Welcome to

3. Traffic Inspection and Firewalls

Communication and Network Security 2024

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

Slides are available as PDF, kramse@Codeberg

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

Goals for today

- Introduce a network tool for isolation, firewalls
- Discuss what they *filter on* – network packets
- See how they relate to VLANs – logical/virtual seperation for hosts

Photo by Thomas Galler on Unsplash

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

- Nmap scanning firewalls
- Nmap full scan - strategy
- Nmap reporting

Introduce firewalls

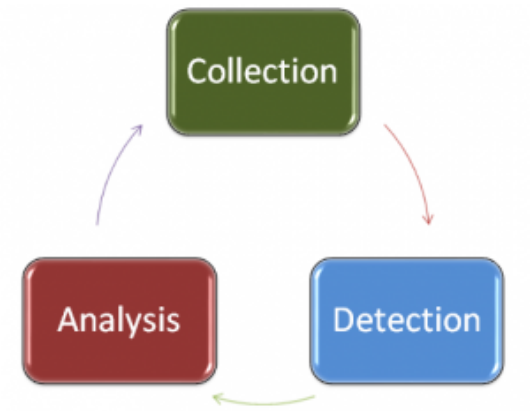
Reading Summary

ANSM chapter 1,2,3 - 73 pages
https://en.wikipedia.org/wiki/Firewall_(computing)
http://www.wilyhacker.com/ Cheswick chapter 2 og 3 PDF, ca 55 pages

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

The next time you are at your console, review some logs. You might think. . . "I don't know what to look for". Start with what you know, understand, and don't care about. Discard those. Everything else is of interest.
Semper Vigilans, Mike Poor

Reading Summary, continued



ANSM chapter 1: The Practice of Applied Network Security Monitoring

- Vulnerability-Centric vs. Threat-Centric Defense
- The NSM cycle: collection, detection, and analysis
- Full Content Data, Session Data, Statistical Data, Packet String Data, and Alert Data
- Security Onion is nice, but a bit over the top - quickly gets overloaded

Reading Summary, continued

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

Source: Wikipedia

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

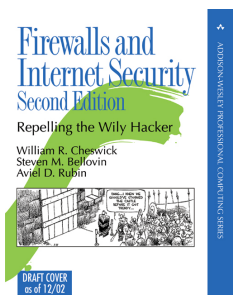
http://www.wilyhacker.com/

Cheswick chapter 2 PDF *A Security Review of Protocols: Lower Layers*

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

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

Reading Summary, continued



http://www.wilyhacker.com/

Cheswick chapter 3 PDF *Security Review: The Upper Layers*

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

Security Onion

Security Onion is a Linux distro for IDS (Intrusion Detection) and NSM (Network Security Monitoring).
http://securityonion.net

Nice starting point for researching dashboards/network packets

Baseline Skills

- 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, 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

Reading Summary, continued

ANSM chapter 2: Planning Data Collection

- The Applied Collection Framework (ACF)
- The ACF involves four distinct phases: Identify threats to your organization, quantify risk, identify relevant data feeds, and refine the useful elements
- Risk Analysis
- Lots of terms used, but only defined later in the book

Reading Summary, continued

ANSM chapter 3:The Sensor Platform

- Full Packet Capture (FPC) Data
- Session Data
- Statistical Data
- Packet String (PSTR) Data
- Log Data
- Sensor Placement, designing etc.

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!

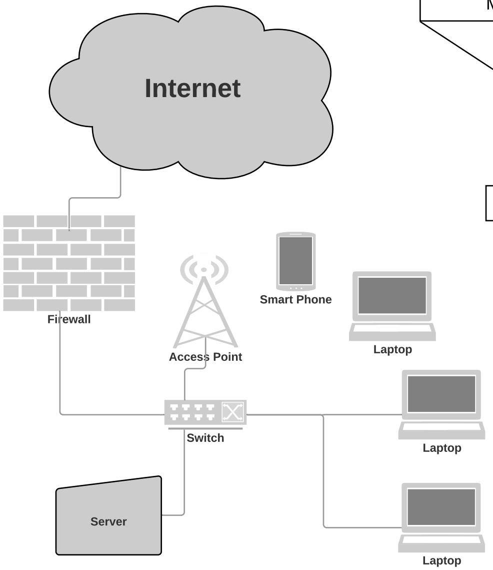
Internet in a Box

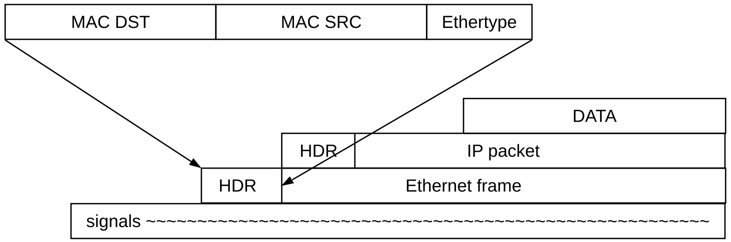
The main purpose of showing this box, are:

