Welcome to

Firewall Overview: The Modern Firewall Infrastructure

PROSA 2022 with apperance of the TROOPERS20 presentation I didn't give

Henrik Kramselund he/him han/ham hkj@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: hkj@zencurity.dk Mobile: +45 2026 6000

You are welcome to drop me an email

Call to Action

We need to be aware of the impact of our behaviour on others. When a person discovers they have made a mistake, they should acknowledge this and apologise.

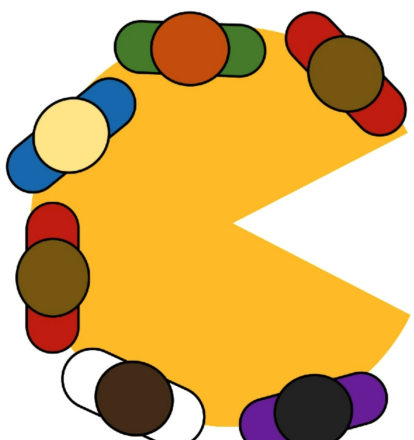
Rationale Our goals in having this Code of Conduct are:

- **To help everyone feel safe and included.** Many people will be new to our community. Some may have had negative experiences in other communities. We want to set a clear expectation that harassment and related behaviours are not tolerated here. If people do have an unpleasant experience, they will know that this is neither the norm nor acceptable to us as a community.
- **To make everyone aware of expected behaviour.** We are a diverse community; a CoC sets clear expectations in terms of how people should behave.

Source: RIPE Code of Conduct Publication date: 05 Oct 2021 https://www.ripe.net/publications/docs/ripe-766

”Pac-Man”rule

The rule is:
When standing as a group of people, always leave room for 1 person to join your group.
More memorably, stand like Pac-Man!



The new person, who has been given permission to join your group, will gather up the courage, and join you!
Another important point, the group should now readjust to leave another space for a new person.
https://ericholscher.com/blog/2017/aug/2/pacman-rule-conferences/

What we want – Learning, coexistence, to be ourselves

❤ Picture from RIPE NCC IPv6 hackathon ITU Copenhagen November 2017: ❤

https://labs.ripe.net/author/becha/results-of-the-ripe-ncc-hackathon-version-6/

Hvordan skaber vi gode miljøer

- BornHack, har skabt et virkeligt godt miljø som nydes af alle fra børn/unge til gamle, og alle mennesker
- Da vi startede i 2016 vidste vi at vi ville lave et miljø – hvor alle skulle være velkomne.
- Vi valgte at genbruge store dele af Code of Conduct fra EMF, og lavede BornHack Code of Conduct
  https://bornhack.dk/conduct/
- Det har været en stor success og I er allesammen velkomne til sommer, Aug 3. til 10. 2022
- Der er også May Contain Hackers https://mch2022.org/
  MCH2022 is the successor of a string of similar events happening every four years since 1989.
  These are GHP, HEU, HIP, HAL, WTH, HAR, OHM and SHA.

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
  Wi-Fi is also related to VLAN, but not much about wireless today
- Building a firewall infrastructure, talk about parts, show and tell
- The Modern Firewall Infrastructure
  Pruning networks without breaking them

Photo by Thomas Galler on Unsplash

Plan for today: Firewall Overview

Subjects

A) Introduction and basics

- Traffic inspection and firewalls
- Generic IP Firewalls stateless filtering vs stateful inspection
- IEEE 802.1q VLAN
- Common features in firewalls

B) Examples of firewalls, more hands-on show it

C) The Modern Firewall Infrastructure

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

Introducing firewalls

Some of these slides are part of the course:
Communication and Network Security

specifically the slideshow
3. Traffic Inspection and Firewalls

Firewalls defined, multiple definitions

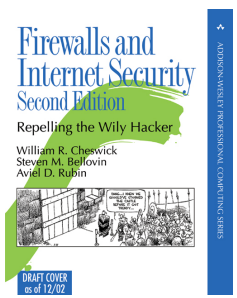
> In computing, a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.[1] A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.[2]

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

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

Source: http://www.wilyhacker.com/ Cheswick 1994 book and 2nd ed 2003

Firewalls and Internet Security, 1994



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

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

Cheswick chapter 3 *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

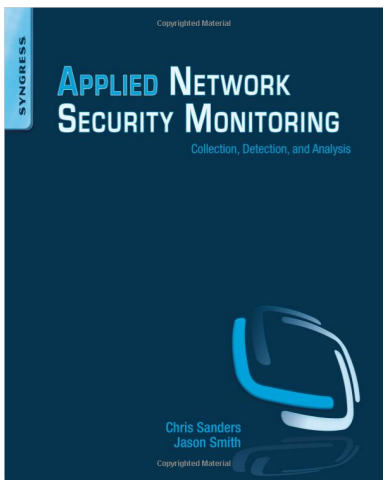
http://www.wilyhacker.com/

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

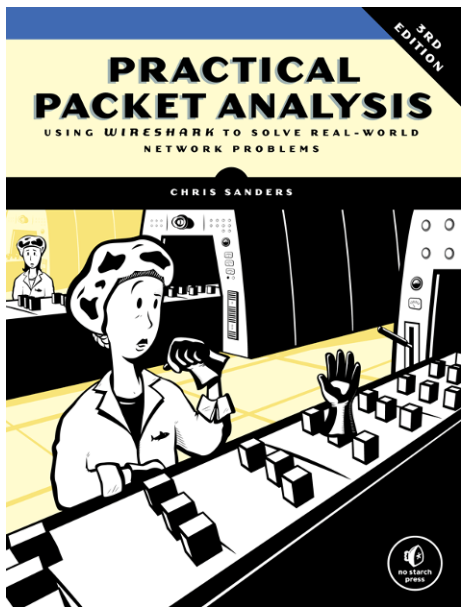
Book: Applied Network Security Monitoring (ANSM)



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

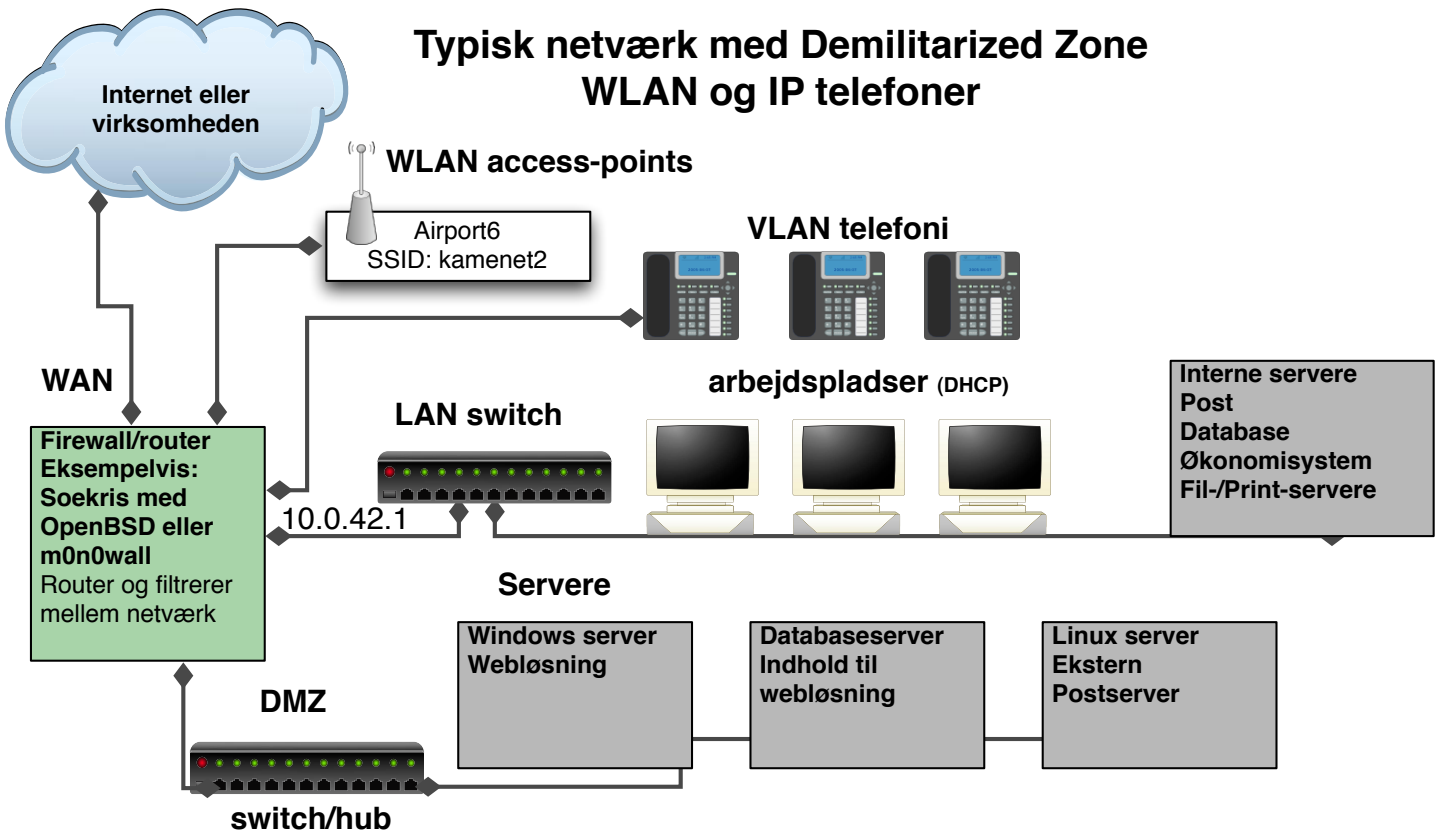
Network Protocol Knowledge Needed for Network Security

