 10Mbps	192.0.2.0/24

- Create an address plan for your services
- Monitor your traffic – how much UDP and TCP do you have, roughly
- Above is a simplified example – dig deeper into your traffic

## Designing the protection – address families & protocols



Ad-dress family	Proto-col	Services and ports	Prefix
IPv4	TCP	25, 80, 8003, 443, 4443	192.0.2.0/25
IPv4	UDP	53	192.0.2.128/25
IPv6	UDP	53	2001:DB8:ABCD:0053::/60
IPv6	TCP	80 443	2001:DB8:ABCD:1000::/56

- Direction is also very important – servers that never initiate connections have fewer requirements
- How much traffic do you have that uses IPv6 yet? Should an IPv6 DDoS take up all resources?
- Maybe let IPv4 only use a part, so at least some customers can visit using IPv6?
- Maybe you can do an allow list for allocated networks, since not all is used yet



## Config example: SNMP

```
snmp {  
    description "Router-CPH-01";  
    location "Copenhagen, Denmark";  
    contact "noc@zecurity.com";  
    community yourcommunitynotmine {  
        authorization read-only;  
        clients {  
            10.1.1.1/32;  
            10.1.2.2/32;  
        }  
    }  
}
```

If you must use SNMPv2 then at least put it into separate VLAN! ⚡



## Stateless firewall filter throw stuff away

```
hlk@MX-CPH-02> show configuration firewall filter all | no-more
/* This is a sample, better to use BGP flowspec and RTBH */
inactive: 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 also 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 { count some-server-block; 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:

```
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 1024;
        attack-threshold 200;
        source-threshold 1024;
        destination-threshold 2048;
        timeout 20;
    }
    land;
} Always select your own settings YMMV
```

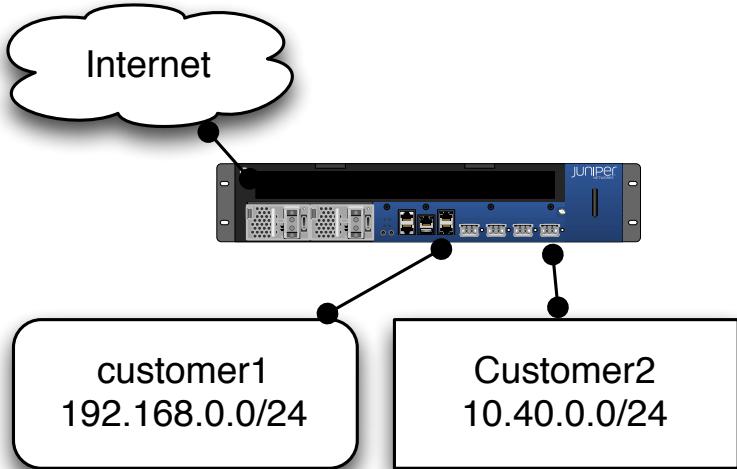


## uRPF unicast Reverse Path Forwarding

Reverse path forwarding (RPF) is a technique used in modern routers for the purposes of ensuring loop-free forwarding of multicast packets in multicast routing and to help prevent IP address spoofing in unicast routing.

Source: [http://en.wikipedia.org/wiki/Reverse\\_path\\_forwarding](http://en.wikipedia.org/wiki/Reverse_path_forwarding)

# Strict vs loose mode RPF



```
user@router# show interfaces
ge-0/0/0 {
    unit 2 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 192.168.0.254/24;
        }
    }
}
ge-0/0/1 {
    unit 2 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.40.0.254/24;
        }
    }
}
```

## Strict mode RPF



### Configuring Unicast RPF Strict Mode

In strict mode, unicast RPF checks whether the incoming packet has a source address that matches a prefix in the routing table, **and whether the interface expects to receive a packet with this source address prefix.**

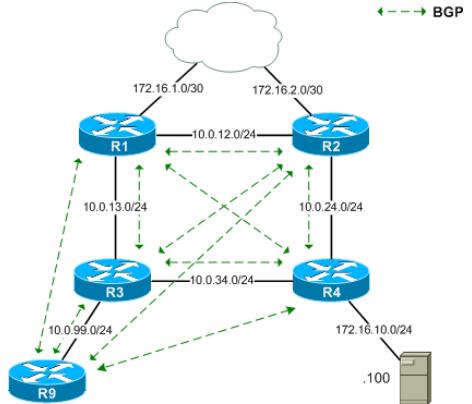
# uRPF Junos config with loose mode