- 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





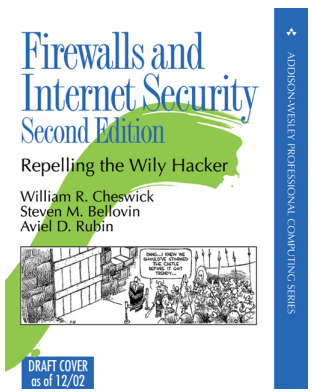
Ethernet frames contain packets

```
 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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet IPv4 Packet Header

- Network traffic is sent using network protocols with fields, values and destinations
- Can we inspect this and decide to pass or block traffic, certainly! Does it help, sure!

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.[1] A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.[2]

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

> 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

What is a packet filter

We may want to distinguis between different types of firewalls/devices:

- 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!

They are all firewalls – or firewall devices!

Definition of firewalls – Wikipedia

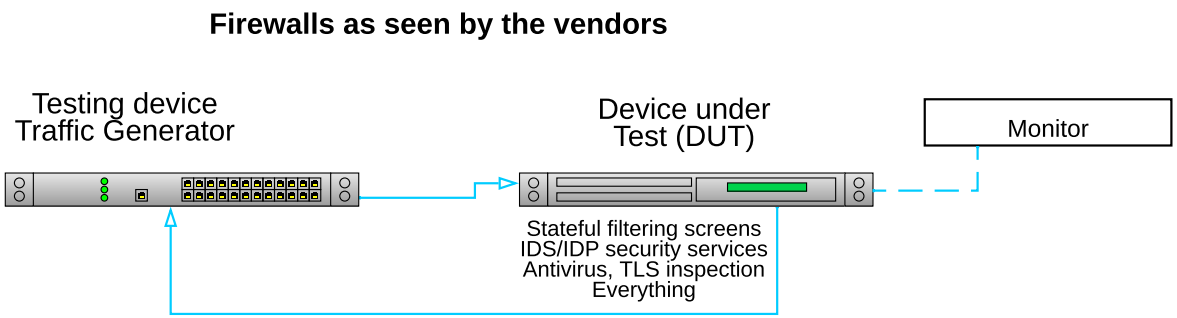
Another short definition that encapsulates this is found on Wikipedia, and may suffice in many situations. Again there will typically be multiple networks, zones or areas of the networks with varying degrees of trust.

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

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

TL;DR Not necessarily a single device

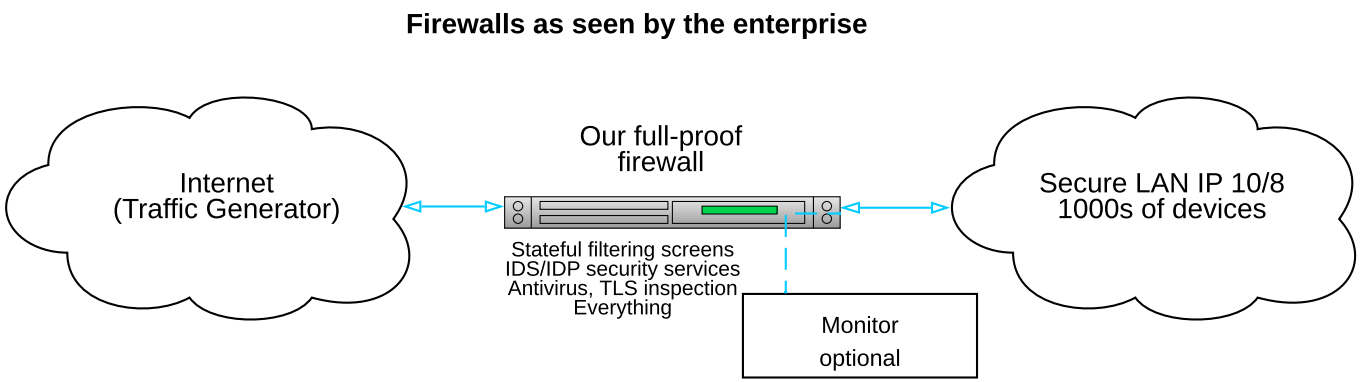
A firewall – in the vendor eyes



"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**

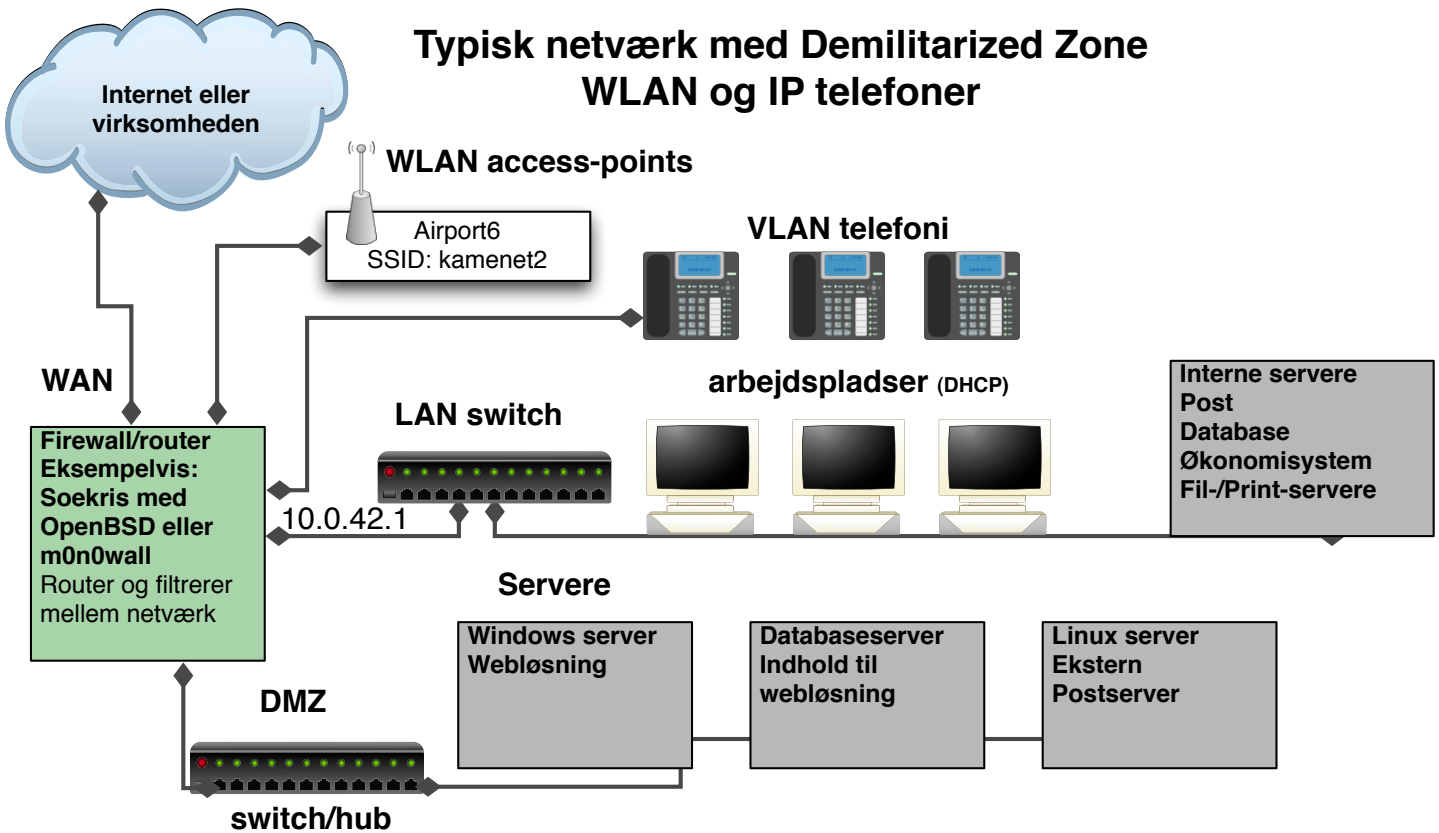
Network Protocol Knowledge Needed for Network Security

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 **RFC0791**. 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

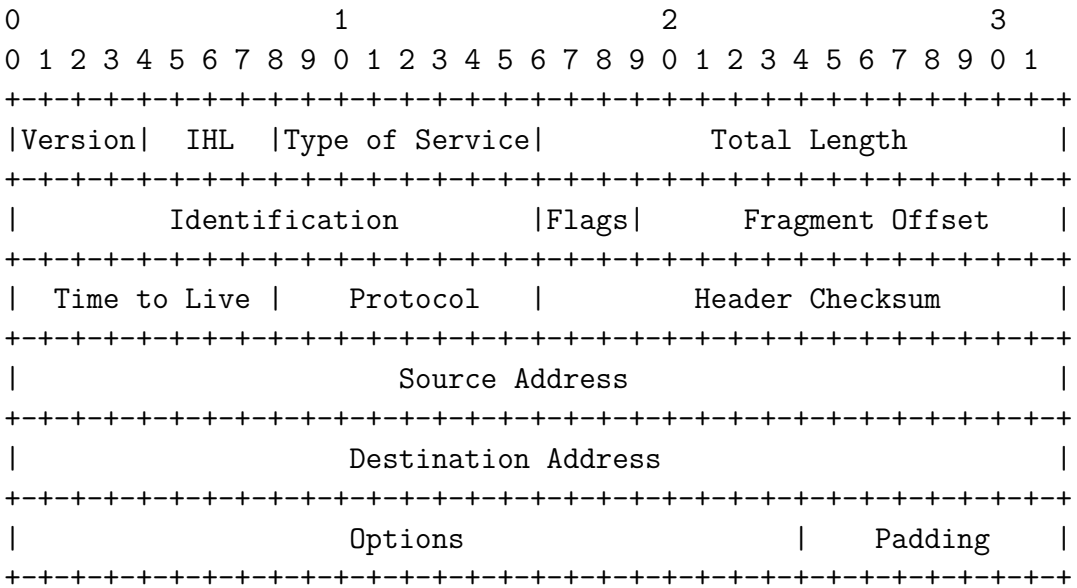
The interesting part is typically what it allows!

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



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

Modern Firewalls

Basically some filtering between networks or network segments

Typically they contain:

- Some interface, maybe web interface, often command line interface
- TCP/IP filtering options – packets flowing in and out, direction, protocol, ports etc.
- Should be able to handle both IPv6 and legacy IPv4
- Often they have predefined rules for common use-cases
  Is this really a good thing if you can easily configure a bad protocol like Server Message Block to and from the Internet?
- Most legacy setups use Network Address Translation (NAT) – NAT is a kludge and bad!
- Most platforms have extra network related features DHCP servers, DNS caching servers etc.

The firewall devices are mostly allowing some **stateful filtering** which are much easier to configure than a pure network packet filter

Goal is to implement rules – a security policy for isolation and data flow

Sample rules from OpenBSD PF

```
# hosts and networks
router="192.0.2.1"
webserver="192.0.2.80"
homenet=" 198.51.100.0/24, 203.0.113.0/24 "
wlan="198.51.100.0/24
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

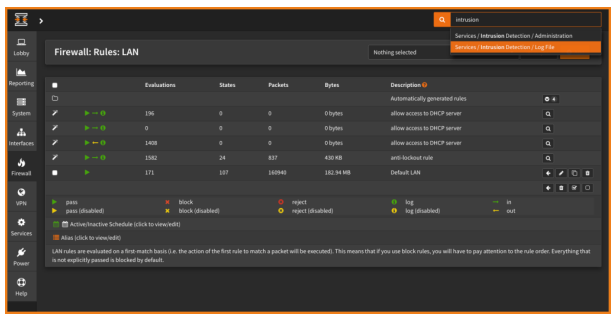
- Checkpoint Firewall-1 http://www.checkpoint.com
- Cisco Firewall http://www.cisco.com
- Juniper SRX http://www.juniper.net
- Palo Alto https://www.paloaltonetworks.com/
- Fortinet https://www.fortinet.com/
- Multiple others exist

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

Open source based firewalls

- Linux firewalls based on the in-kernel Netfilter, recommend using 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
- Mac OS X uses OpenBSD PF
- NetBSD IPFilter (IPF) by default and has also NPF, their PF version is outdated
- FreeBSD PF, IPFW og IPFilter (IPF) http://www.freebsd.org
- The other BSDs Net, Free and Mac OS X has older version of the OpenBSD PF, should really be renamed now

OPNsense GUI based and easy to install



OPNsense https://opnsense.org/

Firewall built on FreeBSD with web interface

Originally thoughts from m0n0wall and later https://www.pfsense.org/

Danish companies have been using these for many years now

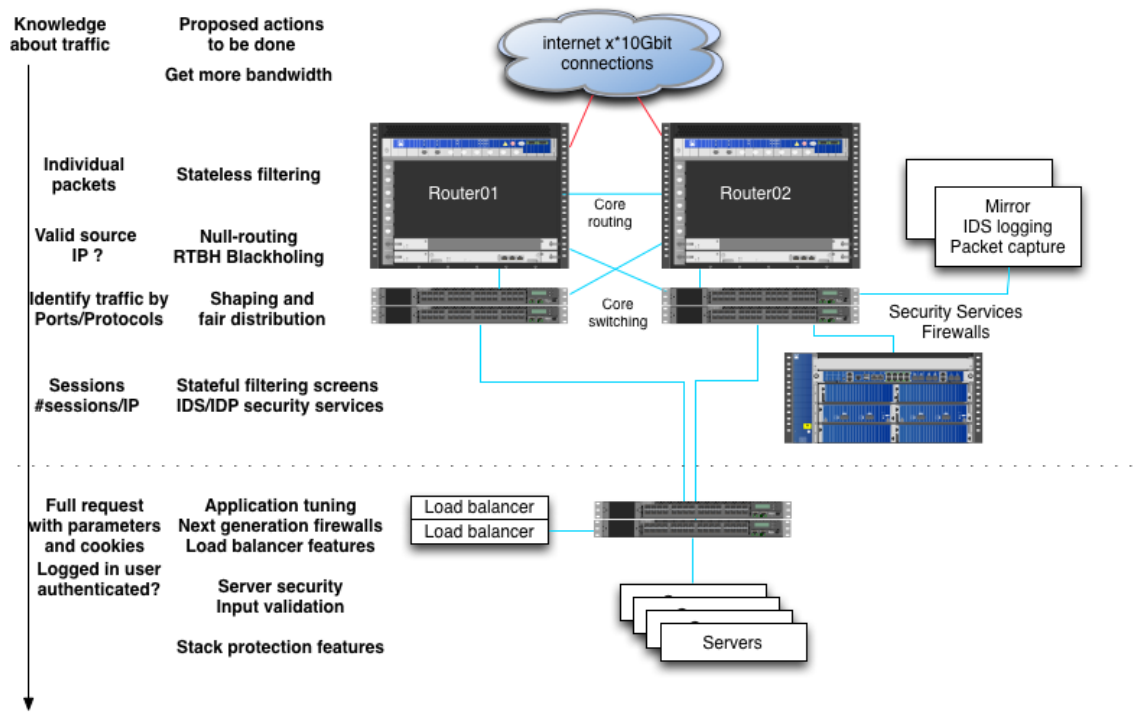
Uncomplicated Firewall (UFW)

```
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)
```