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

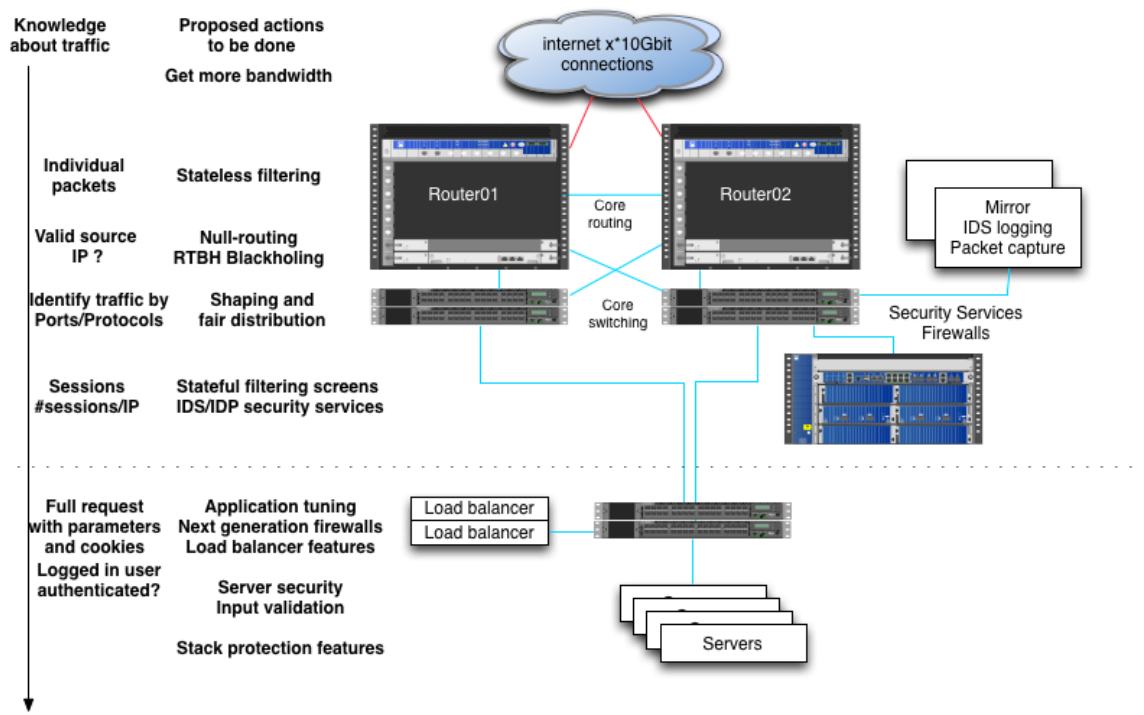


Generic IP Firewalls

En firewall er noget som blokerer traffik på Internet

En firewall er noget som tillader traffik på Internet

Firewall er ikke alene



Forsvaret er som altid - flere lag af sikkerhed!

Firewallrollen idag

Idag skal en firewall være med til at:

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

Det kræver overblik!

Modern Firewalls

Basalt set et netværksfilter - det yderste fæstningsværk

Indeholder typisk:

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

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

Sample rules from OpenBSD PF

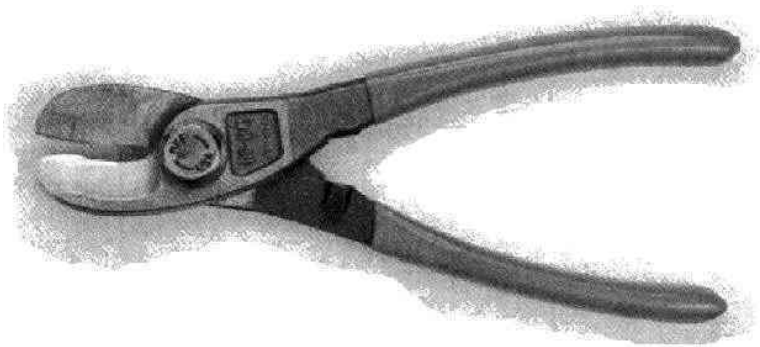
```
# hosts and networks
router="217.157.20.129"
webserver="217.157.20.131"
homenet=" 192.168.1.0/24, 1.2.3.4/24 "
wlan="10.0.42.0/24"
wireless=wi0
set skip lo0
# things not used
spoofed=" 127.0.0.0/8, 172.16.0.0/12, 10.0.0.0/16, 255.255.255.255/32 "

# 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
```

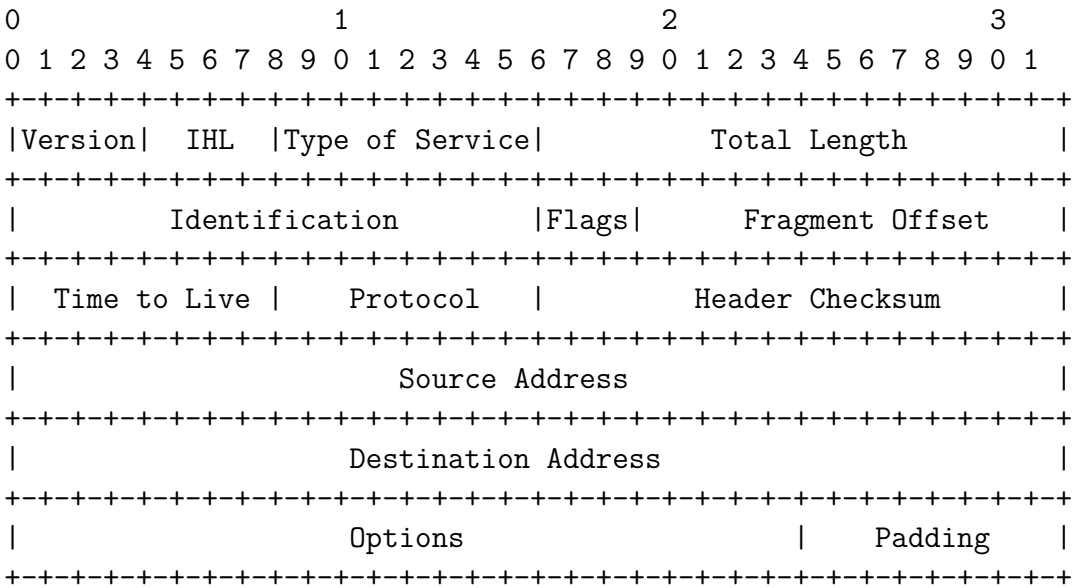
netdesign - med firewalls



- Hvor skal en firewall placeres for at gøre størst nytte?
- Hvad er forudsætningen for at en firewall virker?
  At der er konfigureret et sæt fornuftige regler!
- Hvor kommer reglerne fra? Sikkerhedspolitikken!