```
xe-5/1/1 {  
    description "Transit: Blah (AS65512)";  
    unit 0 {  
        family inet {  
            rpf-check {  
                mode loose;  
            }  
            filter {  
                input all;  
                output all;  
            }  
            address xx.yy.xx.yy/30;  
        }  
        family inet6 {  
            rpf-check {  
                mode loose;  
            }  
            address 2001:xx:yy/126;  
        }  
    }  
}
```

See also: <http://www.version2.dk/blog/den-danske-internettrafik-og-bgp-49401>

# Remotely Triggered Black Hole Configurations



Picture from packetlife.net showing R9 as a standalone "management" router for route injection.

<http://packetlife.net/blog/2009/jul/6/remotely-triggered-black-hole-rtbh-routing/>

<https://ripe65.ripe.net/presentations/285-inex-ripe-routingwg-amsterdam-2012-09-27.pdf> <https://www.inex.ie/rtbh>

# Remotely Triggered Black Hole at upstreams



## 6. Black Hole Server (Optional)

```
#####
#          NOTE          #
# The Cogent Black Hole server will allow customers to announce a /32 route      #
# to Cogent and have all traffic to that network blocked at Cogents backbone.    #
# All peers on the Cogent black hole server require a password and IP address     #
# from your network for Cogent to peer with.                                     #
#####
```

[ ] Please set up a BGP peer on the Cogent Black Hole server

Black Hole server password:

Black Hole server peer IP:

North American Black Hole Peer: 66.28.8.1

European Black Hole Peer: 130.117.20.1

Source:

[http://cogentco.com/files/docs/customer\\_service/guide/bgpq.sample.txt](http://cogentco.com/files/docs/customer_service/guide/bgpq.sample.txt)

Better drop single /32 host than whole network!

## More information about DDoS testing



More DDoS and testing for DDoS can be found in this presentation:

### **Simulated DDoS Attacks, Breaking the Firewall Infrastructure Henrik Kramselund**

DDoS Attacks have become a daily annoyance for many, and we need to create robust infrastructure. This tutorial will go through a proposed method for testing your own infrastructure using off-the-shelf tools like packet generators hping3 and t50 on Kali Linux.

The goal for the tutorial is to explain: \* How to create DDoS attack simulations \* My actual experience with doing this - testing banks, etc. \* Evaluate how good is this, value proposition for you

Can be found at <https://ripe72.ripe.net/wp-content/uploads/presentations/32-simulated-ddos-ripe.pdf> or  
<https://github.com/kramse/security-courses/tree/master/presentations/pentest/simulated-ddos-ripe>

# 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

See also: *Security in a Mixed IPv4 and IPv6 World*, presentation by me

<https://github.com/kramse/security-courses/tree/master/presentations/network/ipv6-security-in-mixed-v4-v6>



## IPv6 is more/less secure than IPv4

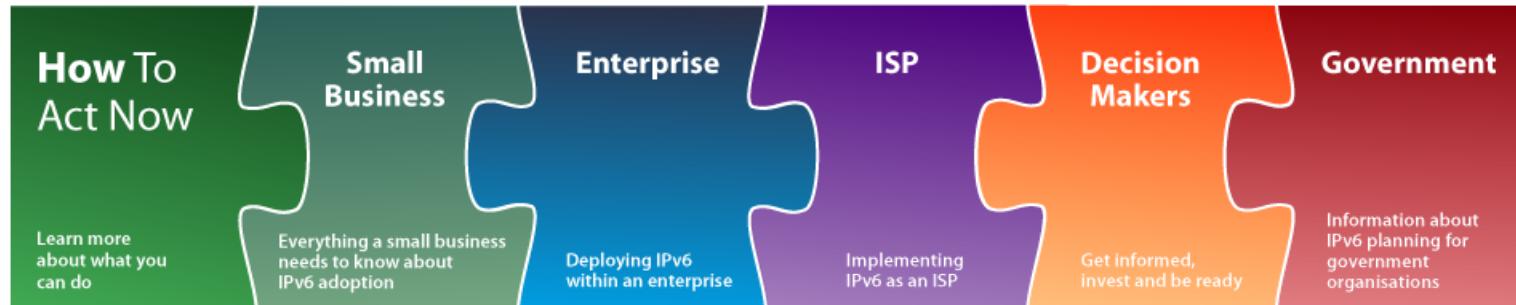
There are two big misconceptions about IPv6 security:

- IPv6 is more secure than IPv4
- IPv6 is less secure than IPv4

*Neither* are true. Both assume that comparing IPv6 security with IPv4 security is meaningful. It is not.

Source: David Holder at, <https://blog.apnic.net/2019/03/18/common-misconceptions-about-ipv6-security/>

# IPv6 is already in your network!



Picture from the IPv6 Act Now web site, RIPE NCC

- You have both, you will keep on having both
- Unless you have very strict control and turn one or the other OFF always, you have both IPv4 and IPv6 in your network!
- My suggestion, realize IPv6 is here, take control



# IPv6 Neighbor Discovery Protocol (NDP)

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

Address Resolution Protocol (ARP) is replaced

NDP expands on the ARP concept, similar command arp -an compared to ndp -an

Can do some things we knew from DHCPv4 still DHCPv6 exist

**Note ICMPv6 often need to be added to firewall rules for NDP!** 

# ARP vs NDP



So at the low level, near the hardware we have protocols connecting IP addresses with MAC addresses, Ethernet and Wi-Fi are commonly found