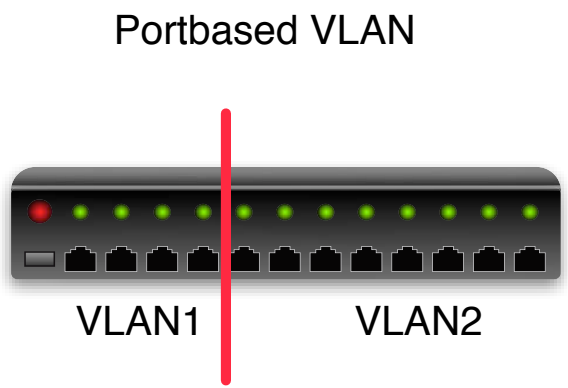
- Extremely easy to use – I recommend and use the (Uncomplicated Firewall) UFW

Specialized Firewall devices are NOT Alone



Use Defense in Depth – all layers have features

Together with Firewalls - Virtual LAN (VLAN)



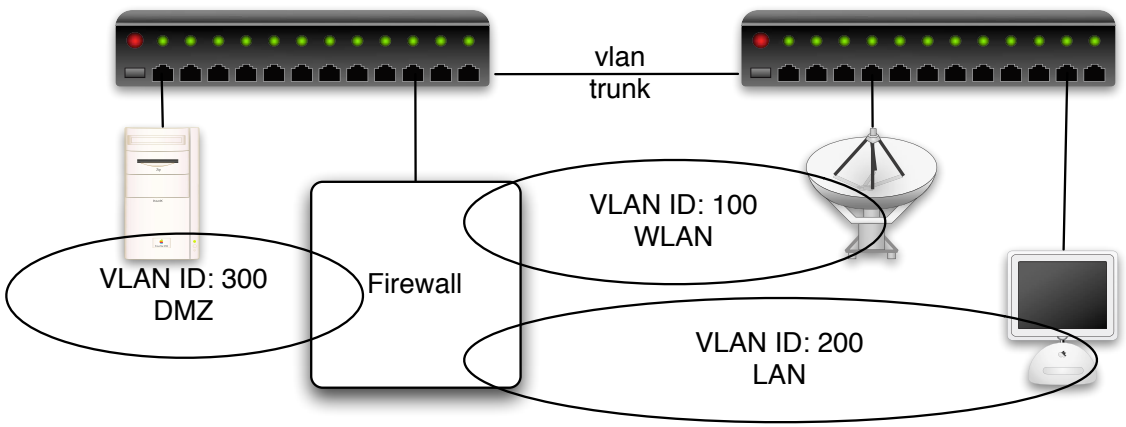
Managed switches often allow splitting into zones called virtual LANs

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

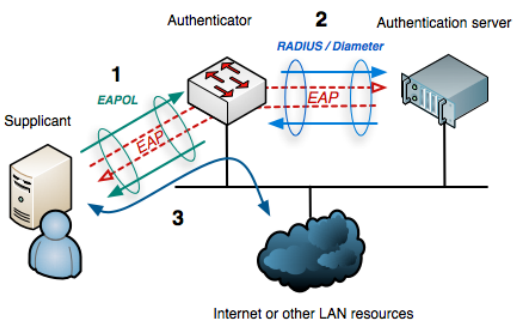
Virtual LAN, to pass from one to another, must use a router/firewall

Allows separation/segmentation and protects traffic from many security issues

Used in most, if not all, Wi-Fi networks – each SSID has a VLAN behind it

Network Access Control – Connecting clients more securely

Talking about standard, another useful one:
IEEE 802.1x – Port Based Network Access Control



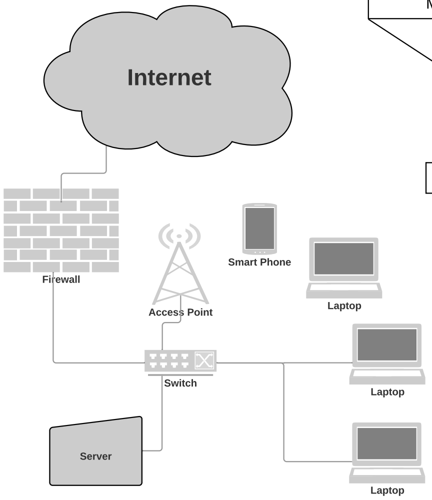
Authentication protocol ensures user validation before port access

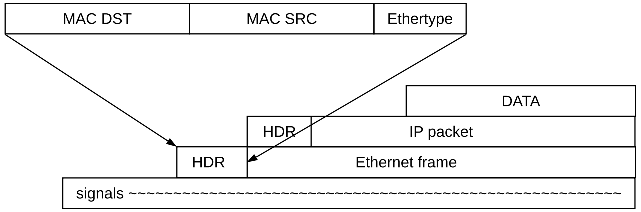
Can authenticate using username and then password or certificate

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





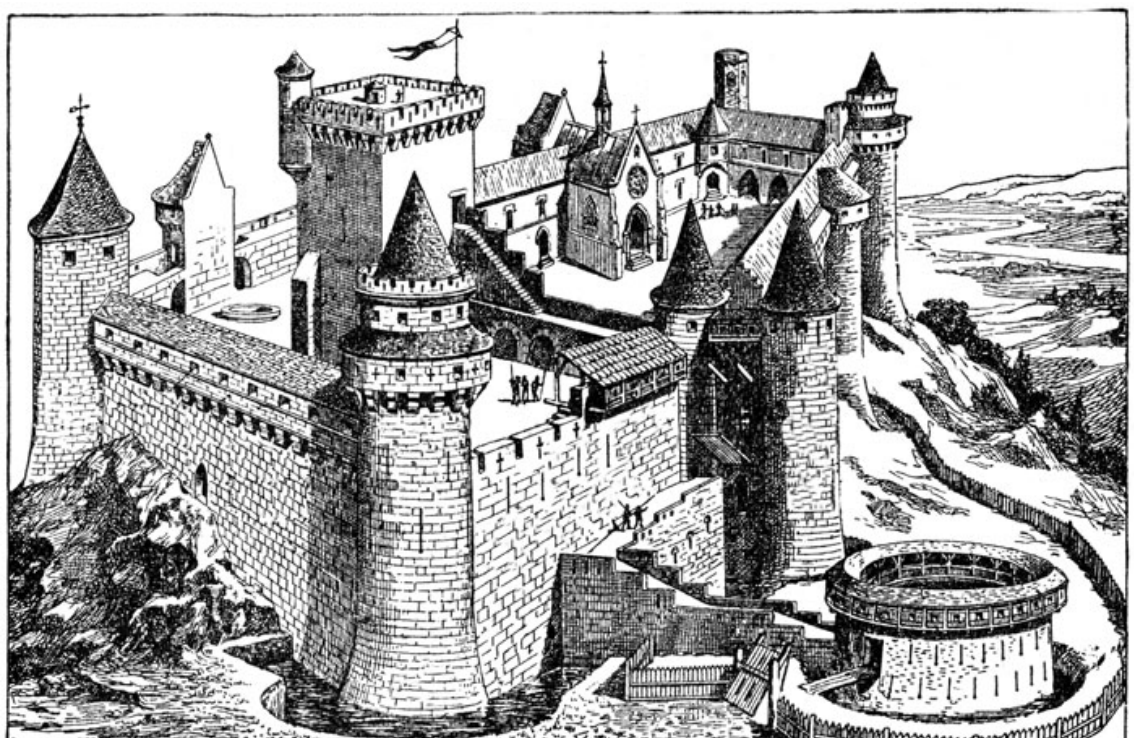
Ethernet frames contain packets

```
 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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet IPv4 Packet Header

- 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

Firewall concepts

When writing firewall rules there are differences

Some firewalls are *first match* and some are *last match*

- First match – when the packet being inspected match a rule the action block/pass is performed immediately
- Last match – whenever a packet matches a rule, mark the block/pass decision, keep going to the last rule, and **then** perform action

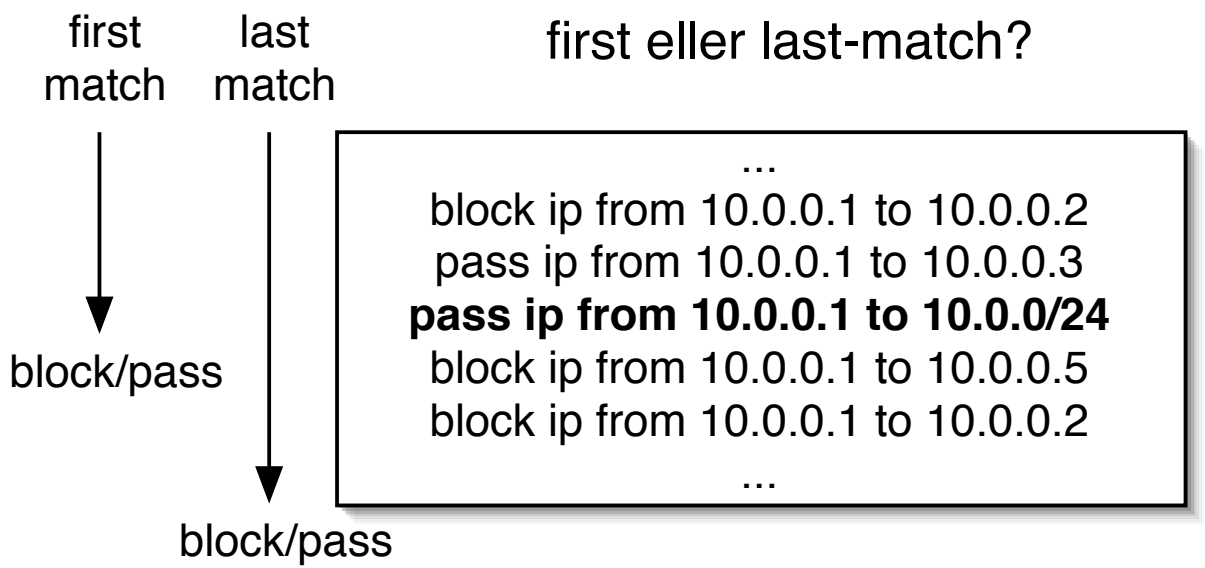
So beware which kind of firewall you are working on

FreeBSD IPFW is first match

OpenBSD PF is last 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 - example 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
```