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

Packet filtering



Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste stateful inspection

Kommercielle firewalls

- Checkpoint Firewall-1 http://www.checkpoint.com
- Cisco ASA , Meraki, Firepower http://www.cisco.com
- Fortinet https://www.fortinet.com/
- Juniper SRX http://www.juniper.net
- Palo Alto https://www.paloaltonetworks.com/
- Big-IP F5 https://www.f5.com

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

Open source baserede firewalls

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

NB: kun eksempler og dem jeg selv har brugt

Hardware eller software

Man hører indimellem begrebet *hardware firewall*

Det er dog et faktum at en firewall består af:

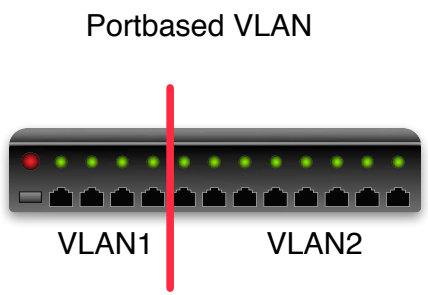
- Netværkskort - som er hardware
- Filtreringssoftware - som er *software*!

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

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

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

Together with Firewalls - VLAN Virtual LAN



Nogle switche tillader at man opdeler portene

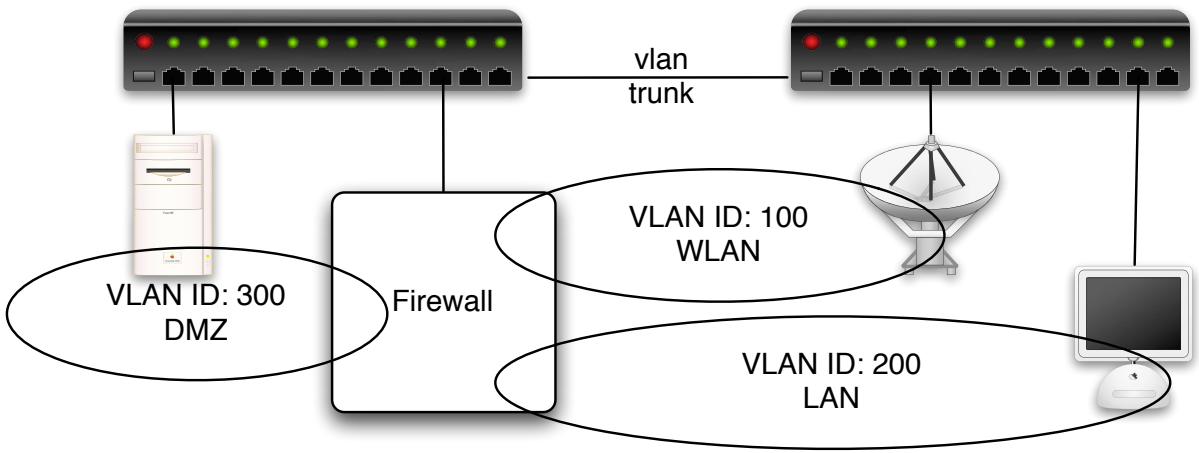
Denne opdeling kaldes VLAN og portbaseret er det mest simple

Port 1-4 er et LAN

De resterende er et andet LAN

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

IEEE 802.1q – virtual LAN



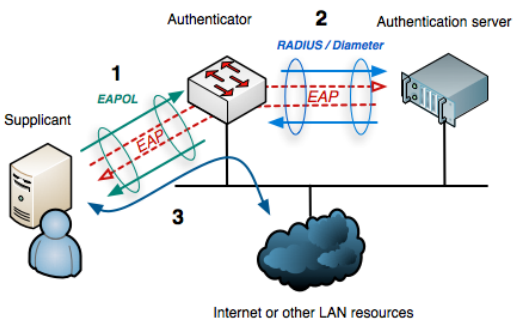
Med 802.1q tillades VLAN tagging på Ethernet niveau

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

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

Network Access Control – Connecting clients more securely

IEEE 802.1x Port Based Network Access Control



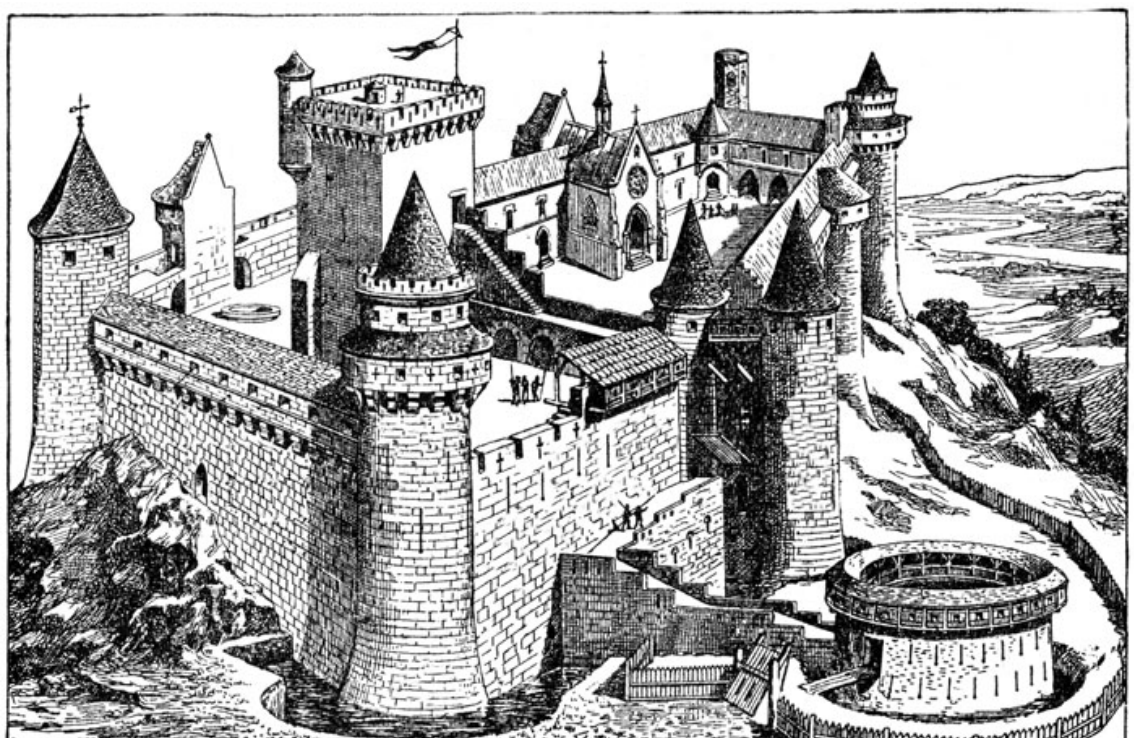
Denne protokol sikrer at man valideres før der gives adgang til porten