```
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 Expire      St Flgs Prbs
::1                           (incomplete)       lo0 permanent R
2001:16d8:ffd2:cf0f:21c:b3ff:fec4:e1b6 0:1c:b3:c4:e1:b6 en1 permanent R
fe80::1%lo0                    (incomplete)       lo0 permanent R
fe80::200:24ff:fec8:b24c%en1 0:0:24:c8:b2:4c     en1 8h54m51s S R
fe80::21c:b3ff:fec4:e1b6%en1 0:1c:b3:c4:e1:b6       en1 permanent R
```

## ARP and NDP problems



- This mapping is used in your operating system, keep a dynamic ARP/neighbor cache – a table
- Switches map devices to ports – tables
- Routers remember your IP, so it can send responses back – tables
- A table has a maximum size! This can cause problems 
- This is all done without ANY security – you can lie, attackers can lie
- See ARP spoofing and a sample tool [https://en.wikipedia.org/wiki/ARP\\_spoofing](https://en.wikipedia.org/wiki/ARP_spoofing) and <https://en.wikipedia.org/wiki/DSniff>

## Similarities between IPv4 and IPv6 security



- Rogue DHCP servers can be done in both, plus false router advertisements in IPv6
- MAC address overflow can be done in both
- Unfiltered access can be abused
- DNS spoofing can be abused
- Sniffing unencrypted traffic is the same
- MITM Attacks are the same! 
- Application attacks are the same! Example Web attack over IPv4/IPv6 - often address family doesn't matter 
- Flooding attacks are possible at various places

## Disparities between IPv4 and IPv6 security, example



The functionality provided by IPv6's Type 0 Routing Header can be exploited in order to achieve traffic amplification over a remote path for the purposes of generating denial-of-service traffic. This document updates the IPv6 specification to deprecate the use of IPv6 Type 0 Routing Headers, in light of this security concern.

Source RFC5095

- Routing headers – flexible, but hard to filter  
Updated RFC8200 *Internet Protocol, Version 6 (IPv6) Specification* recommend order for those
- IPv6 Type 0 routing header, fixed  
Deprecated officially in RFC 5095 <https://www.rfc-editor.org/rfc/rfc5095.txt>
- Other stuff better specified, like RFC5722 – see later

We are relying on vendors to create updated software, but must install those updates

# Operational Security Considerations for IPv6 Networks



Internet Engineering Task Force (IETF)  
Request for Comments: 9099  
Category: Informational  
ISSN: 2070-1721

É. Vyncke

Cisco

K. Chittimaneni

M. Kaeo

Double Shot Security

E. Rey

ERNW

August 2021

Operational Security Considerations for IPv6 Networks

Source: <https://www.rfc-editor.org/rfc/rfc9099.txt>

- Fantastic reference
- Another from RIPE NCC, 191 slides! IPv6 Security Training Course April 2021,  
<https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf>

## Address planning – helps security for both IPv4 and IPv6!



IPv6 address allocations and overall architecture are important parts of securing IPv6. Initial designs, even if intended to be temporary, tend to last much longer than expected. Although IPv6 was initially thought to make renumbering easy, in practice, it may be extremely difficult to renumber without a proper IP Address Management (IPAM) system. [RFC7010] introduces the mechanisms that could be utilized for IPv6 site renumbering and tries to cover most of the explicit issues and requirements associated with IPv6 renumbering. **A key task for a successful IPv6 deployment is to prepare an addressing plan. Because an abundance of address space is available, structuring an address plan around both services and geographic locations allows address space to become a basis for more structured security policies to permit or deny services between geographic regions.** [RFC6177] documents some operational considerations of using different prefix sizes for address assignments at end sites.

Source: RFC 9099

- You have space, use it!

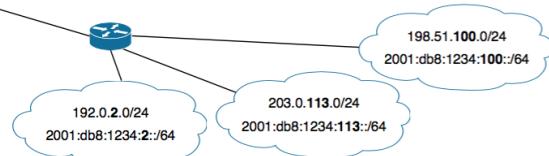
# Network Architecture and Address planning



## 3.1. Direct Link Between IPv4 and IPv6 Subnets

If the existing IPv4 networks use only /24 subnets (for example, from 203.0.113.0 to 203.0.113.255), a direct link can be established between IPv4 addresses and the new IPv6 addresses. In this case, you can include the penultimate number of the IPv4 address (113 in 203.0.113.0/24, for example) in the IPv6 subnet. The IPv6 address will then be 2001:db8:1234:113::/64.

Such an IPv4-to-IPv6 transition could appear as follows:



Source: picture from Surfnet Preparing and IPv6 Address Plan

- Take the opportunity to re-design your network! Create a design, consider it green field, work towards it!
- Use /127 for point-to-point links, add loopback addresses on routers, allows filtering of access to management
- You can also make parts IPv6-only, Veronika McKillop at TROOPERS19 *Microsoft IT (secure) journey to IPv6-only*  
<https://troopers.de/troopers19/agenda/h7sv7v/>



# 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 neighboradv 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 neighborsol keep state

... probably not working AS IS
```