This is a stateful filtering ruleset, rule 300 checks if an existing connection is known matching the packet incoming

The deny rule in bold show that the current configuration is a *best current practice* default deny

Last match - example OpenBSD PF

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

block in

```
pass out keep state

pass quick on { lo $int_if }

# Allow port http=80 and port domain=53
# on the IP of the IP of the external interface ($ext_if)
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
```

This is also a default deny, all packets incoming are maked as block, so unless they are allowed by a later rule – thrown away

Example Linux iptables/netfilter

```
# 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
```

Anti-pattern blocking 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
```

- Don't throw away anything ICMP, kills functionality like Path MTU
- The first four ones in bold are *needed*

Firewall configuration

Best firewall starts with the design

- Drawings – lots of drawings and topology
- An addressing plan! This is very important
- Then use a GUI for your first experience
- Plan for long term care
- Plan for updates
- Systems and services behind the firewall must still be hardened and configured securely

Block outgoing traffic too

Some services should *not* cross firewalls, at least not to the internet

Some services are too *fragile*

- Windows SMB file sharing is *only* for small internal networks
- Unix NFS is like-wise *only* for internal use
- Outgoing email should only go via dedicated relays
- LDAP outgoing, why?! See the log4j CVE-2021-44228
- Create a list, document them and consider them dead!

Making a positive list of allowed protocols would be best, but may require too many resources to implement and update

Special features

- Network Address Translation - NAT
- IPv6 functionality is not an option
- Rules for bandwidth restriction
- VLAN is a requirement
- Redundante firewalls – OpenBSD uses pfsync and CARP
- VPN like IPsec, L2TP, TLS VPN, Wireguard features
- Deep inspection – can maybe filter DNS domains, URLs or similar

Proxy servers and Web Application Firewalls (WAF)

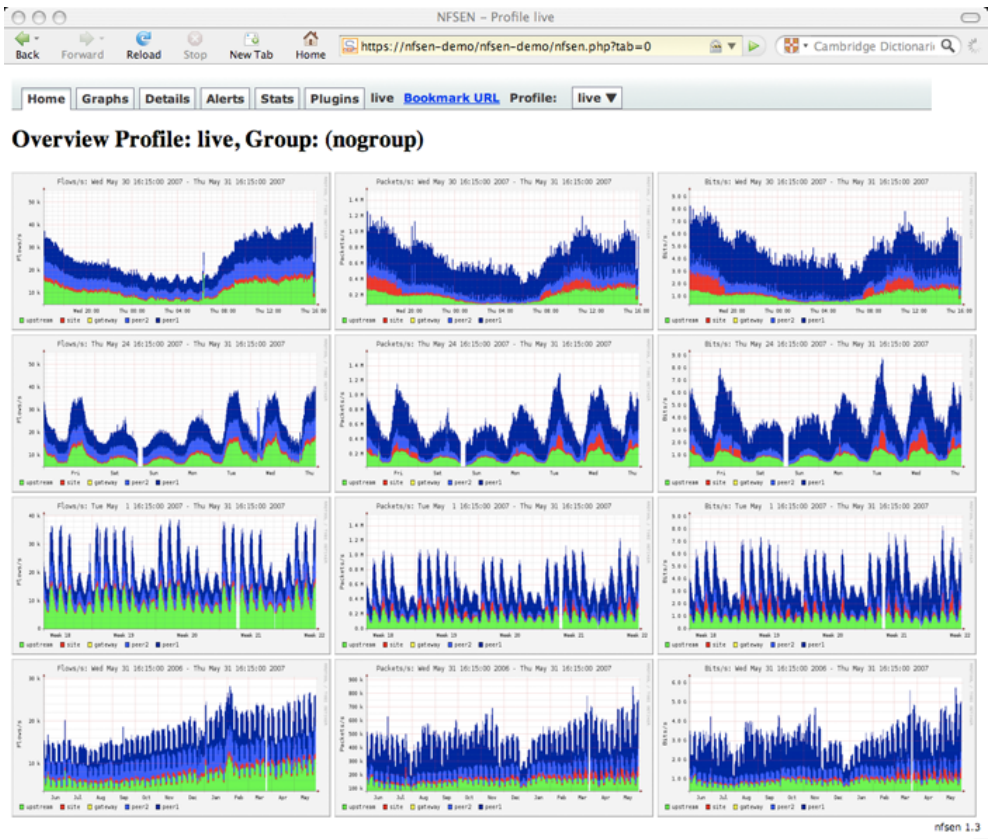
- Filtrering at higher layers is also possible
- Web proxies for clients can help security a lot – a centralized filter for everyone
- Reverse proxies for web applications are called Web Application Firewalls (WAF) – and filter incoming web requests, and outgoing answers. Can help with attacks like SQL injection and exfiltration of data
- Depending on your network it can replace or be combined with filtering on DNS servers, and I would prefer to filter domains with DNS
- I would also prefer blocking large prefixes of IP destinations using routers/stateless packet filters – maybe use BGP for distributing *lists*

Netflow and Session Logging

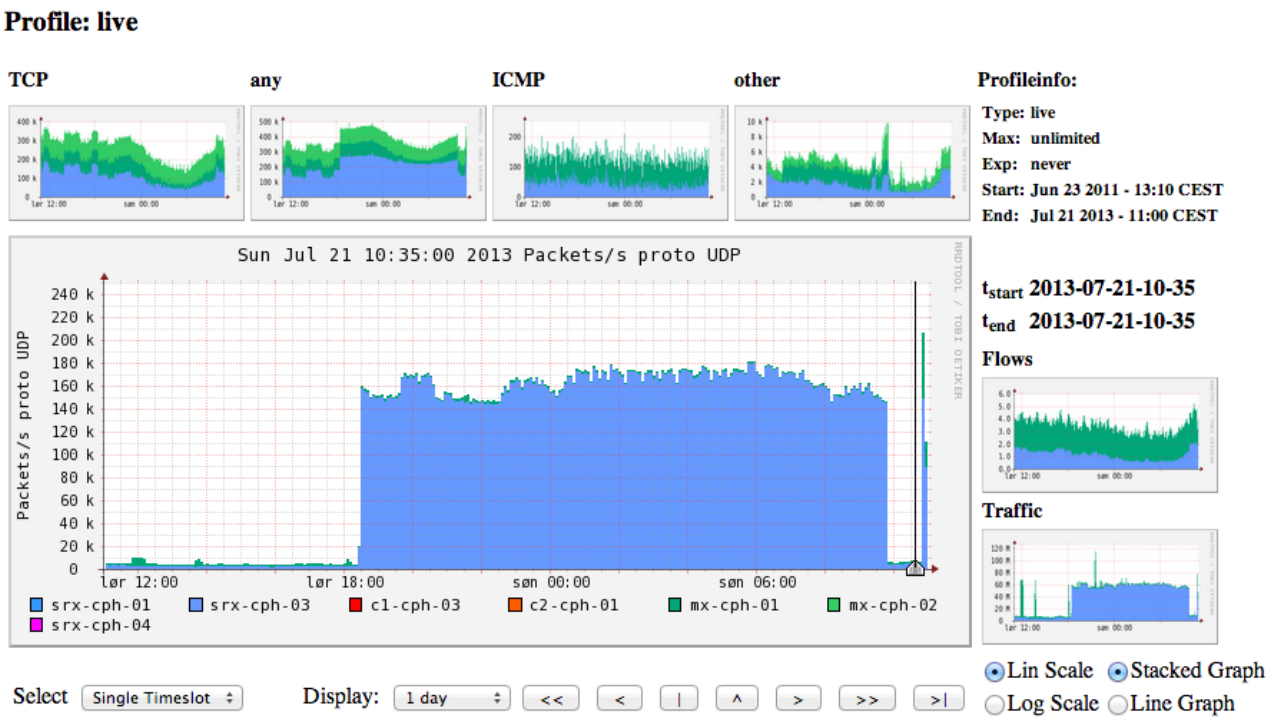
- Netflow is getting more important, more data share the same links
- Accounting is important
- Detecting DoS/DDoS and problems is essential
- Netflow sampling is vital information - 123Mbit, but what kind of traffic
- NFSen is an old but free application http://nfsen.sourceforge.net/
- Currently also 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

Netflow using NFSen

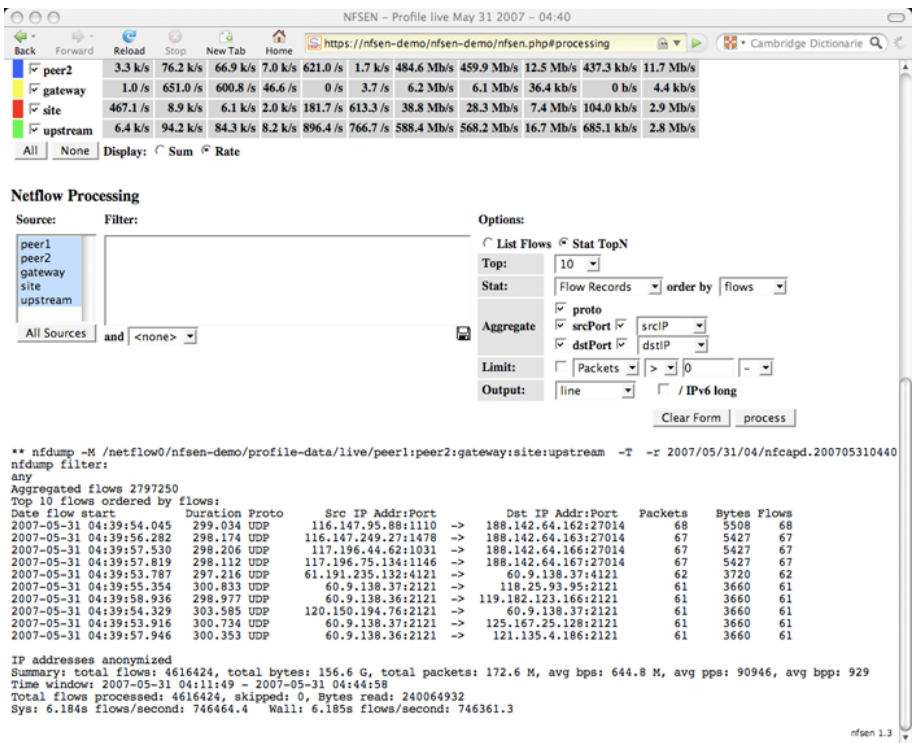


Netflow NFSen



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

Netflow processing from the web interface



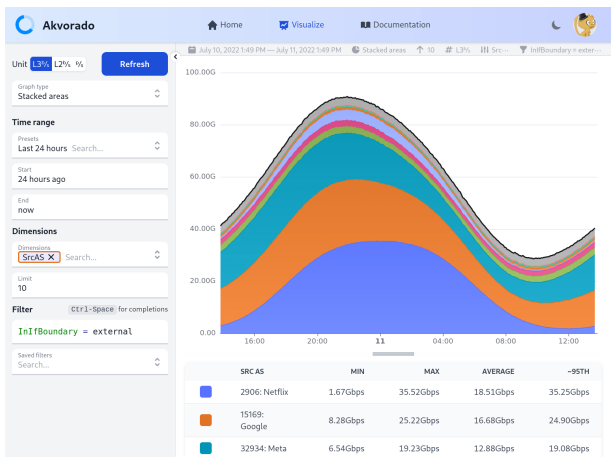
Bringing the power of the command line forward

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

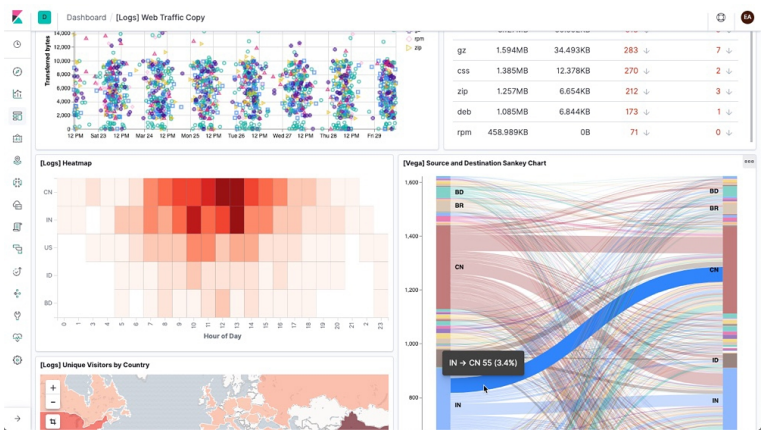
Akvorado: flow collector, enricher and visualizer



This program receives flows (currently Netflow/IPFIX and sFlow), enriches them with interface names (using SNMP), geo information (using IPinfo.io), and exports them to Kafka, then ClickHouse. It also exposes a web interface to browse the collected data.

Source: Picture and text from https://github.com/akvorado/akvorado

Big Data tools: Elasticsearch and Kibana



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!

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

Installing Cloud Firewalls Cilium example

What about cloud platforms and *firewalls*

I use Kubernetes with 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 1.15.1 \
    --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=1.15.1 \
               --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 1.15.1
  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 1.15.1 --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`

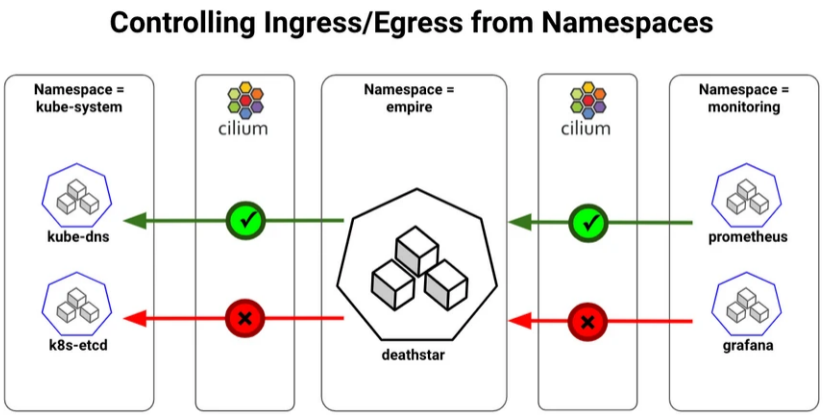
After install Cilium