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

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

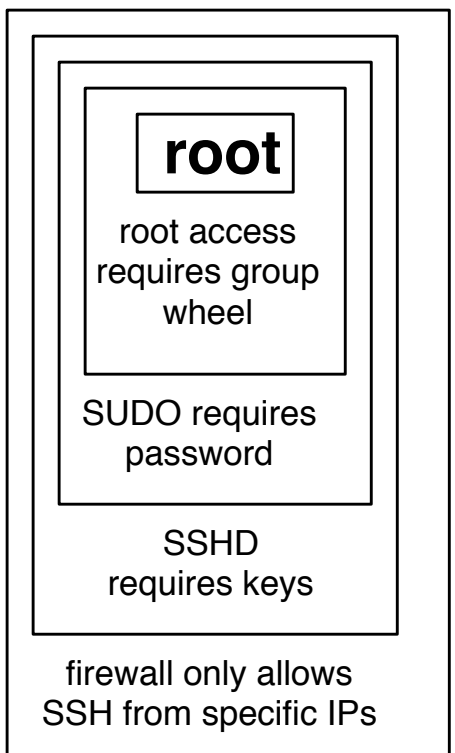
Bruges til Wi-Fi netværk

Defense in depth



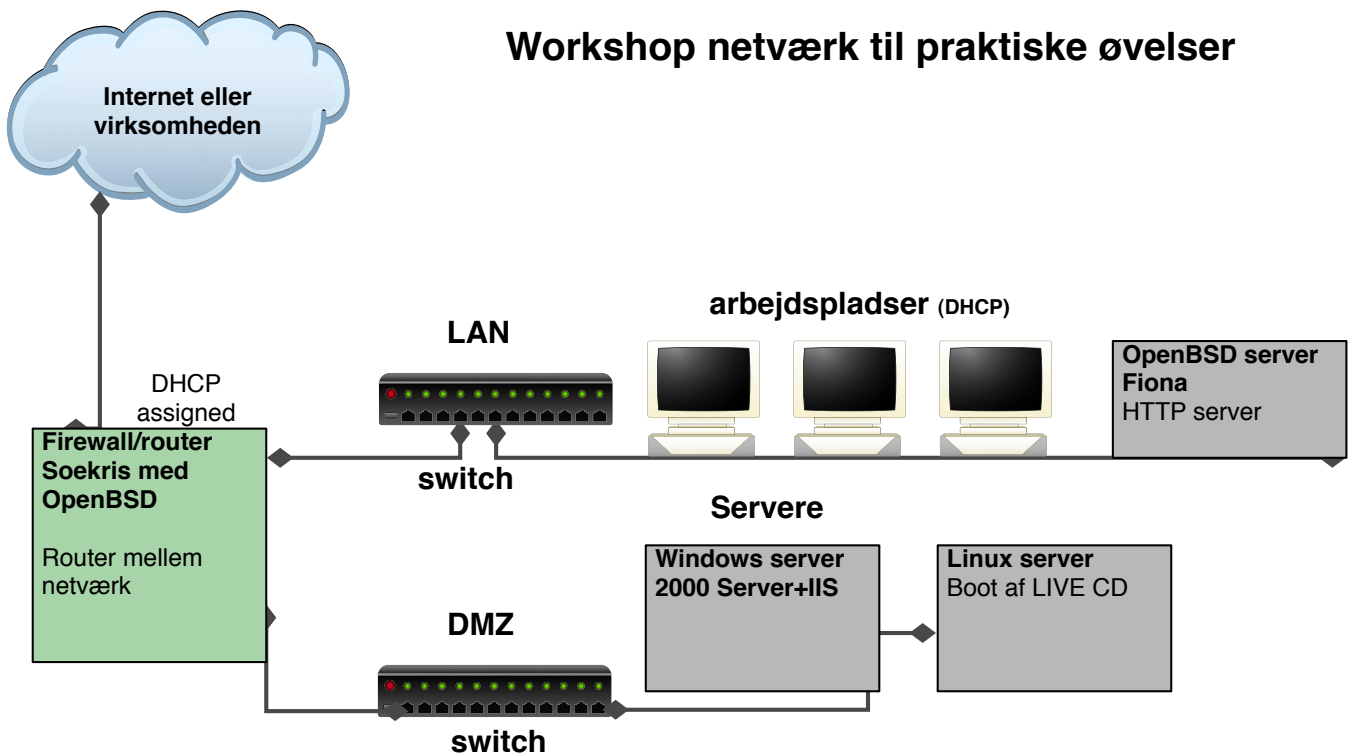
Picture originally from: http://karenswhimsy.com/public-domain-images

Defense in depth - layered security

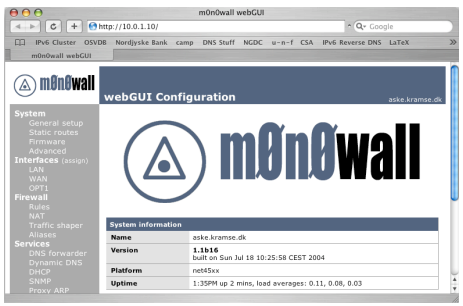


Multiple layers of security! Isolation!

Small networks



m0n0wall



Firewall bygget på FreeBSD med grafisk web interface

Idag erstattet af pfSense https://www.pfsense.org/
og OPNsense https://opnsense.org/

De nye bruges i produktion i danske firmaer

Firewall koncepter

Rækkefølgen af regler betyder noget!

- To typer af firewalls: First match - når en regel matcher, gør det som angives block/pass Last match - marker pakken hvis den matcher, til sidst afgøres block/pass

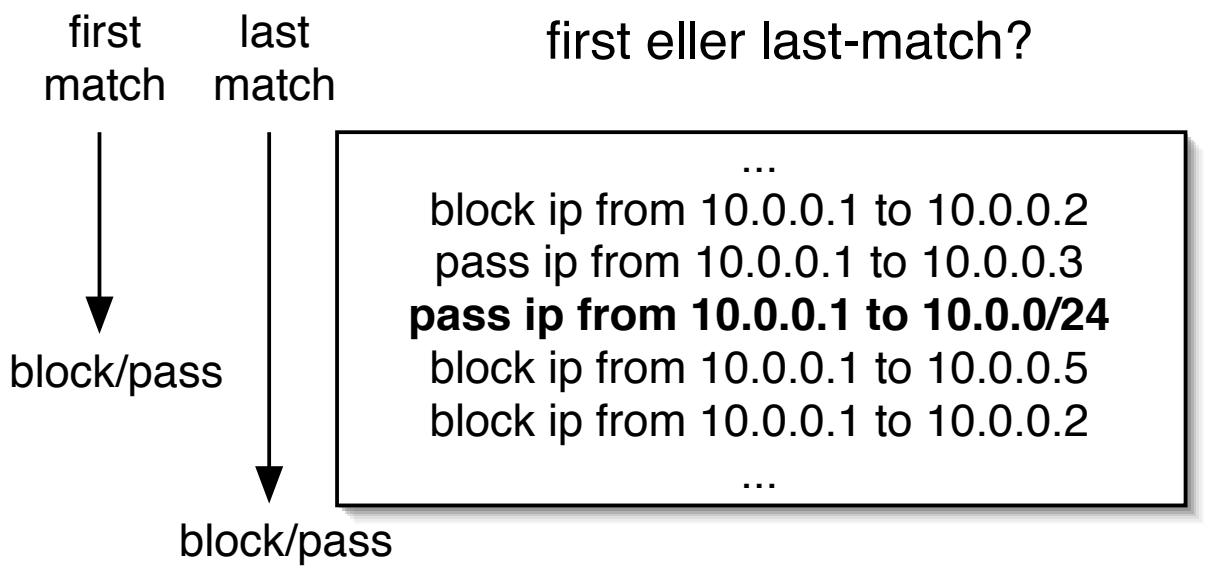
Det er ekstremt vigtigt at vide hvilken type firewall man bruger!

OpenBSD PF er last match

FreeBSD IPFW er first match

Linux iptables/netfilter er last match

First or Last match firewall?



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

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

First match - IPFW

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

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

Den sidste regel nås aldrig!

Last match - OpenBSD PF

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

block in

```
pass out keep state

pass quick on { lo $int_if }

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

Pakkerne markeres med block eller pass indtil sidste regel

nøgleordet *quick* afslutter match - god til store regelsæt

Linux iptables/netfilter eksempel

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

NB: husk at aktivere IP forwarding

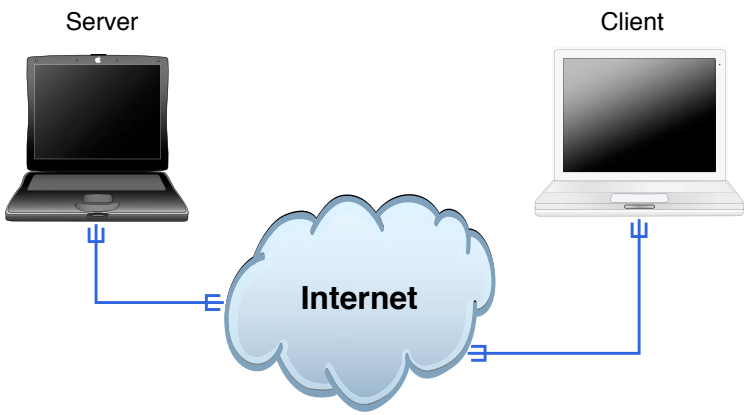
Anbefaler UFW Uncomplicated Firewall