## IPv4 + IPv6 serviceeksempel, med tables

PF har tabeller og simpel regel til PF konfigurationsfilen kan give adgang til både IPv4 og IPv6:

```
table <webservers> { 2001:7b8:3e4:72::20 10.0.72.20 }  
...  
pass in on $ext_if proto tcp to <webserver> port http
```



# Creating an Access Control List (ACL)

```
(config)#ipv6 access-list RA-GUARD
(config-ipv6-acl)#sequence 3 deny icmp any any router-advertisement
(config-ipv6-acl)#sequence 6 permit ipv6 any any
(config-ipv6-acl)#exit
(config)#interface FastEthernet0/5
(config-if)#ipv6 traffic-filter RA-GUARD in
```

Source: example copied from RIPE NCC IPv6 Security Training materials:

<https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf>

- Best practice, and not that hard to do
- ACL, filtering and firewalling will create longer lasting protection
- Paired with a nice address plan you can easily put restrictions on traffic flow, without hurting functionality or the business
- Does ANY client in ANY office NEEEEEED to connect to ANY UPS, Virtualisation and printer across the world ...

## Get IPv6 prefix!



You can ask RIPE NCC for an IPv6 provider independent prefix, through a LIR – I have a LIR!

- YOU can't request directly, but need to find a RIPE NCC member to request it  
Hint: Zencurity Aps is a member
- It will cost you about EUR 100 per year and you will get minimum /48
- You can move this space from provider to provider  
more easily than migrating from their IP space to some new providers space
- You can have this announced via multiple providers – redundancy
- Read more about this at:  
<https://www.ripe.net/manage-ips-and-asns/ipv6/request-ipv6/how-to-request-an-ipv6-pi-assignment>
- If you want to play with IPv6 try an IPv6 tunnel broker like  
<https://tunnelbroker.net/> or <https://www.route48.org/>

# Design a robust network Isolation and segmentation



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

## Concrete advice for enterprise networks



- Portscanning - start using portscans in your networks, verify how far malware and hackers can travel, and identify soft systems needing updates or isolation
- Have separation – anywhere, starting with organisation units, management networks, server networks, customers, guests, LAN, WAN, Mail, web, ...
- Use Web proxies - do not allow HTTP directly except for a short allow list, do not allow traffic to and from any new TLD
- Use only your own DNS servers, create a pair of Unbound servers, point your internal DNS running on Windows to these  
Create filtering, logging, restrictions on these Unbound DNS servers  
<https://www.nlnetlabs.nl/projects/unbound/about/> and also <https://pi-hole.net/>
- Only allow SMTP via your own mail servers, create a simple forwarder if you must

Allow lists are better than block list, even if it takes some time to do it



## Capture data and logs!

- Run DNS query logs – when client1 is infected with malware from domain malwareexample.com, then search for more clients infected
- Run Zeek and gather information about all HTTPS sessions – captures certificates by default, and we can again search for certificate related to malwareexample.com
- Run network logging – session logs in enterprise networks are GREAT (country wide illegal logging is of course NOT)

Make sure to check with employees, inform them!