```
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

- 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!

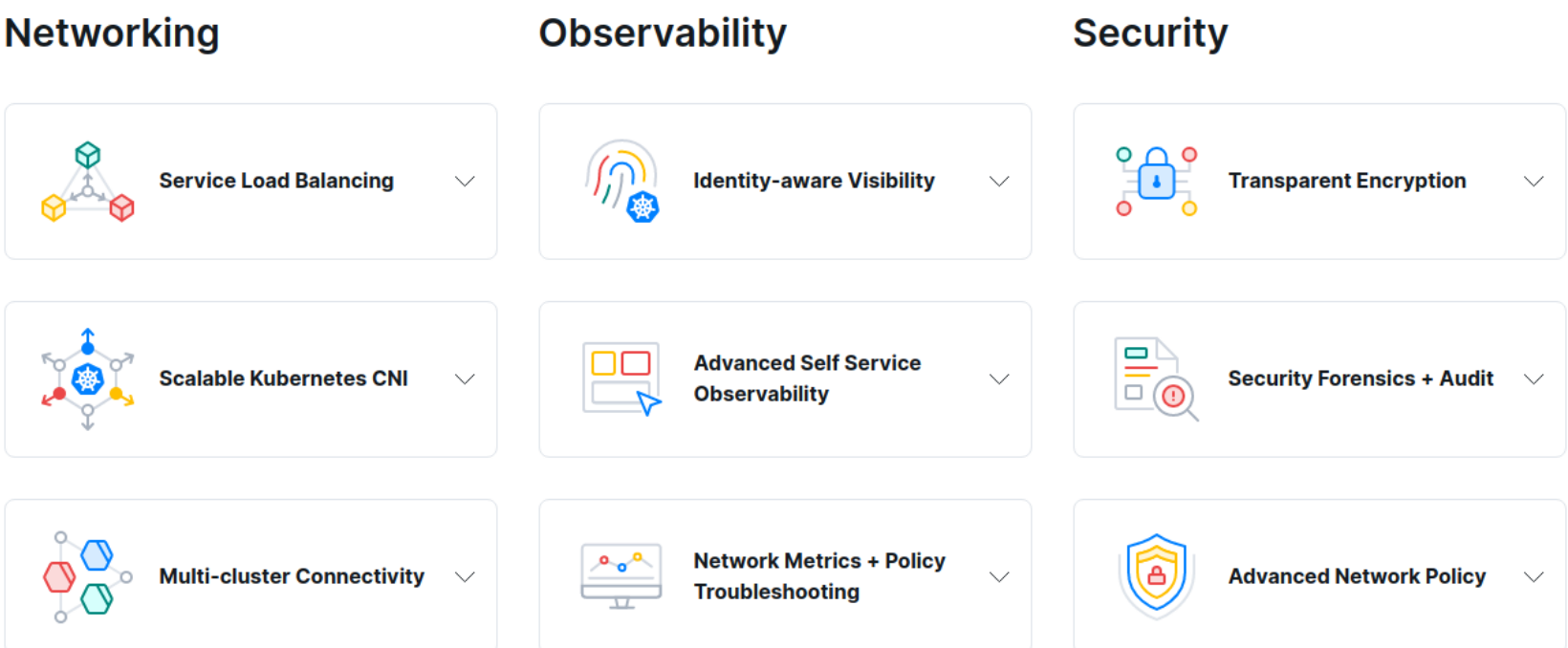
Cilium overview



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

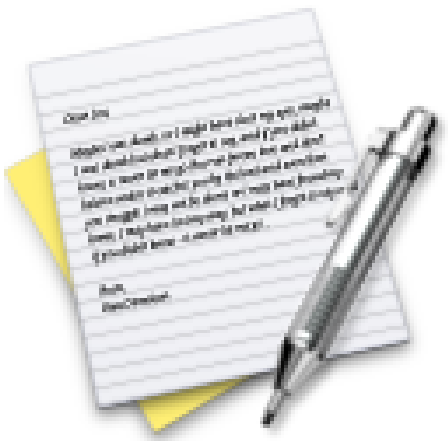
Source: picture and text from https://cilium.io/blog/2018/09/19/kubernetes-network-policies/

Security is more than blocking!



- A lot of features relate to *security*

Exercise



Now lets do the exercise

⚠ Perform nmap service scan 10 min

which is number **28** in the exercise PDF.

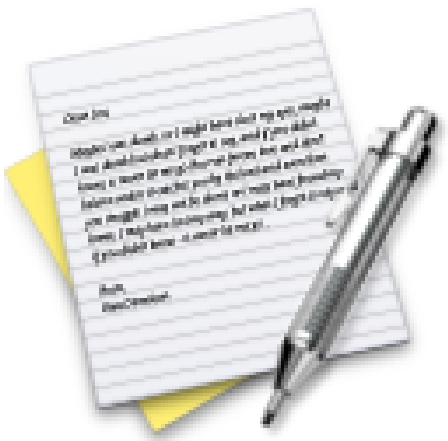
Exercise

Now lets do the exercise

⚠ SSH scanners - 15min

which is number **29** in the exercise PDF.

Exercise



Now lets do the exercise

Configure SSH keys for more secure access

which is number **30** in the exercise PDF.

Exercise

Now lets do the exercise

⚠ Nmap full scan - strategy 15 min

which is number **31** in the exercise PDF.