```
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)
```

Langt nemmere at bruge

Demo: UFW – Uncomplicated Firewall



UFW – Uncomplicated Firewall

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 ICMP types:

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

Firewall konfiguration

Den bedste firewall konfiguration starter med:

- Papir og blyant
- En fornuftig adressestruktur

Brug dernæst en firewall med GUI første gang!

Husk dernæst:

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

Bloker indefra og ud

Der er porte og services som altid bør blokeres

Det kan være kendte sårbare services

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

Kendte problemer som minimum

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
- Eksemplvis 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

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

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

Executive Summary

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

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

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

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

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

Netflow

- 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

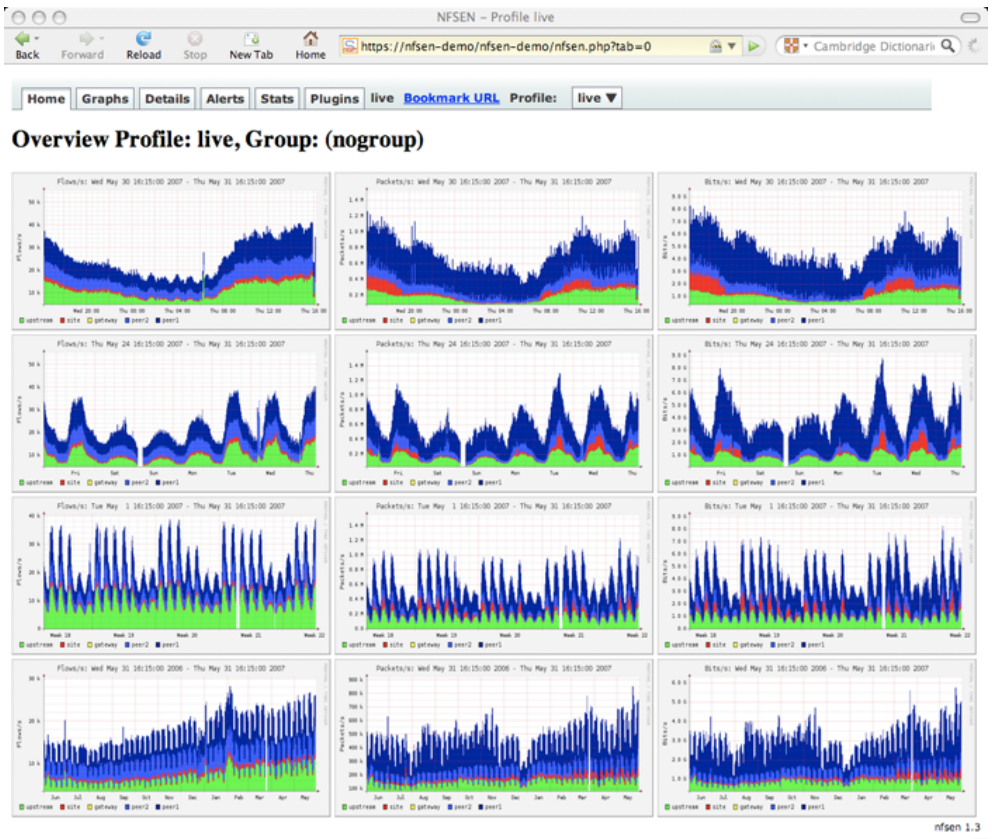
today Netflow version 9 or IPFIX

Source:

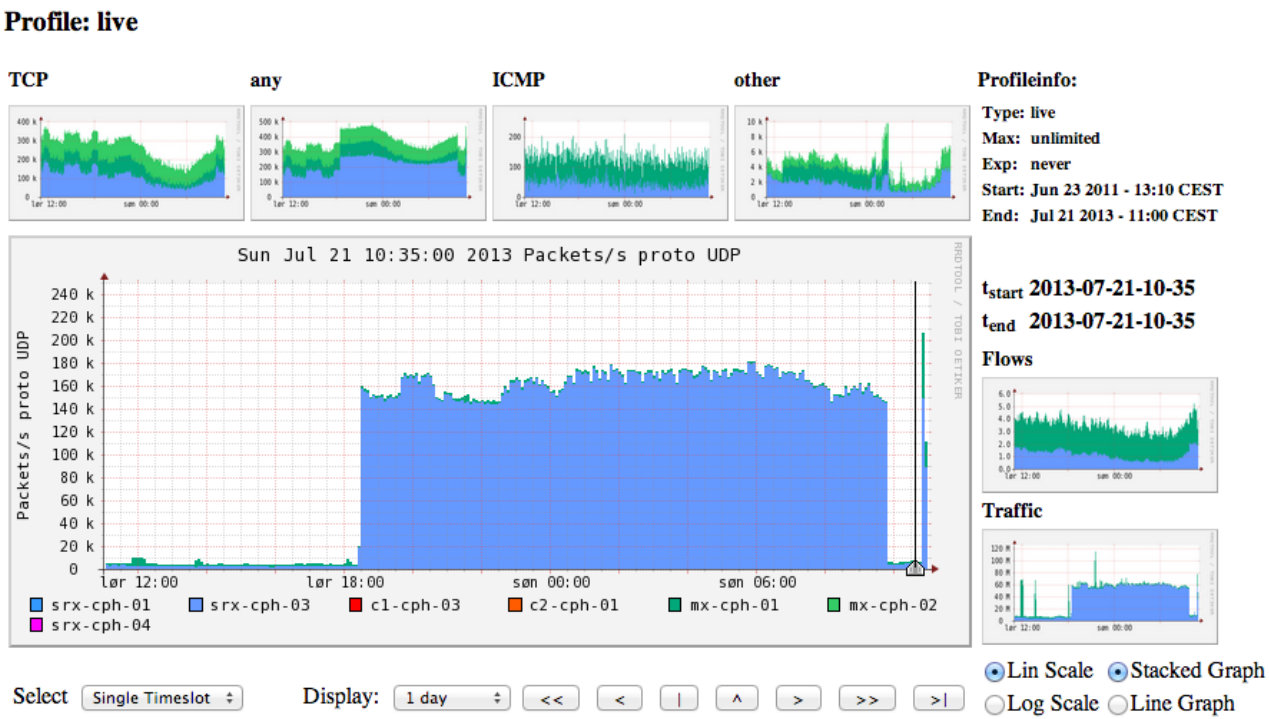
https://en.wikipedia.org/wiki/NetFlow

https://en.wikipedia.org/wiki/IP_Flow_Information_Export

Netflow using NFSen

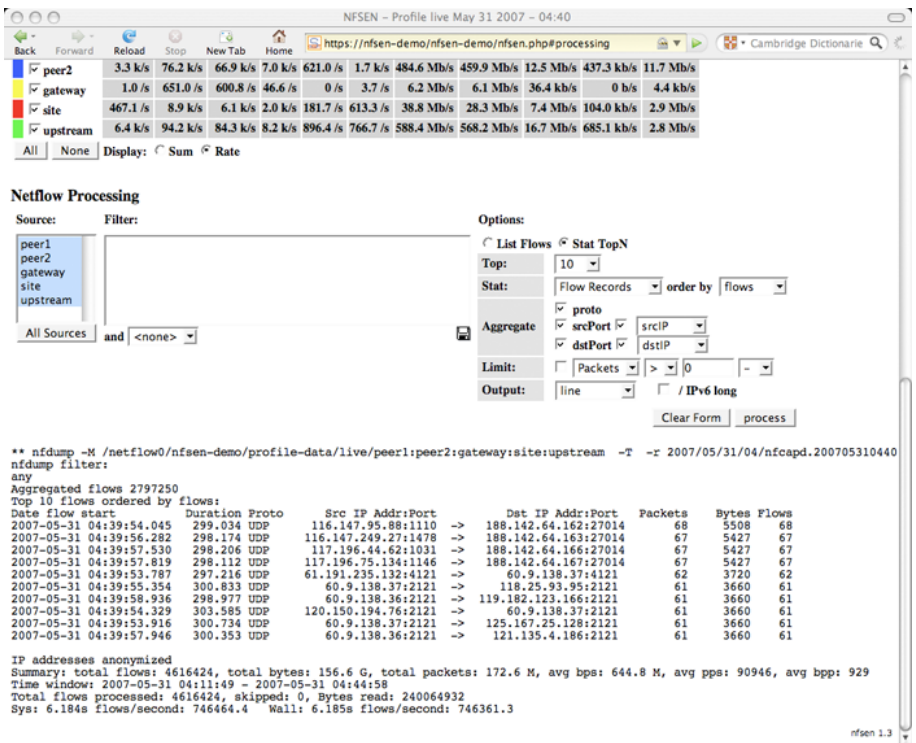


Netflow NFSen



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

How to get started

How to get started searching for security events?

Collect basic data from your devices and networks

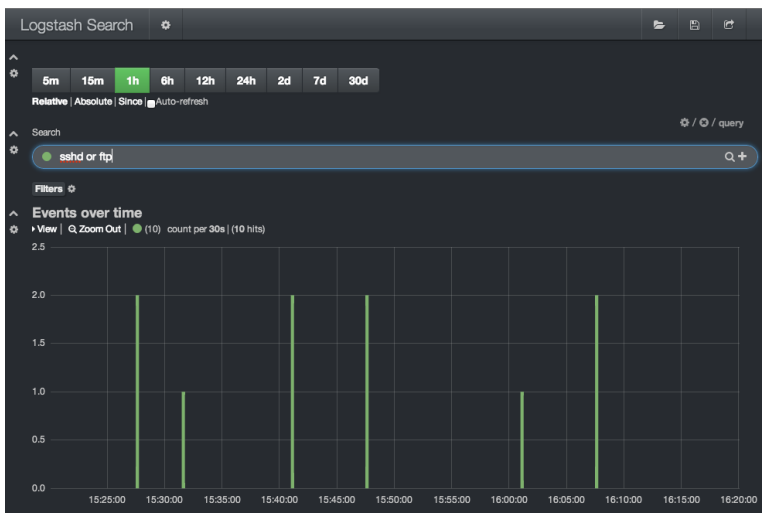
- Netflow data from routers
- Session data from firewalls
- Logging from applications: email, web, proxy systems

Centralize!

Process data

- Top 10: interesting due to high frequency, occurs often, brute-force attacks
- *ignore*
- Bottom 10: least-frequent messages are interesting

View data efficiently

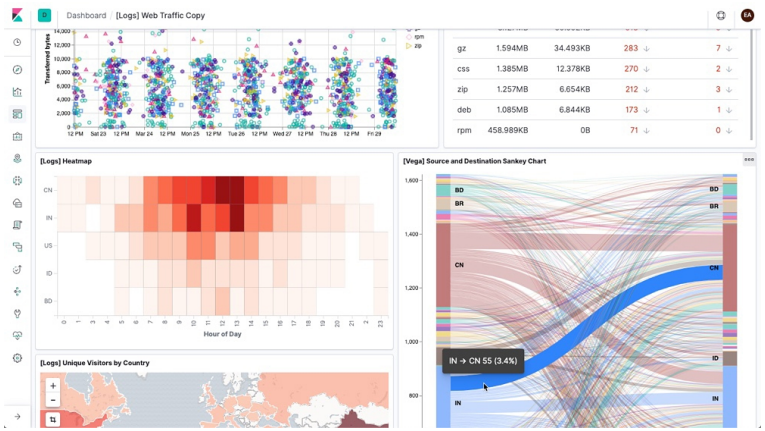


View data by digging into it easily - must be fast

Logstash and Kibana are just examples, but use indexing to make it fast!

Other popular examples include Graylog and Grafana

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

Non-programmers can create, save, and share dashboards

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

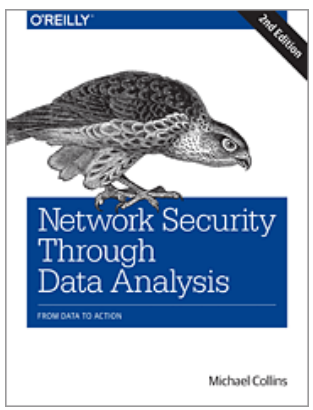
Ansible configuration management

```
- apt: name= item  state=latest
  with_items:
        - unzip
        - elasticsearch
        - logstash
        - redis-server
        - nginx
- lineinfile: "dest=/etc/elasticsearch/elasticsearch.yml state=present
  regexp='script.disable_dynamic: true' line='script.disable_dynamic: true'"
- lineinfile: "dest=/etc/elasticsearch/elasticsearch.yml state=present
  regexp='network.host: localhost' line='network.host: localhost'"
```

only requires SSH+python http://www.ansible.com

I have a set of 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

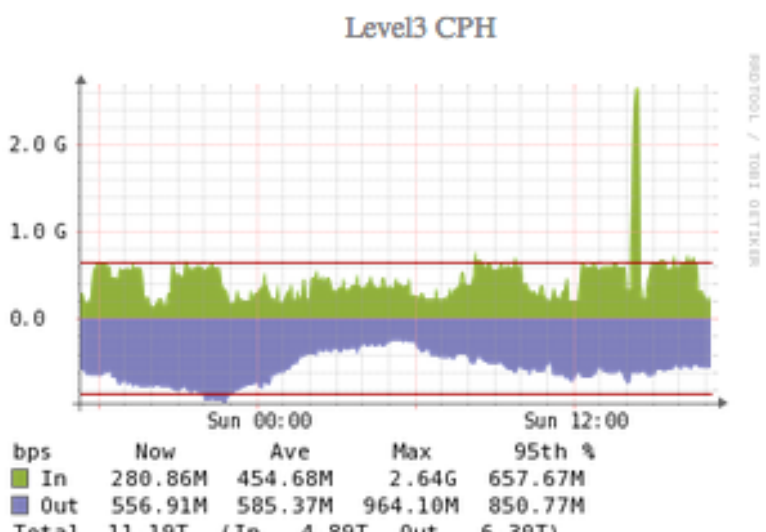


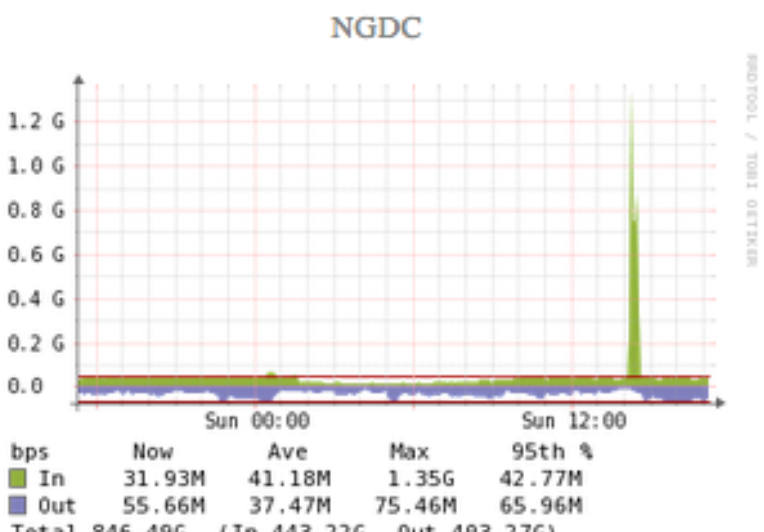
Low page count, but high value! Recommended.

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

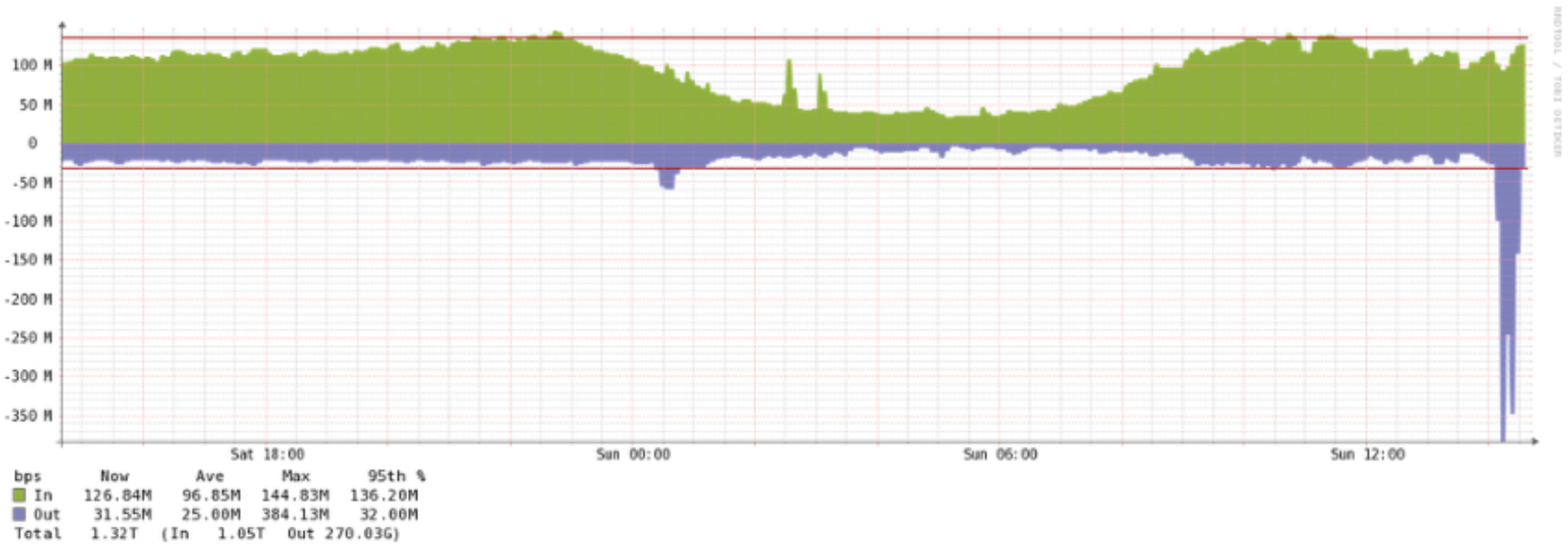
DDoS traffic before filtering





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

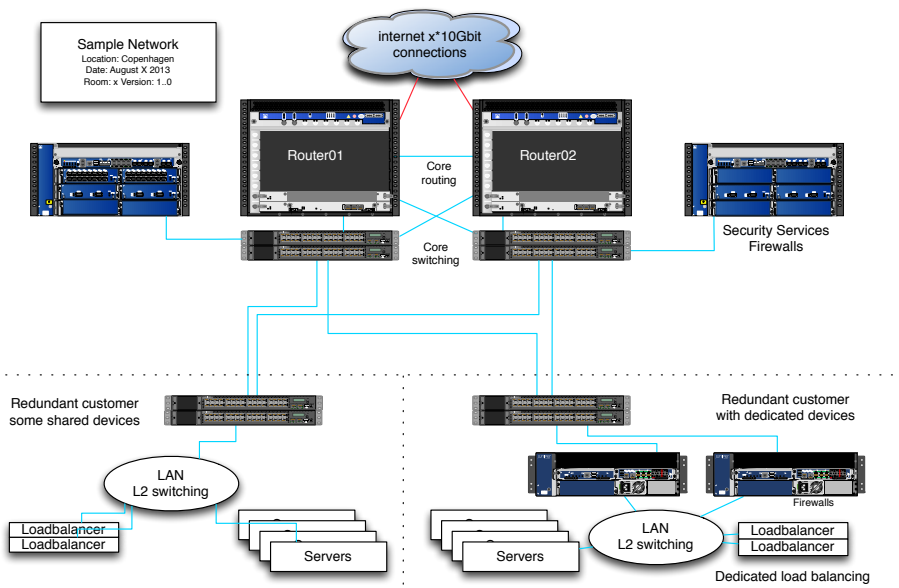
DDoS traffic after filtering



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 dummynet
- Linux Traffic Control

Det kaldes også traffic shaping

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:

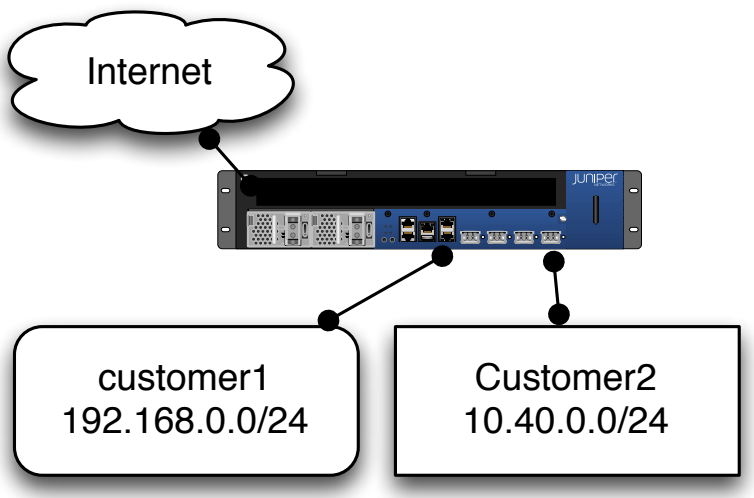
```
hlk@srx-kas-05# show security screen ids-option untrust-screen
icmp {
    ping-death;
}
ip {
    source-route-option;
    tear-drop;
}
tcp {    Note: UDP flood setting also exist
    syn-flood {
        alarm-threshold 1024;
        attack-threshold 200;
        source-threshold 1024;
        destination-threshold 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

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

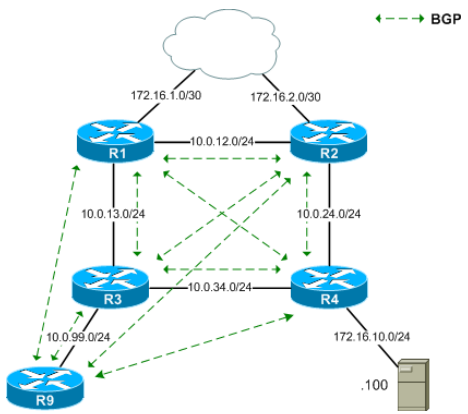
In strict mode, unicast RPF checks whether the incoming packet has a source address that matches a prefix in the routing table, **and whether the interface expects to receive a packet with this source address prefix.**

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

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

IPv6 neighbor discovery protocol (NDP)

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

ARP er væk

NDP erstatter og udvider ARP, Sammenlign `arp -an` med `ndp -an`

Til dels erstatter ICMPv6 således DHCP i IPv6, DHCPv6 findes dog

NB: bemærk at dette har stor betydning for firewallregler!

ARP vs NDP

```
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
```

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
```

IPv4 + IPv6 serviceeksempel, med tables

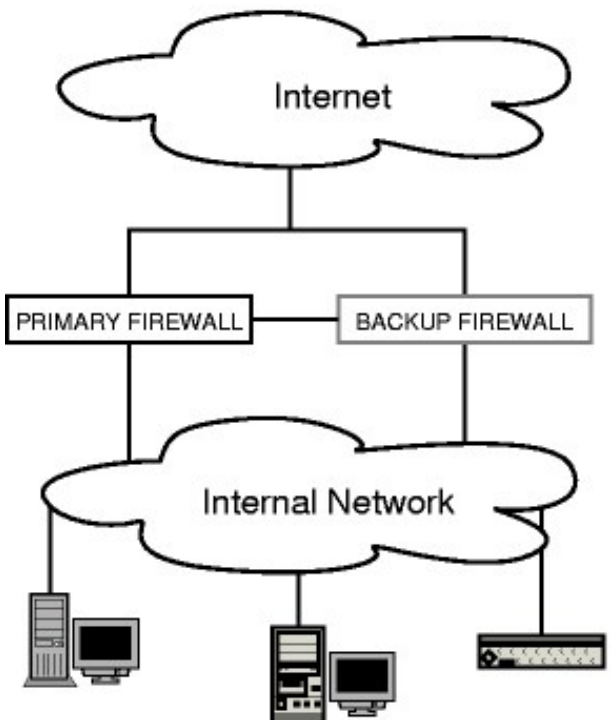
PF har tabeller og simpel regel til PF konfigurationsfilen kan give adgang til både IPv4 og IPv6:

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

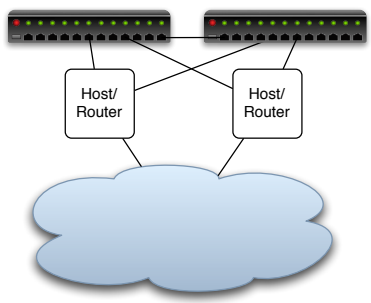
Redundante firewalls

.

- Mange producenter giver mulighed for redundante firewalls/routere
- Eksempler VRRP, CARP, HSRP Cisco, VARP Arista
- OpenBSD Common Address Redundancy Protocol CARP - både IPv4 og IPv6 overtagelse af adresse både IPv4 og IPv6
- pfsync - sender opdateringer om firewall states mellem de to systemer



Redundante forbindelser IP-niveau



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

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

CARP Common Address Redundancy Protocol, findes på OpenBSD og FreeBSD

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

Produktionsmodning af miljøer

Tænk på det miljø som servere og services skal udsættes for

Sørg for hærdning og tænk generel sikring:

- Opdateret software - ingen kendte sikkerhedshuller eller sårbarheder
- Fjern **single points of failure** - redundant strøm, ekstra enheder, to DNS servere fremfor en
- Adskilte servere - interne og eksterne til forskellige formål
  Eksempelvis den interne postserver hvor alle e-mail opbevares og en DMZ-postserver til ekstern post
- Lav filtre på netværket, eller på data - firewalls og proxy funktioner
- Begræns adgangen til at læse information
- Begræns adgangen til at skrive information - eksempelvis databaser
- Brug **least privileges** - sørg for at programmer og brugere kun har de nødvendige rettigheder til at kunne udføre opgav
- Følg med på områderne der har relevans for virksomheden og *jeres* installation

Meld jer på security mailinglister for de produkter I benytter, også open source

Change management

Er der tilstrækkeligt med fokus på software i produktion

Kan en vilkårlig server nemt reetableres

Foretages rettelser direkte på produktionssystemer

Er der fall-back plan

Burde være god systemadministrator praksis

Hardened network device configurations

Alle services skal være konfigureret korrekt:

- Administration kun fra jump host og egne administrator netværk, SSH og HTTPS
- Alle protokoller med mulighed for *secrets* bør evalueres for om det skal benyttes
- Protokoller som BGP skal benytte route import filtre
- Protokoller som OSPF bør benytte policies OG secrets
- Brug, Router protect filter således at kun relevant adgang tillades til services på udstyret
- Brug Reverse Path Forwarding uRPF / RPF

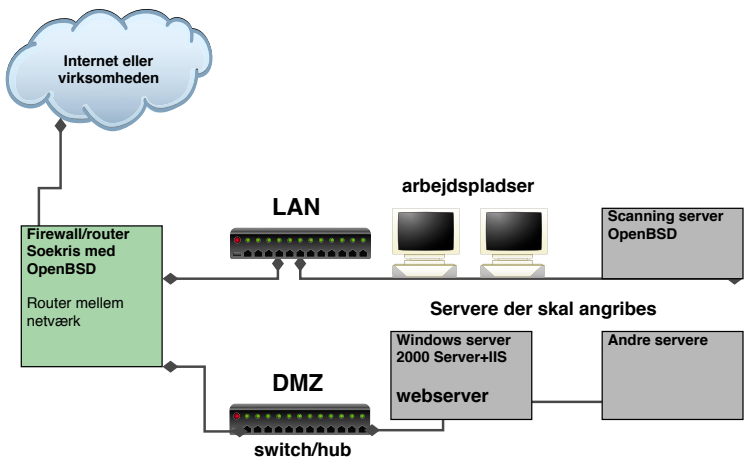
Design a robust network Isolation and segmentation

Hvad kan man gøre for at få bedre netværkssikkerhed?

- Bruge switche - der skal ARP spoofes og bedre performance
- Opdele med firewall til flere DMZ zoner for at holde udsatte servere adskilt fra hinanden, det interne netværk og Internet
- Overvåge, læse logs og reagere på hændelser

Husk du skal også kunne opdatere dine servere

Basic Network Security Pattern Isolate in VLANs



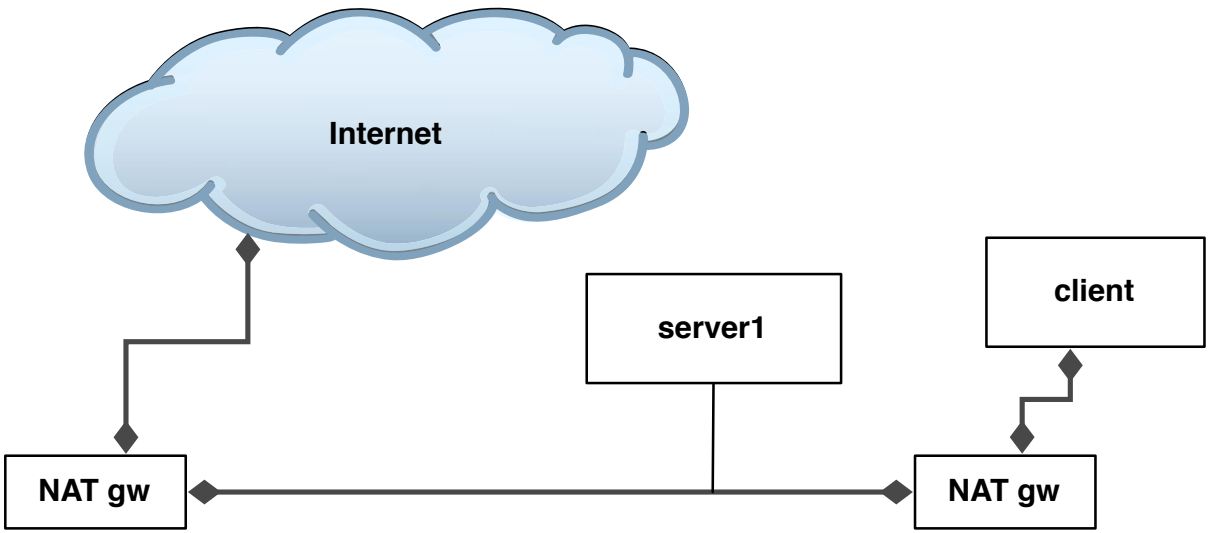
Du bør opdele dit netværk i segmenter efter trafik

Du bør altid holde interne og eksterne systemer adskilt!

Du bør isolere farlige services i jails og chroots

Brug port security til at sikre basale services DHCP, Spanning Tree osv.

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

ICMPv4 beskedtyper

Type

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

Tillad disse ICMP types:

- 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

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

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

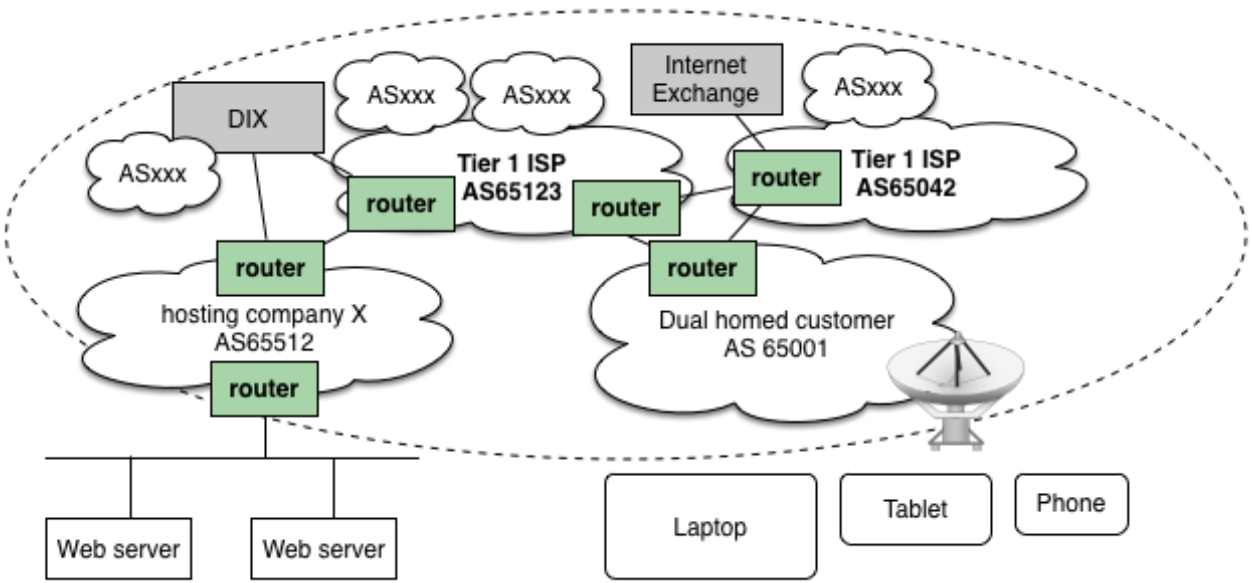
Problems related to traffic with spoofed source IP addresses

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

Hosting og internet-udbydere



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

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

At være på internet

RFC-2142 Mailbox Names for Common Services, Roles and Functions

Du BØR konfigurere dit domæne til at modtage post for følgende adresser:

- postmaster@domæne.dk
- abuse@domæne.dk
- webmaster@domæne.dk, evt. www@domæne.dk

Du gør det nemmere at rapportere problemer med dit netværk og services

Our Networks

Next we will now design a network, using our sample hardware