# DROP SOME TRAFFIC NOW



- Drop some traffic on the border of everything
- Seriously do NOT allow Windows RPC across borders
- Border here may be from regional country office back to HQ
- Border may be from internet to internal networks
- Block Windows RPC ports, 135, 137, 139, 445
- Block DNS directly to internet, do not allow clients to use any DNS, fake 8.8.8.8 if you must internally
- Block SMTP directly to internet
- Create allow list for internal networks, client networks should not contact other client networks but only relevant server networks

You DONT need to allow direct DNS towards internet, except from your own recursive DNS servers

If you get hacked by Windows RPC in 2022, you probably deserve it, sorry for being blunt

Best would be to analyze traffic and create allow lists, some internal networks to not need internet at all

# Default permit



One of the early implementers of firewalls Marcus J. Ranum summarized in 2005 The Six Dumbest Ideas in Computer Security [https://www.ranum.com/security/computer\\_security/editorials/dumb/](https://www.ranum.com/security/computer_security/editorials/dumb/) which includes the always appropriate discussion about default permit versus default deny.

## #1) Default Permit

This dumb idea crops up in a lot of different forms; it's incredibly persistent and difficult to eradicate. Why? Because it's so attractive. Systems based on "Default Permit" are the computer security equivalent of empty calories: tasty, yet fattening.

The most recognizable form in which the "Default Permit" dumb idea manifests itself is in firewall rules. Back in the very early days of computer security, network managers would set up an internet connection and decide to secure it by turning off incoming telnet, incoming rlogin, and incoming FTP. Everything else was allowed through, hence the name "Default Permit." This put the security practitioner in an endless arms-race with the hackers.

- Allow all current networks today on all ports for all protocols *is* an allow list  
Which tomorrow can be split into one for TCP, UDP and remaining, and measured upon
- Measure, improve, repeat

## We cannot do X



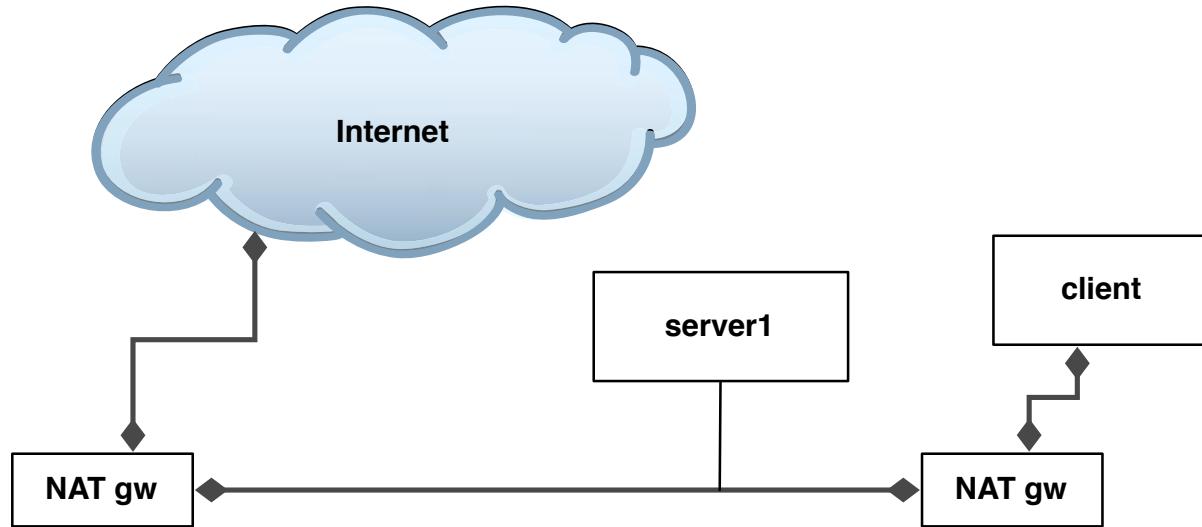
We cannot block SMTP from internal networks, since we do not know for sure if vendor X equipment needs to send the MOST important email alert at some unspecific time in the future

Cool, then we can do an allow list starting today on our border firewall:

```
table <smtp-exchange> { $exchange1 $exchange2 $exchange3 }
table <smtp-unknown> persist file "/firewall/mail/smtp-internal-unknown.txt"
# Regular use, allowed
pass out on egress inet proto tcp from smtp-exchange to any port 25/tcp
# Unknown, remove when phased out
pass out on egress inet proto tcp from smtp-internal to any port 25/tcp
```

Year 0 the unknown list may be 100% of all internal networks, but new networks added to infrastructure are NOT added, so list will shrink – evaluate the list, and compare to network logs, did networks send ANY SMTP for 1,2,3 years?

## 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!

## Pattern use IDS to get flow, connections and data



Use Intrusion Detection Systems - IDS

Angrebsværktøjerne efterlader spor

Det anbefales at have IDS og flow opsamling som minimum

Hostbased IDS - kører lokalt på et system og forsøger at detektere om der er en angriber inde

Network based IDS - NIDS - bruger netværket

Automatiserer netværksovervågning:

- bestemte pakker kan opfattes som en signatur
- analyse af netværkstrafik - FØR angreb
- analyse af netværk under angreb - sender en alarm

# The Zeek Network Security Monitor



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



**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.

Zeek is the tool formerly known as Bro, changed name in 2018. <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>

Workshop materials available:

<https://github.com/kramse/security-courses/tree/master/courses/networking/suricatazeek-workshop>



## Anti-pattern blokering af ALT ICMP

```
# Simple stateful network firewall rules for allowing ICMP in IPv6 ICMPv6
# Allow ICMPv6 destination unreach
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 1
# Allow NS/NA/toobig (don't filter it out)
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 2
# Allow timex Time exceeded
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 3
# Allow parameter problem
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 4
# IPv6 ICMP - echo request (128) and echo reply (129)
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 128,129
# IPv6 ICMP - router solicitation (133) and router advertisement (134)
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 133,134
# IPv6 ICMP - neighbour discovery solicitation (135) and advertisement (136)
    $fwcmd6 add pass ipv6-icmp from any to any icmptypes 135,136
```

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

## Anti-pattern blokering af DNS opslag på TCP



Det bliver (er) nødvendigt med DNS opslag over TCP

Store svar kræver TCP

Det betyder at firewalls skal tillade DNS opslag via TCP

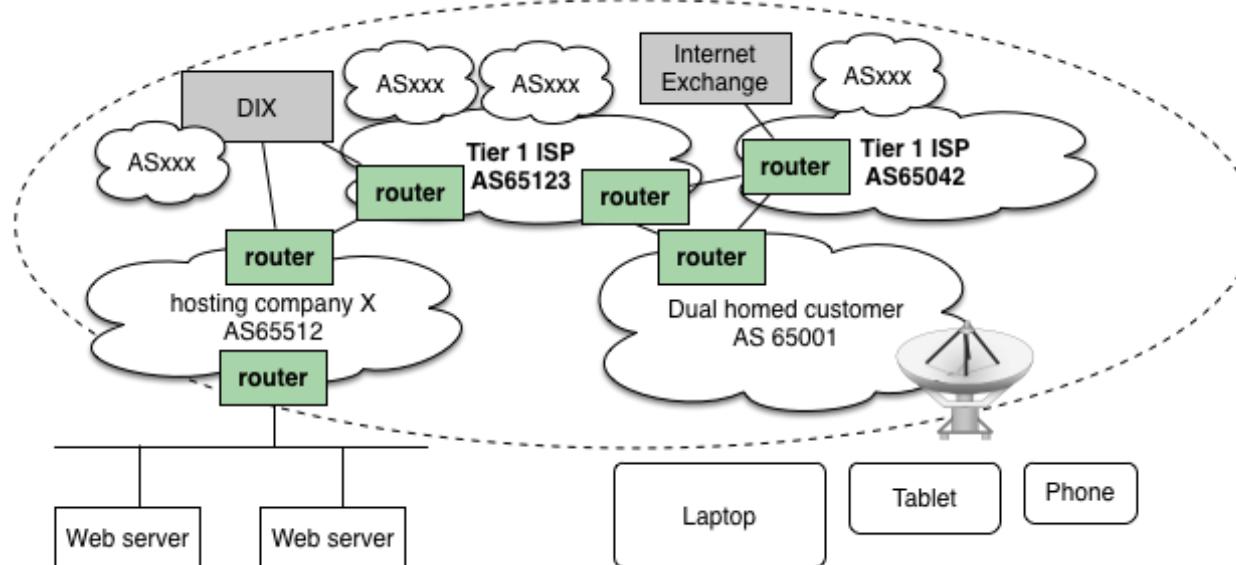
De nye forslag DNS over TLS (DoT) og DNS over HTTPS (DoH)

DNS kryptering bliver med TCP

Anbefaling i enterprise netværk:

Brug en caching nameserver, således at det kun er den som kan lave DNS opslag ud i verden

# Hosting og internet-udbydere



- Data krydser mange internetudbydere
- Det er stadig muligt at spoofe mange steder fra

# Routing and BGP Solutions



- Filtering, ingress / egress:  
"reject external packets that claim to be from the local net"
- See also Reverse Path forwarding [https://en.wikipedia.org/wiki/Reverse-path\\_forwarding](https://en.wikipedia.org/wiki/Reverse-path_forwarding)
- Routers and routing protocols must be more skeptical  
Routing filters implemented everywhere, auth on routing protocols OSPF/BGP etc.
- Has been recommended for some years, but not done in all organisations
- BGP routing Resource Public Key Infrastructure RPKI
- BCP38 is RFC2827: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*  
<http://www.bcp38.info/>
- *Mutually Agreed Norms for Routing Security*, <https://www.manrs.org/>

# Mutually Agreed Norms for Routing Security (MANRS)



Mutually Agreed Norms for Routing Security (MANRS) is a global initiative, supported by the Internet Society, that provides crucial fixes to reduce the most common routing threats.

Problems related to incorrect routing information

Problems related to traffic with spoofed source IP addresses

Problems related to coordination and collaboration between network operators

<https://www.manrs.org/isps/>

[https://www.manrs.org/wp-content/uploads/2018/09/MANRS\\_PDF\\_Sep2016.pdf](https://www.manrs.org/wp-content/uploads/2018/09/MANRS_PDF_Sep2016.pdf)

# Expected Actions in MANRS for Network Operators



1. Prevent propagation of incorrect routing information

Clear routing policies and systems for correctness, route import filters

2. Prevent traffic with spoofed source IP addresses

Validate source address from end-users and infrastructure, use BCP38

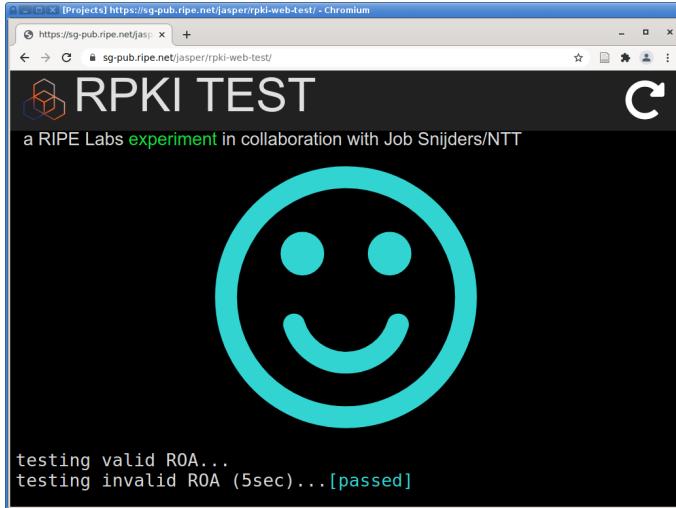
3. Facilitate global operational communication and coordination between network operators

Maintain contact information in databases like Whois, PeeringDB <https://www.peeringdb.com/>  
Advanced

4. Facilitate validation of routing information on a global scale

Use RPSL [https://en.wikipedia.org/wiki/Routing\\_Policy\\_Specification\\_Language](https://en.wikipedia.org/wiki/Routing_Policy_Specification_Language)

# RPKI Testing



- Check your own networks! Ask your ISP to check RPKI  
<https://sg-pub.ripe.net/jasper/rpki-web-test/>
- Read more about RPKI at:  
<https://www.ripe.net/manage-ips-and-asns/resource-management/rpki>