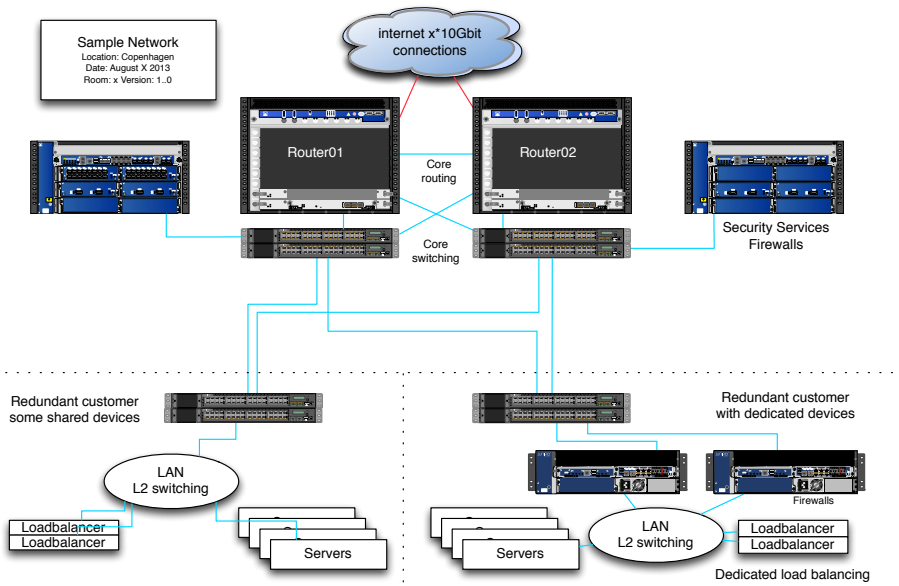
Exercise

Now lets do the exercise

Reporting HTML 15 min

which is number 32 in the exercise PDF.

Bottlenecks exist, but where



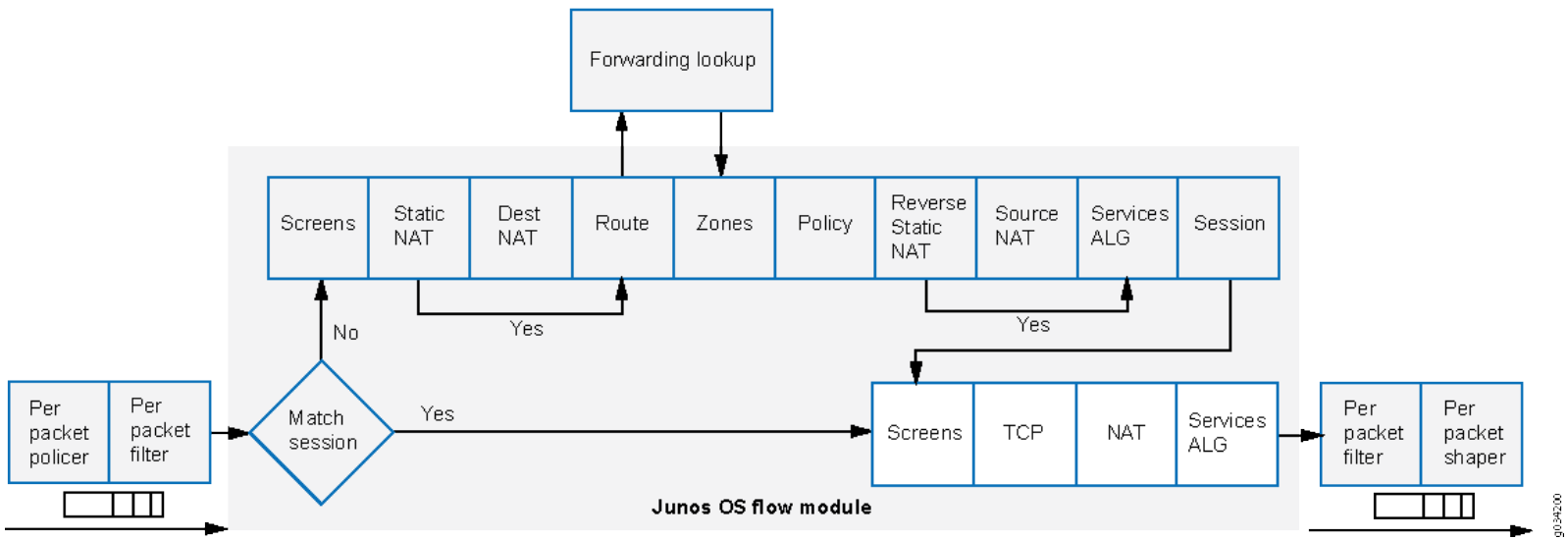
- Lower layer attacks Transport Layer Attacks TCP SYN flood – packet based
- Higher layer attacks like Slowloris and web attacks – keep sessions running
- Protect everything without loosing functionality or creating administrative nightmare

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

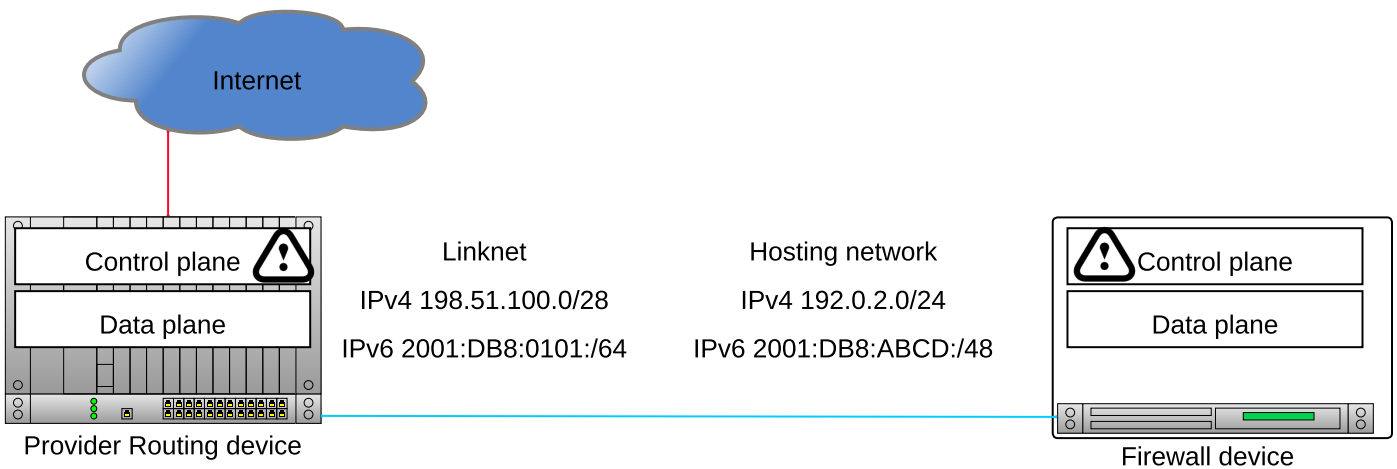
Packet processing in firewalls – detailed view



Traffic Processing on SRX Series Devices Overview

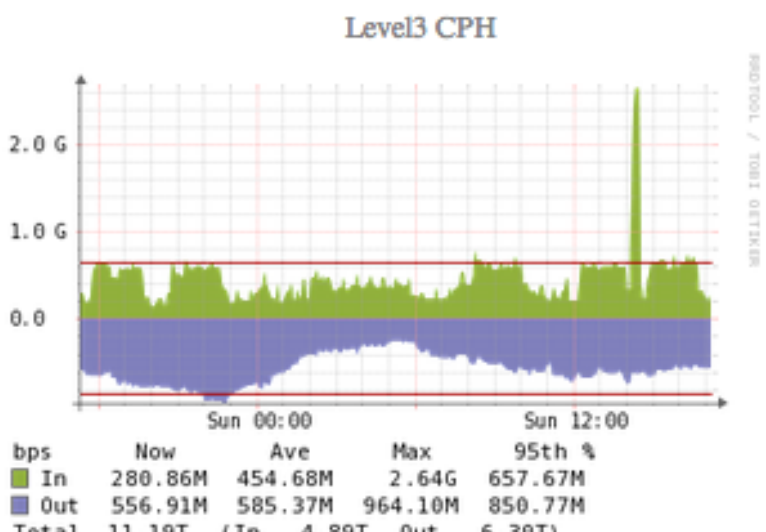
https://www.juniper.net/documentation/us/en/software/junos/flow-packet-processing/topics/topic-map/security-srx-devices-processing-overview.html

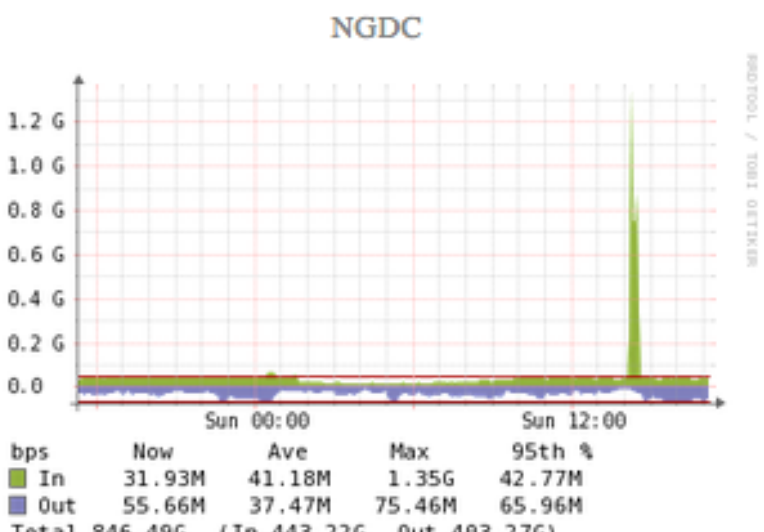
Scanning and Attacking – Pressure Points and Scope



- 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

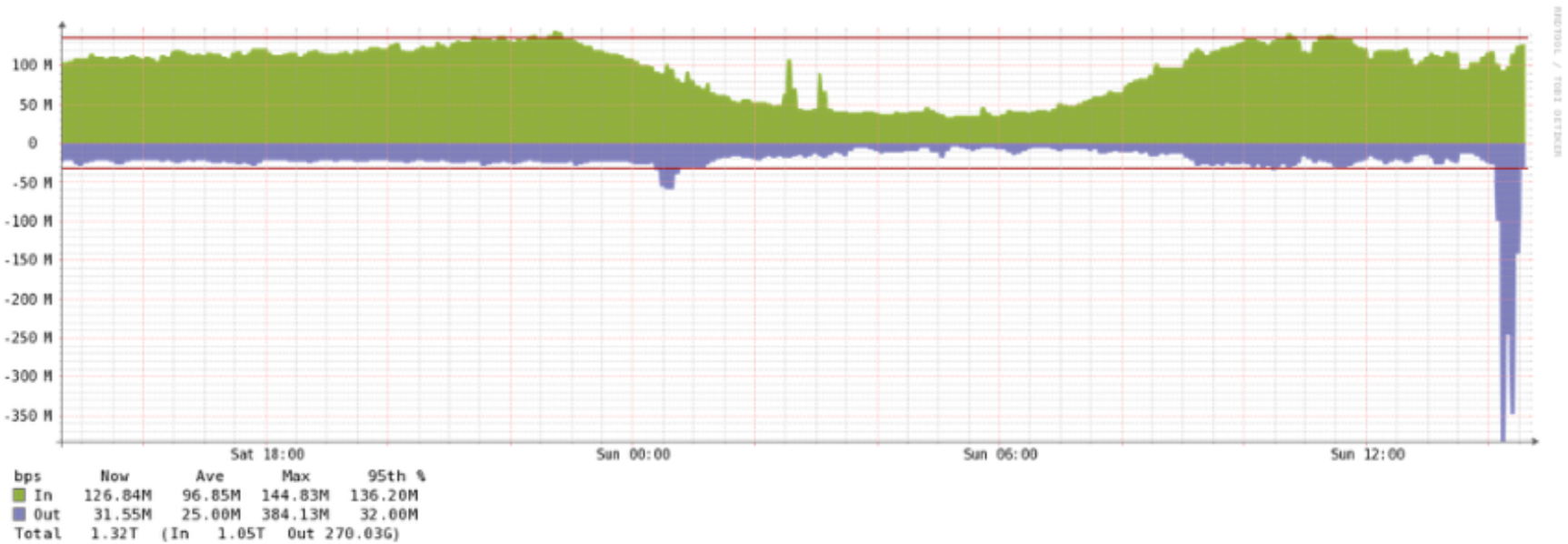
DDoS traffic before filtering





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

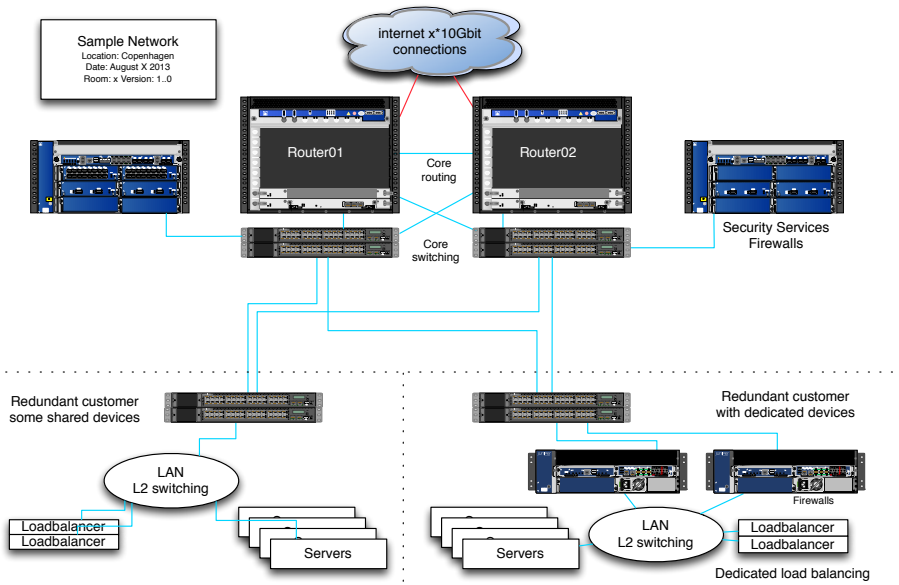
DDoS traffic after filtering



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.
- Traffic shaping is available on multiple platforms like switches and routers

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

Address family	Protocol	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

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

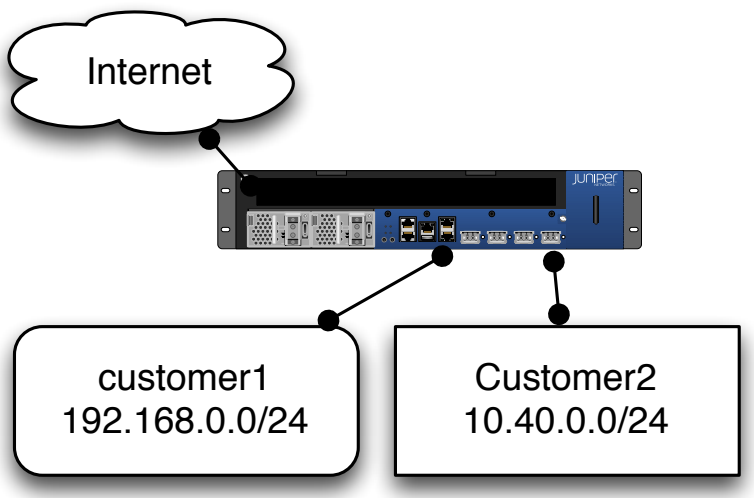
Reverse path forwarding (RPF) is a technique used in modern routers for the purposes of ensuring loop-free forwarding of multicast packets in multicast routing and to help prevent IP address spoofing in unicast routing.

Source: http://en.wikipedia.org/wiki/Reverse_path_forwarding

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.**

Strict vs loose mode RPF



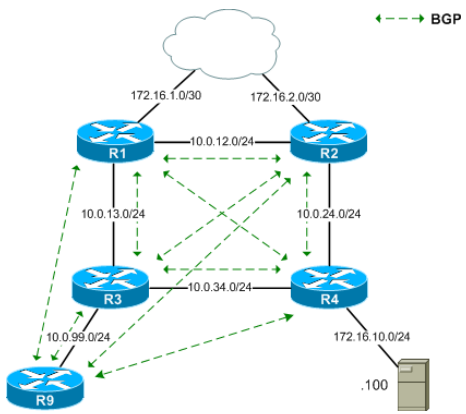

```
user@router# show interfaces
ge-0/0/0 {
        unit 2 {
                family inet {
                        rpf-check fail-filter rpf-special-case-dhcp;
                        address 192.168.0.254/24;
                }
        }
}
ge-0/0/1 {
        unit 2 {
                family inet {
  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

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

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

IPv6 is more/less secure than IPv4

There are two big misconceptions about IPv6 security:

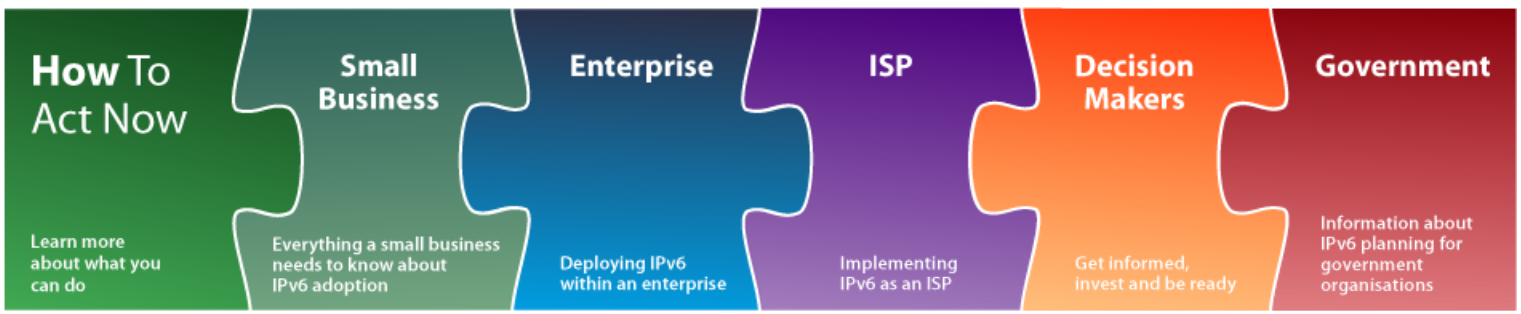
- IPv6 is more secure than IPv4
- IPv6 is less secure than IPv4

  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/

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 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

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

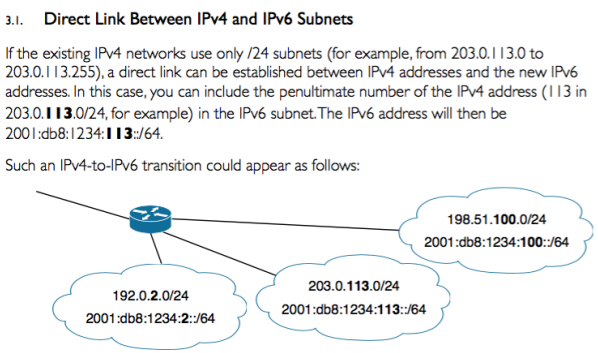
Address planning – helps security for both IPv4 and IPv6!

> 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 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
```

Providing both IPv4 + IPv6 with simple tables

The OpenBSD PF has an elegant solution for providing the same rules for both protocols, a table of addresses

```
table <webservers> { 2001:db8:1:2::80 192.0.2.80 }
...al logging is of course NOT)

pass in on $ext_if proto tcp to <webserver> port http
```

This will allow 80/tcp on both IPv4 and IPv6

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/

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

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/ 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-echange 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?

Conclusion

- Implement firewalls – take control over network packets
- Read the Fine manuals – your devices already has a lot to offer
- Make a policy for networks, make incremental changes, configure security for new parts and VLANs in the network
  Over time the older ones will be phased out, replaced or can have the same configuration applied with little trouble
- Start from the bottom and from client ports, or from server ports if you like
- Learn some Linux and use open source projects, really, will save you thosands of USD/EUR/DKK

Questions?

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

You are always welcome to send me questions later via email

Mobile: +45 2026 6000

Email: hlk@zencurity.com

Further literature

Recommended literature from my courses system security and communication and network security

- *Defensive Security Handbook: Best Practices for Securing Infrastructure*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7 284 pages
- *Forensics Discovery*, Dan Farmer, Wietse Venema 2004, Addison-Wesley 240 pages. Can be found at http://www.porcupine.org/forensics/forensic-discovery/
- *Applied Network Security Monitoring Collection, Detection, and Analysis*, 2014 Chris Sanders ISBN: 9780124172081 - shortened ANSM
- *Practical Packet Analysis - Using Wireshark to Solve Real-World Network Problems*, 3rd edition 2017, Chris Sanders ISBN: 9781593278021 - shortened PPA
- *Linux Basics for Hackers Getting Started with Networking, Scripting, and Security in Kali*. OccupyTheWeb, December 2018, 248 pp. ISBN-13: 978-1-59327-855-7 - shortened LBfH

Resources

Long list of various references follow, YMMV. I have found these useful in some way

- https://theinternetprotocolblog.wordpress.com/2020/11/28/ipv6-security-best-practices/
- https://insinuator.net/2019/02/ipv6-security-in-an-ipv4-only-environment/
  via https://mobile.twitter.com/enno_insinuator/status/1285681172719316992
- https://www.caida.org/catalog/papers/2016_dont_forget_lock/dont_forget_lock.pdf
  via https://twitter.com/Enno_Insinuator/status/1224147916022898689
- https://static.ernw.de/whitepaper/ERNW_Whitepaper68_Vulnerability_Assessment_Cisco_ACI_signed.pdf
  Classic with example of something locked down on IPv4 but not on IPv6
  I have found similar on management interfaces for a large network myself, if you came from a specific source port, you could connect to management on all core routers around the network. Router protection filter for IPv6 was not secure.

Further resources

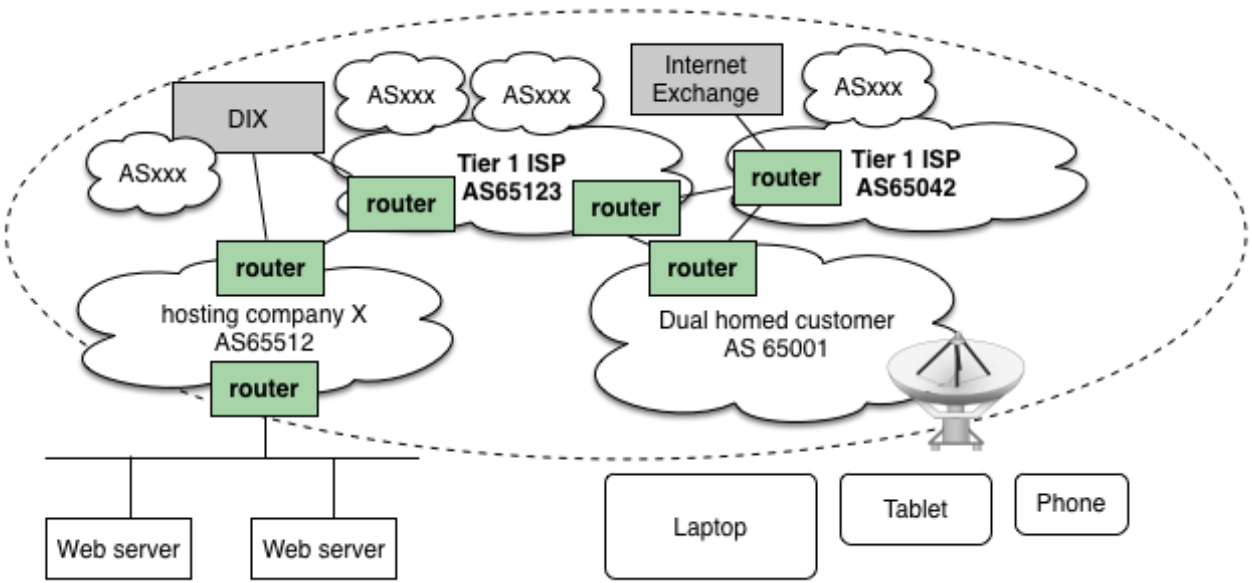
- *IPv6 and IPv4 Threat Comparison and BestPractice Evaluation (v1.0)* Sean Convery (sean@cisco.com) Darrin Miller (dmiller@cisco.com) https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.7165&rep=rep1&type=pdf 43 pages short enough, nicely structured
- https://www.cisco.com/web/SG/learning/ipv6_seminar/files/02Eric_Vyncke_Security_Best_Practices.pdf Updated? Advanced https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2020/pdf/BRKSEC-3200.pdf
- Mixed resources, maybe not useful
- https://www.varonis.com/blog/ipv6-security - Apply IPv4 best practices when applicable ... and IPv6 Security is not distinct from IPv4 security
- https://www.hpc.mil/images/hpcdocs/ipv6/infoblox-best-practices-for-ipv6-security-excerpt.pdf routing security and stuff
- https://www.nist.gov/publications/guidelines-secure-deployment-ipv6 from 2010, but maybe some good advice - and goes to show IPv6 security advise has been around for some time

Resources LIRs and others

Grateful to be part of such communities! Tried finding recent references, more can be found across their sites:

- RIPE April 2021, 191 pages!
  https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf
- ISOC 2019 https://www.internetsociety.org/deploy360/ipv6/security/faq/
- APNIC 2019 https://blog.apnic.net/2019/03/18/common-misconceptions-about-ipv6-security/
- May 2022 *Apple Platform security guide*, includes IPv6
  https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf
- *JANET IPv6 Technical Guide*, IPv4 security equivalence page 49
  https://repository.jisc.ac.uk/8349/1/janet-ipv6-technical-guide.pdf
- *Network Reconnaissance in IPv6 Networks* https://www.rfc-editor.org/rfc/rfc7707.txt
- RFC6092 2011 *Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service*
  https://datatracker.ietf.org/doc/html/rfc6092

Hosting og internet-udbydere



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

Routing and BGP Solutions

- Filtrering, 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
- 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

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

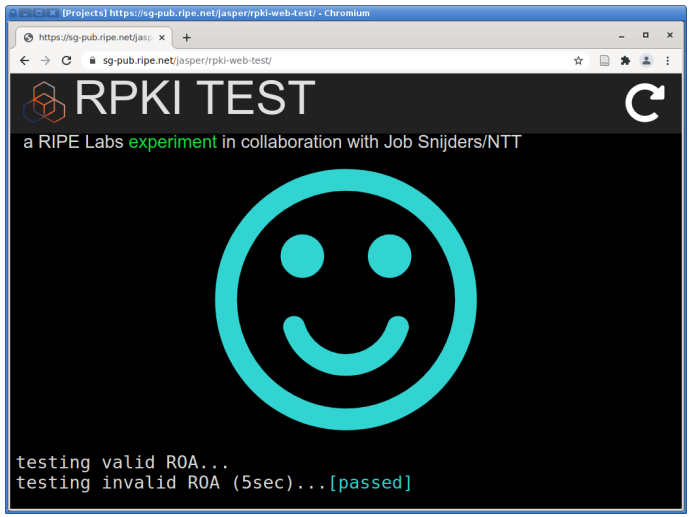
Maintain contact information in databases like Whois, PeeringDB https://www.peeringdb.com/

Advanced

4. Facilitate validation of routing information on a global scale

Use RPSL https://en.wikipedia.org/wiki/Routing_Policy_Specification_Language

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

The Zeek Network Security Monitor

Together with firewalls – 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

Together with firewalls – Suricata is a high performance Network IDS, IPS and Network Security Monitoring engine.

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

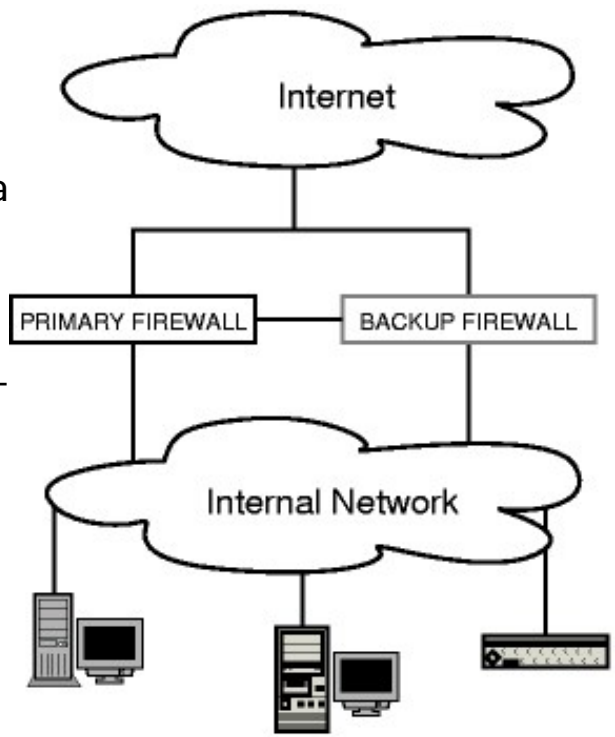
Workshop materials available:

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

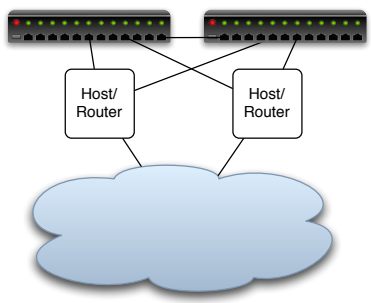
Redundante firewalls

.

- Mange producenter giver mulighed for redundante firewalls/routere
- Eksempler med fail-over protokoller som: 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, den synkroniserer alt state på PF ind og udgående, inklusiv NAT.



Redundante forbindelser IP-niveau



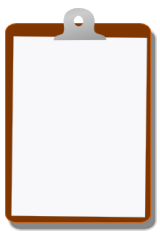
HSRP Hot Standby Router Protocol, Cisco protokol, RFC-2281

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

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools