Welcome to

Network Automation and Monitoring Basics

How to manage networks

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

Slides are available as PDF, kramse@Github

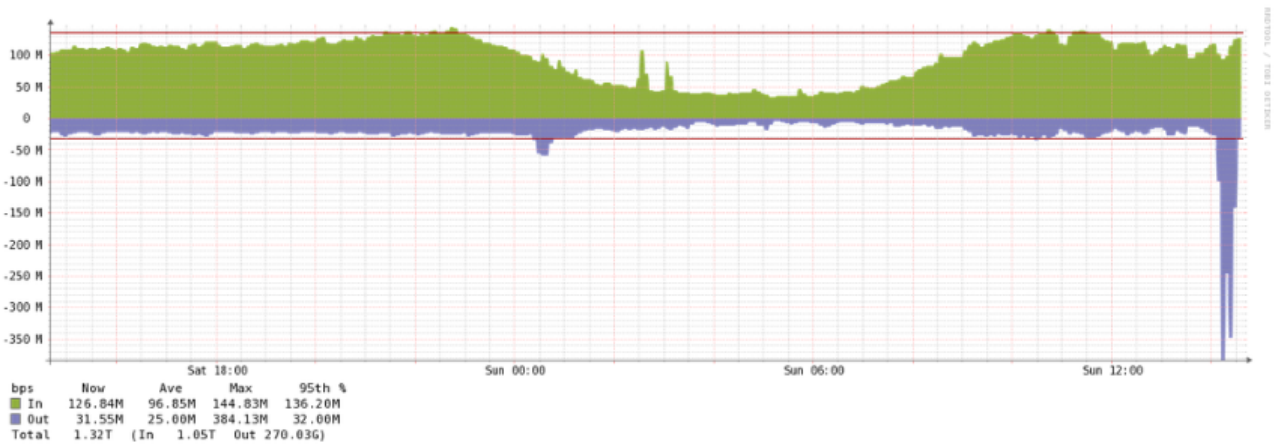
network-automation-monitoring.tex in the repo security-courses

Contact information

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

You are welcome to drop me an email

Goals: Network Automation and Monitoring Basics



- Introduce network management suitable for modern times
- Introduce resources, programs, people, authors, documents, sites that further your exploration into network management
- Starting with manual tools, move to some excellent automated ones
- I recommend open source tools, feel free to try these and then decide to pay for commercial ones, if you like

Plan for today

A starter pack for network management

- Network information TCP/IP
- The basic tools for diagnosing network problems
- Smokeping for automated monitoring
- LibreNMS - a fully featured network monitoring system
- Ansible for robust configuration of network devices, servers and other stuff
- Oxidized gathering network device configuration
- Syslog server - Logstash, Elasticsearch, Dashboards

Keywords: IP address plans, core infrastructure, SNMP, DNS, DHCPD, , Netflow, dashboards, TCP, UDP, ICMP, routing, switching, a little BGP, Ansible for network devices with Junos, OpenBSD and Linux as examples, Netflow, sFlow, monitoring systems, LibreNMS, Oxidized, Smokeping

Time schedule

- 17:00 - 18:15 Introduction and basic manual tools
- 30min break
- 18:45 - 19:30 Recommended tools, Nipap, Smokeping, LibreNMS – with breaks somewhere
- 15min break
- 19:45 - 21:00 Centralized solutions, tie together everything, questions – optional

Reklame: SIEM og Loganalyse, Diplom i IT-sikkerhed 5 ECTS
https://kompetence.kea.dk/kurser-fag/siem-og-loganalyse 9.500kr momsfri
24/11 2022, 29/11 2022, 1/12 2022, 6/12 2022, 8/12 2022, 13/12 2022, 15/12 2022 17:00-20:30, Hybrid: Vælg selv ml. online eller fremmøde, Eksamen: 5/01 2023

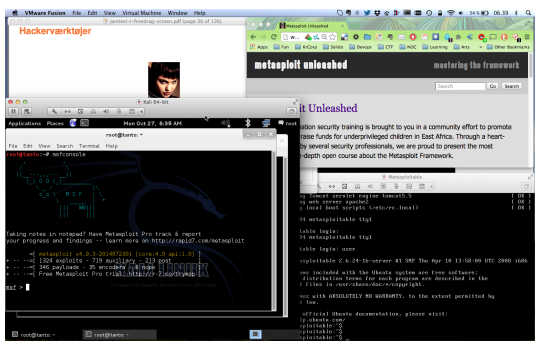
About equipment and exercises

- Bringing a laptop to my courses is not required, but welcome
- Links etc. are in the slides and open source licensed, PDFs
- Exercises booklets are available for many of my courses, see Github
  but it is expected that participants will do any exercises on their own later or at the scheduled hacker days
- The hacker days will be announced in various places
- Events like BornHack are excellent places to arrange hacker days in the network warrior village, or other places

Invite a few friends, make a hacker day and work together!

- All materials will be released as open source at:
  https://github.com/kramse/security-courses/
- Additional resources from the internet linked from lecture plans:
  https://zencurity.gitbook.io/kea-it-sikkerhed/

Hackerlab Setup



- Hardware: modern laptop CPU with virtualisation. Dont forget to enable hardware virtualisation in the BIOS
- Software Host OS: Windows, Mac, Linux
- Virtualisation software: VMware, Virtual box, HyperV pick your poison
- Hackersoftware: Kali Virtual Machine https://www.kali.org/
- If you want to do exercises and work with networks, buy a wireless USB network card and an USB Ethernet card.
- Getting an USB card allows you to use the regular one for the main OS, and insert the USB into the virtual machine

Network Management

> Network management is the process of administering and managing computer networks. Services provided by this discipline include fault analysis, performance management, provisioning of networks and maintaining the quality of service. Software that enables network administrators to perform their functions is called network management software.
> https://en.wikipedia.org/wiki/Network_management

- What are we talking about today
- Complex modern networks
- Hint: easier to consider your network a critical resource, and start monitoring now

What is the Internet

Communication between humans - currently!

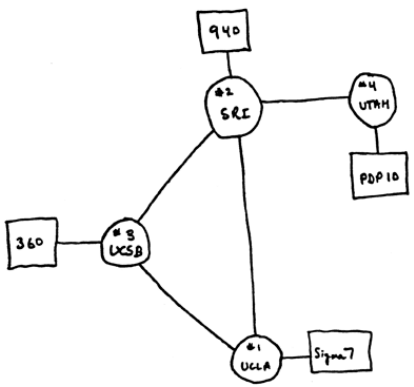
Based on TCP/IP – the internet protocol suite

- best effort
- packet switching (IPv6 calls it packets, not datagram)
- *connection-oriented* TCP
- *connection-less* UDP

RFC-1958:

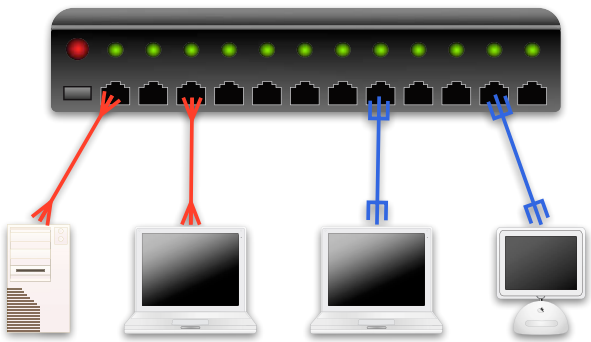
> A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning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

Internet historisk set - anno 1969



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

A switch



Today we use switches, Don't buy a hub, not even for experimenting or sniffing

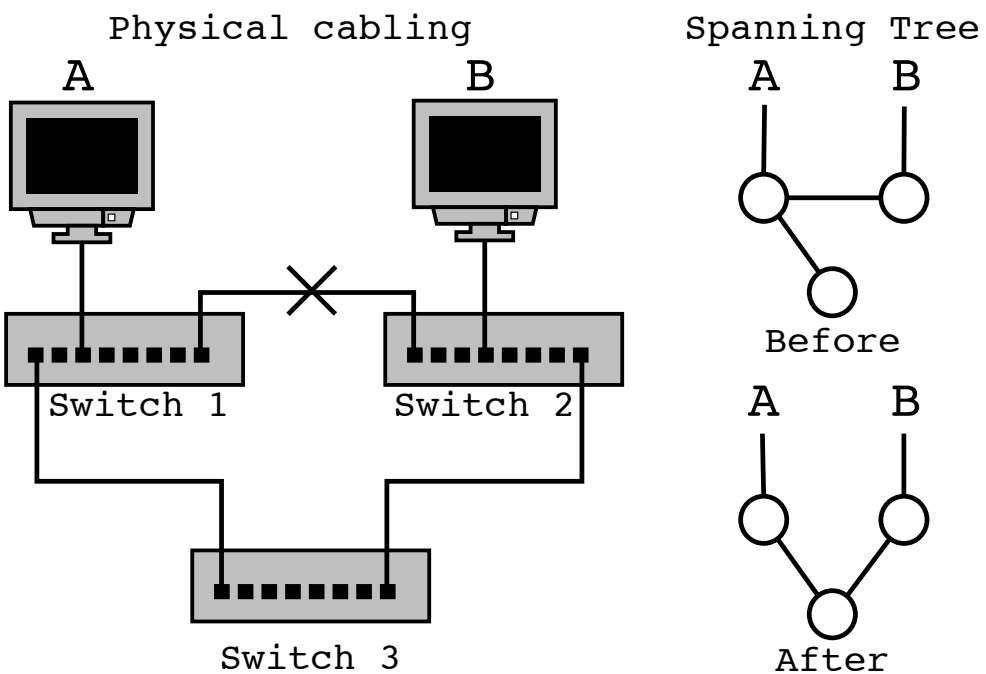
A switch can receive and send data on multiple ports at the same time

Performance only limited by the backplane and switching chips

Can also often route with the same speed

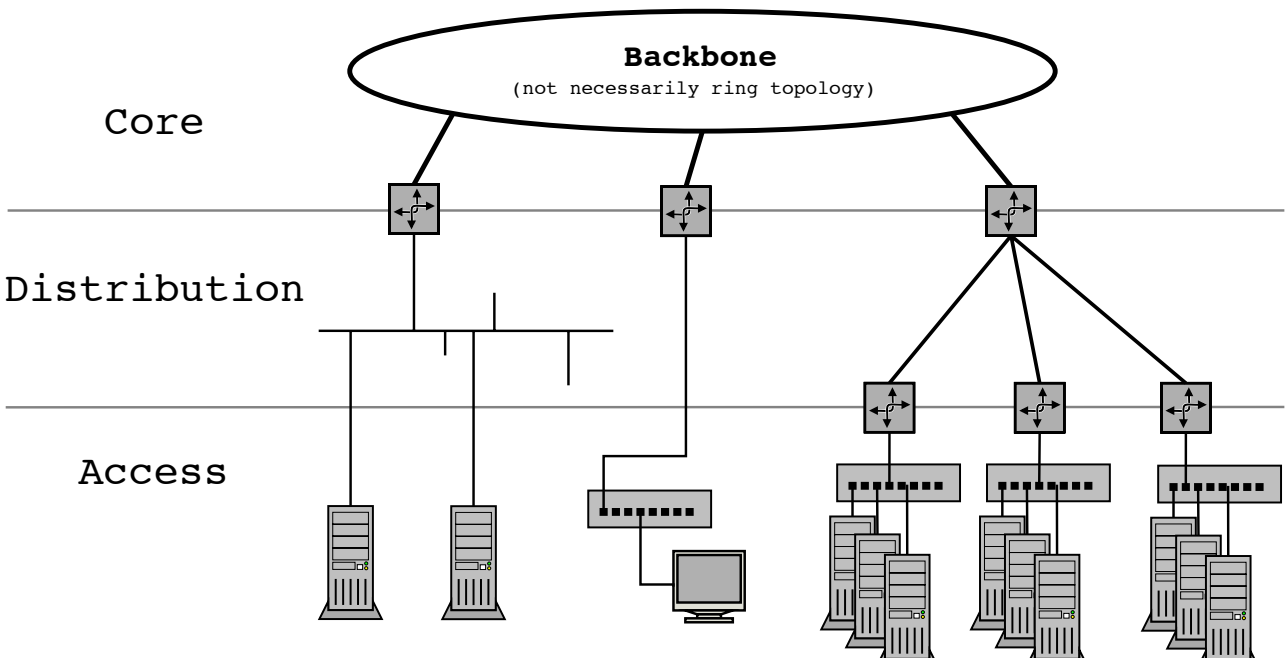
Always buy managed switches with SNMP and IEEE 802.1q, even for home

Topologier og Spanning Tree Protocol



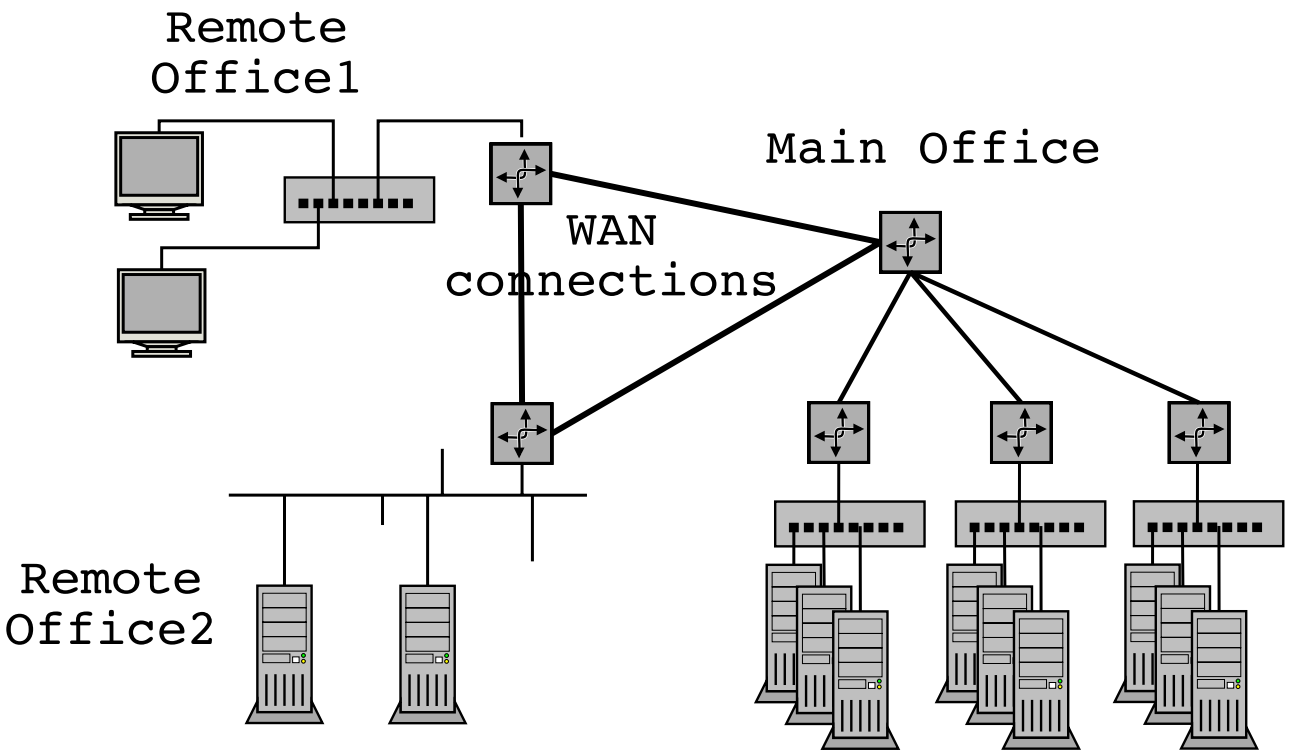
See also Radia Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*

Core, Distribution og Access net



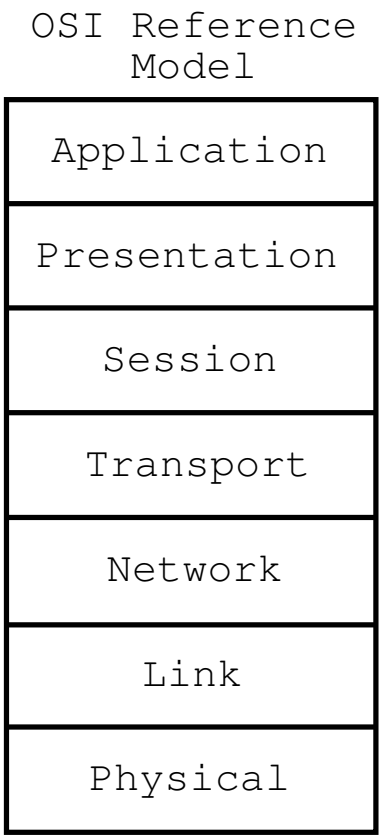
Not always this way - but often

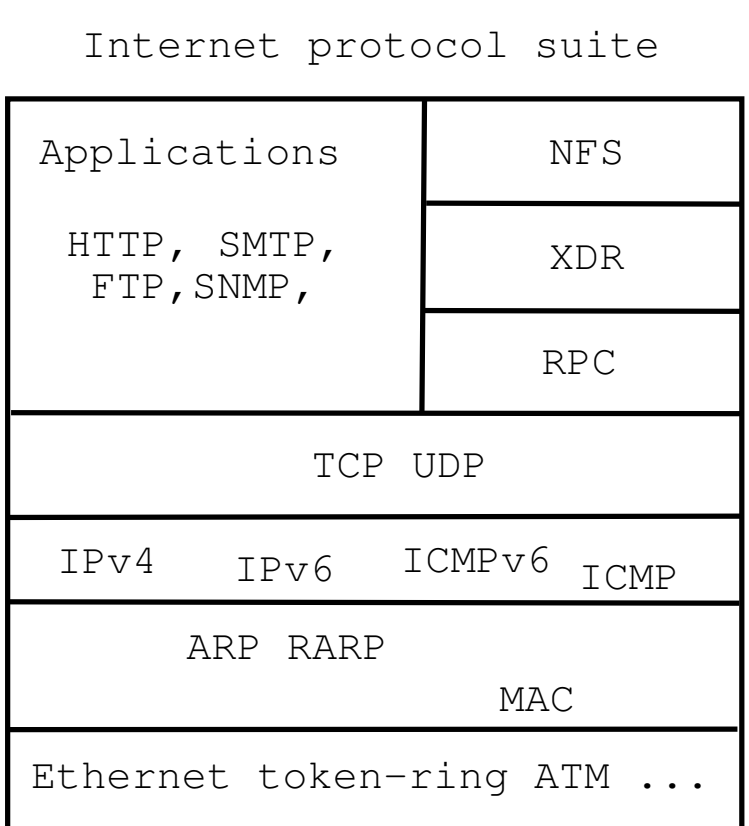
Bridges and routers



Often you inherit a mess

OSI and Internet models





MAC address

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

Network technologies use a layer 2 hardware address

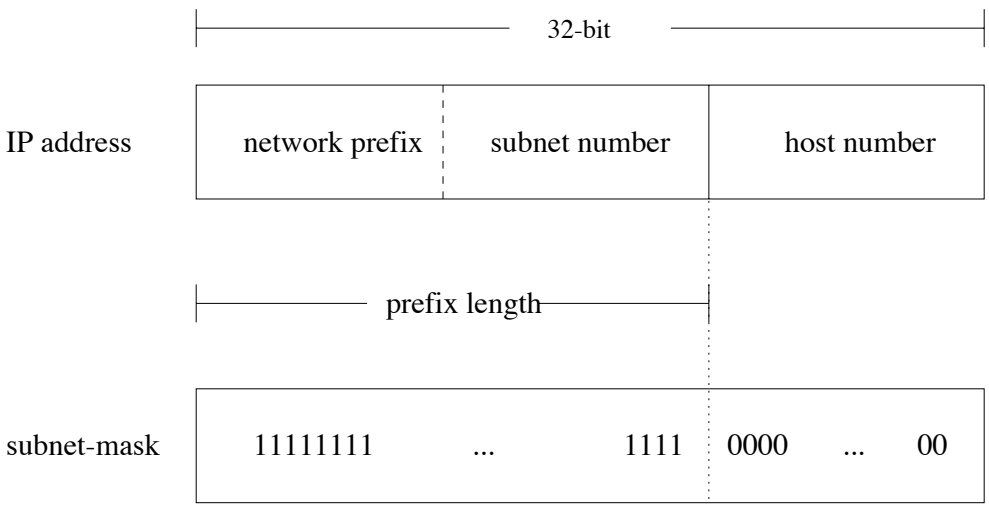
Typically using 48-bit MAC addresses known from Ethernet MAC-48/EUI-48

First half is assigned to companies – Organizationally Unique Identifier (OUI)

Using the OUI you can see which producer and roughly when a network chip was produced

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

Common Address Space



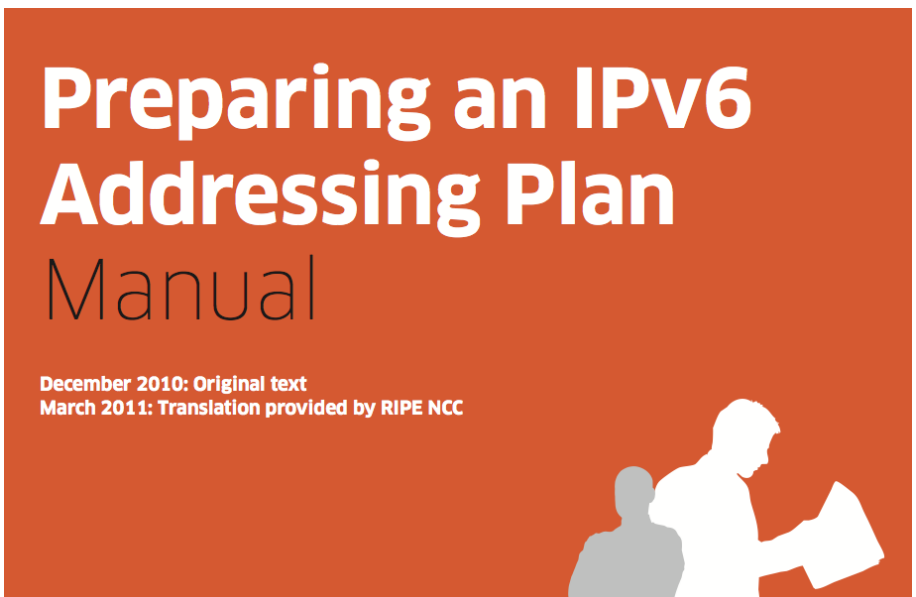
- Internet is defined by the address space, one
- Based on 32-bit addresses, example dotted decimal format 10.0.0.1

CIDR Classless Inter-Domain Routing

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

- Subnet mask originally inferred by the class
- Started to allocate multiple C-class networks - save remaining B-class
  Resulted in routing table explosion
- A subnet mask today is a row of 1-bit
- 10.0.0.0/24 means the network 10.0.0.0 with subnet mask 255.255.255.0
- Supernet, supernetting

Preparing an IPv6 Addressing Plan



http://www.ripe.net/training/material/IPv6-for-LIRs-Training-Course/IPv6_addr_plan4.pdf

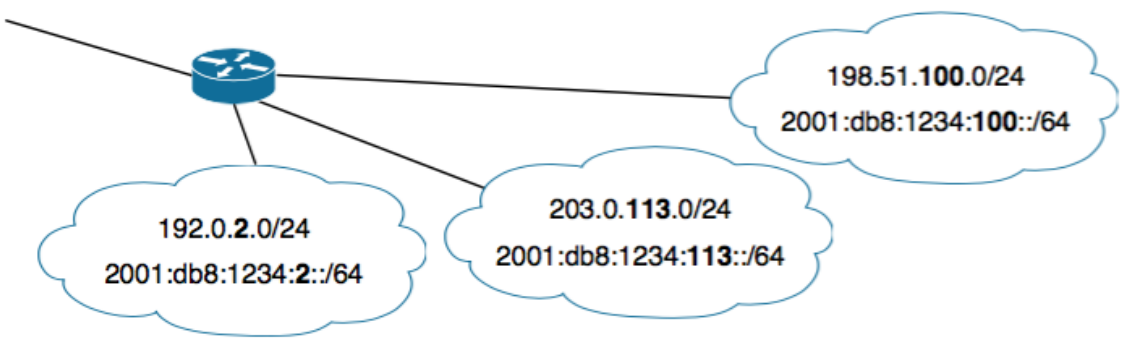
See also https://blog.apnic.net/2019/08/22/how-to-ipv6-address-planning/

Example adress plan input

3.1. Direct Link Between IPv4 and IPv6 Subnets

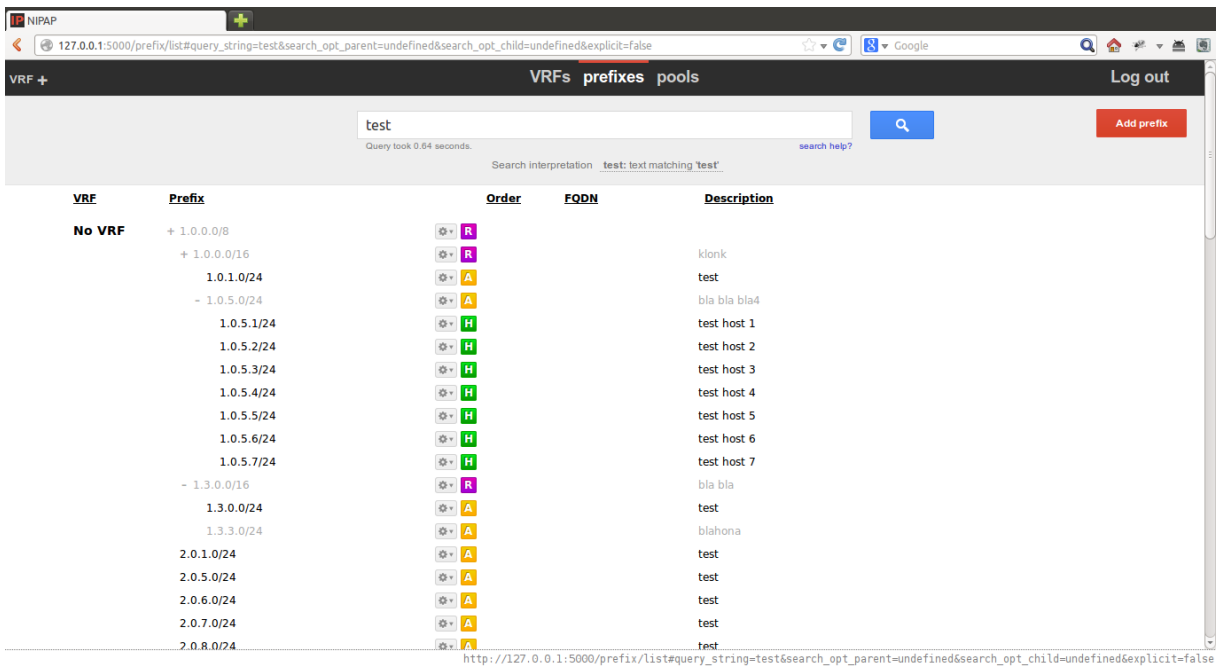
If the existing IPv4 networks use only /24 subnets (for example, from 203.0.113.0 to 203.0.113.255), a direct link can be established between IPv4 addresses and the new IPv6 addresses. In this case, you can include the penultimate number of the IPv4 address (113 in 203.0.**113**.0/24, for example) in the IPv6 subnet. The IPv6 address will then be 2001:db8:1234:**113**::/64.

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



Easy and coupled with VLAN IDs it will work ☺

IP Address Management IPAM



- Recommend Nipap http://spritelink.github.io/NIPAP/

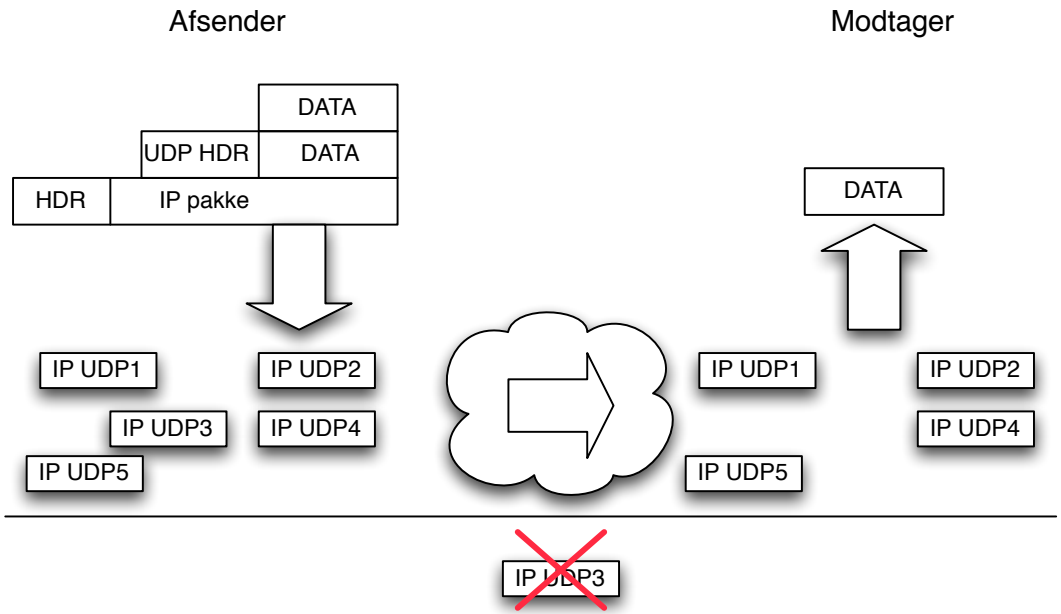
NIPAP example, adding firewall cluster interfaces - multiple VLANs

```
#! /bin/sh
# Add sites to NIPAP,  dont waste time doing it with mouse
addsite ()
{  SITE=$1
   SITEID=$2
  for VLAN in 100 200 300 400 500
  do
    nipap address add family ipv4 prefix 172.$SITEID.$VLAN.1 node $SITE-fw.example.net
    nipap address add family ipv4 prefix 172.$SITEID.$VLAN.2 node $SITE-fw-01.example.net
    nipap address add family ipv4 prefix 172.$SITEID.$VLAN.3 node $SITE-fw-02.example.net
  done
}

addsite dk-odense 1231
addsite dk-svendborg 1232
```

- Automating saves time, and less errors from manual input!

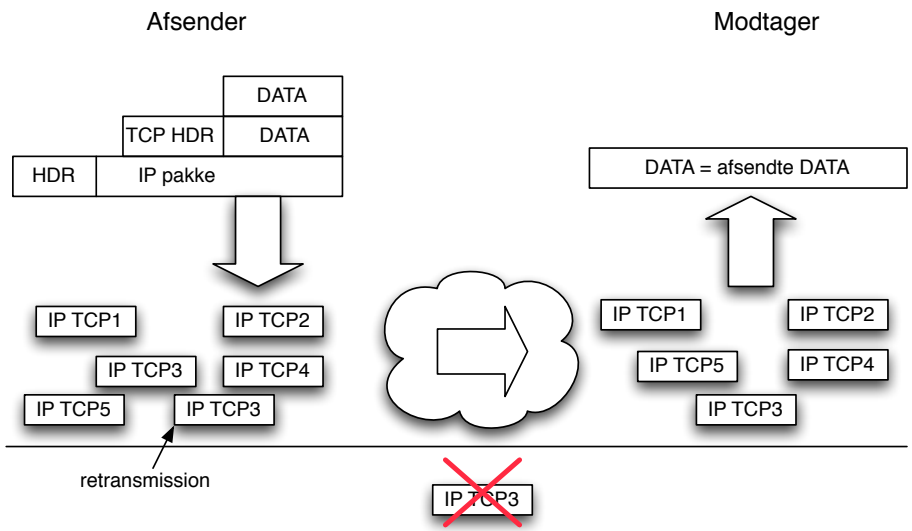
UDP User Datagram Protocol



Connection-less RFC-768, *connection-less*

Used for Domain Name Service (DNS)

TCP Transmission Control Protocol



Connection oriented RFC-791 September 1981, *connection-oriented*

Either data delivered in correct order, no data missing, checksum or an error is reported

Used for HTTP and others

Well-known port numbers

IANA maintains a list of magical numbers in TCP/IP

Lists of protocl numbers, port numers etc.

A few notable examples:

- Port 25/tcp Simple Mail Transfer Protocol (SMTP)
- Port 53/udp and 53/tcp Domain Name System (DNS)
- Port 80/tcp Hyper Text Transfer Protocol (HTTP)
- Port 443/tcp HTTP over TLS/SSL (HTTPS)

Source: http://www.iana.org

ICMP Internet Control Message Protocol

Control protocol, error messages

Common messages

- ICMP ECHO, anyone there?
- Host unreachable
- Port unreachable

signaling

Defined in RFC-792

Don't block all ICMP – wrong!

ICMP messages to allow – Similar for IPv6

Type

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

If you remove all ICMP you will have time outs instead

Allow these ICMP types:

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

IPv6 neighbor discovery protocol (NDP)

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

Address Resolution Protocol (ARP) is gone from IPv6

NDP replaces and expand, command to use `arp -an` replaced by `ndp -an`

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
```

Basic test tools TCP - Ping and Traceroute

We should all know

- `ping` – like sending a radar ping, anything there
- `traceroute` (windows tracert) – find the route packets traverse

and add these!

- `Wireshark` – sniffing and disecting traffic
- `Nmap` and `Nping` – port scan and advanced ping program with TCP/UDP!

Ping

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

ICMP – Internet Control Message Protocol

ECHO – only ICMP message that generates another

ICMP ECHO request sent, and ICMP ECHO reply expected

Same with IPv6, `ping6`

Fun fact: A problem with say TCP or UDP can be reported with ICMP, but ICMP ECHO is the only ICMP message that will result in an ICMP response. No black storms started with ICMP.

traceroute

```
$ traceroute www.kramse.dk
traceroute to www.kramse.dk (185.129.60.130), 30 hops max, 60 byte packets
 1  10.0.42.1 (10.0.42.1)  0.365 ms  0.277 ms  0.239 ms
 2  79.142.xxx.xxx (79.142.xxx.xxx)  5.174 ms  4.979 ms  5.113 ms
 3  bgp2-dix.prod.bolignet.dk (79.142.224.2)  5.538 ms  5.057 ms  5.483 ms
 4  217.74.215.57 (217.74.215.57)  5.990 ms  5.962 ms  5.932 ms
...
 8  185.150.199.178 (185.150.199.178)  7.684 ms  7.647 ms  4.627 ms
 9  * * * // firewall here!
```

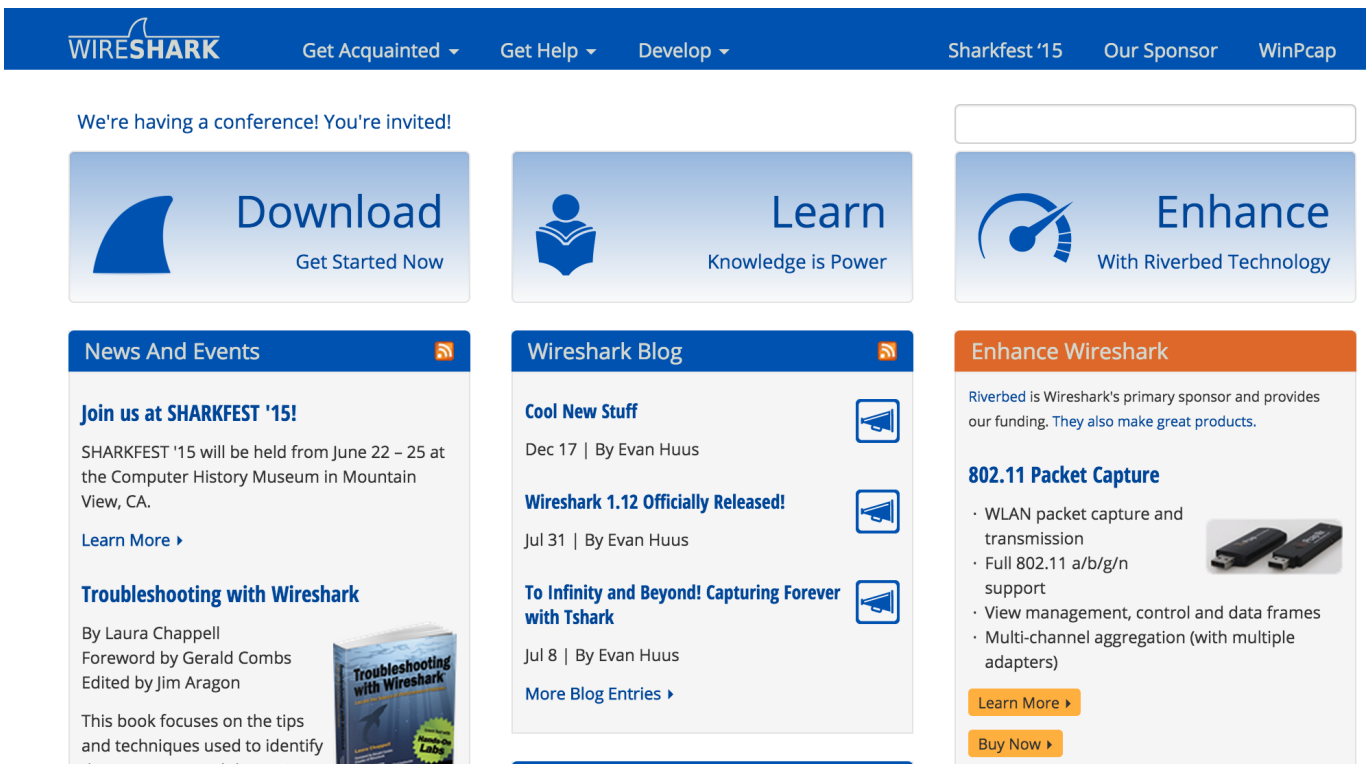
Works using the Time to live (TTL) counter

Sending with $TTL = 1$ returns ICMP from first host/router

Default sends UDP on Unix, and ICMP on Windows

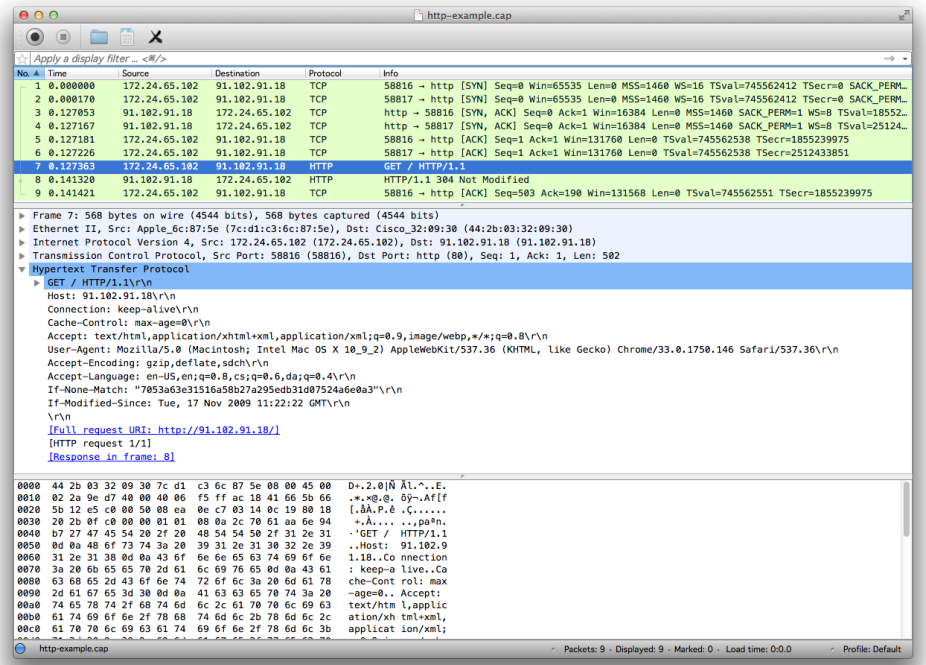
Kali has programs that can emulate, or send using any protocol

Wireshark - graphical network sniffer



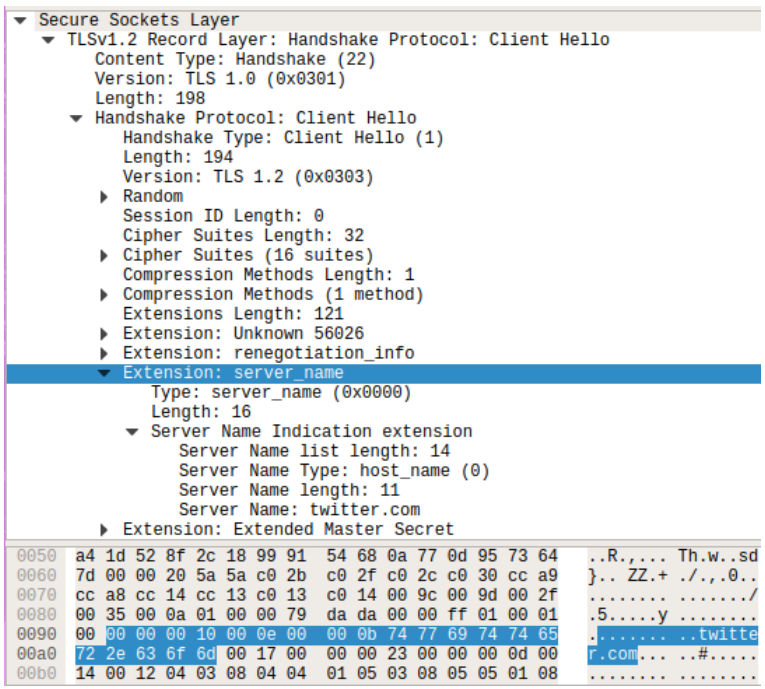
http://www.wireshark.org

Using Wireshark



Capture - Options, select a network interface

Detailed view of network traffic with Wireshark



Notice also the filtering possibilities, capture and view

Wireshark is an advanced manual tool, try right-clicking different places

Remote network debugging

- TShark and Tcpdump, I often use: `tcpdump -nei eth0`
  `tshark -z expert -r download-slow.pcapng`
- Remote packet dumps, `tcpdump -i eth0 -w packets.pcap`
- Story: tcpdump was originally written in 1988 by Van Jacobson, Sally Floyd, Vern Paxson and Steven McCanne who were, at the time, working in the Lawrence Berkeley Laboratory Network Research Group
  https://en.wikipedia.org/wiki/Tcpdump

Great network security comes from knowing networks!

Note About Hardware IPv4 checksum offloading

IPv4 checksum must be calculated for every packet received

IPv4 checksum must be calculated for every packet sent
Usually on a router the Time To Live is decremented, to need re-calculation

Let an ASIC chip on the network card do the work!

Most server network chips today support this and more

Benefit for performance, but beware when using security tools

If every packet in wireshark has wrong checksum, its the network card doing it

Can be turned off, when doing security work

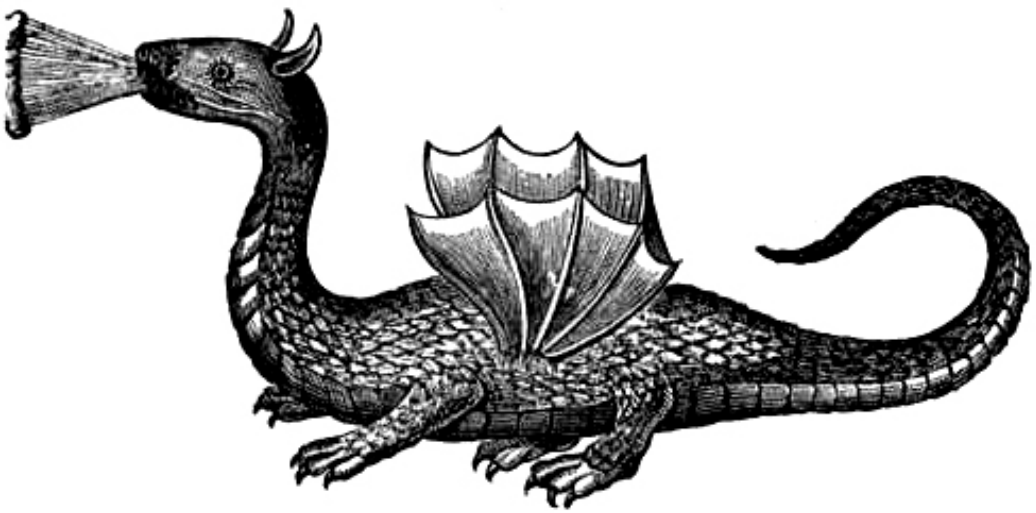
Nping

```
root@KaliVM:~# nping --tcp -p 80 www.zencurity.com
Starting Nping 0.7.70 ( https://nmap.org/nping ) at 2018-09-07 19:06 CEST
SENT (0.0300s) TCP 10.137.0.24:3805 > 185.129.60.130:80  S ttl=64 id=18933 iplen=40  seq=2984847972 win=1480
RCVD (0.0353s) TCP 185.129.60.130:80 > 10.137.0.24:3805  SA ttl=56 id=49674 iplen=44  seq=3654597698 win=16384 <mss 1460>
SENT (1.0305s) TCP 10.137.0.24:3805 > 185.129.60.130:80  S ttl=64 id=18933 iplen=40  seq=2984847972 win=1480
RCVD (1.0391s) TCP 185.129.60.130:80 > 10.137.0.24:3805  SA ttl=56 id=50237 iplen=44  seq=2347926491 win=16384 <mss 1460>
SENT (2.0325s) TCP 10.137.0.24:3805 > 185.129.60.130:80  S ttl=64 id=18933 iplen=40  seq=2984847972 win=1480
RCVD (2.0724s) TCP 185.129.60.130:80 > 10.137.0.24:3805  SA ttl=56 id=9842 iplen=44  seq=2355974413 win=16384 <mss 1460>
SENT (3.0340s) TCP 10.137.0.24:3805 > 185.129.60.130:80  S ttl=64 id=18933 iplen=40  seq=2984847972 win=1480
RCVD (3.0387s) TCP 185.129.60.130:80 > 10.137.0.24:3805  SA ttl=56 id=1836 iplen=44  seq=3230085295 win=16384 <mss 1460>
SENT (4.0362s) TCP 10.137.0.24:3805 > 185.129.60.130:80  S ttl=64 id=18933 iplen=40  seq=2984847972 win=1480
RCVD (4.0549s) TCP 185.129.60.130:80 > 10.137.0.24:3805  SA ttl=56 id=62226 iplen=44  seq=3033492220 win=16384 <mss 1460>

Max rtt: 40.044ms | Min rtt: 4.677ms | Avg rtt: 15.398ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.07 seconds
```

- The Nmap portscanner includes a nice Nping utility
- it allows us to *ping* with TCP, UDP and other protocols
- Controlled with command line options

Challenges in network management



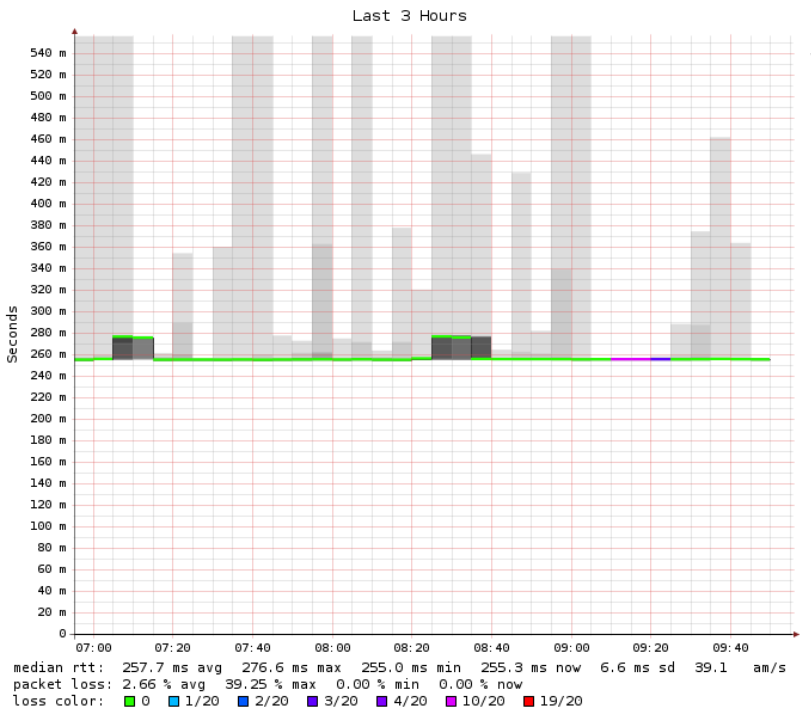
Internet here be dragons

We will jump directly into some solutions

Moving from manual checking to automated

- Wireshark is an advanced manual tool
- How do we continously monitor the network?

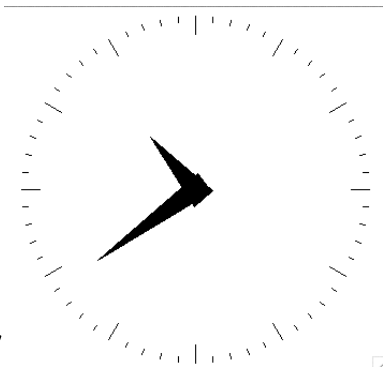
Smokeping packet loss



Old skool, but very usefull https://oss.oetiker.ch/smokeping/

Smokeping latency changed

NTP Network Time Protocol



Vigtigt at netværksenheder bruger korrekt tid, sikkerhed og drift

Server NTP foregår typisk i `/etc/ntp.conf` eller `/etc/ntpd.conf`

det vigtigste er navnet på den/de servere man vil bruge som tidskilde

Brug enten en NTP server hos din udbyder eller en fra http://www.pool.ntp.org/

Eksempelvis:

```
server 0.dk.pool.ntp.org
server 0.europe.pool.ntp.org
server 3.europe.pool.ntp.org
```

SNMP and management

Often we see devices in the network configured using HTTP, Telnet eller SSH

- Where is the configuration stored?
- Is the device functioning?

Today we can use automation like:

- (old RANCID http://www.shrubbery.net/rancid/)
- Ansible https://www.ansible.com/
- Python with SSH libraries
- Oxidized is a network device configuration backup tool. RANCID replacement! https://github.com/ytti/oxidized

SNMP version 2 vs version 3

```
snmpwalk -v2c -c public 172.16.1.2 .1.3.6.1.4.1 | grep site
SNMPv2-SMI::enterprises.13742.6.3.2.2.1.13.1 = STRING: "site-pdu-rack-a-27"
SNMPv2-SMI::enterprises.13742.6.3.5.3.1.3.1.1 = STRING: "switch site-dist-sw-02"
SNMPv2-SMI::enterprises.13742.6.3.5.3.1.3.1.11 = STRING: "router site-mx-01"
```

Use Simple Network Management Protocol, but

- Typical values, packet count on ports, errors observed, speed, firmware, versions,
- SNMP versions 1 and 2c are insecure
- SNMP version 3 created to fix this
- Even better put SNMP and management in seperate VLAN

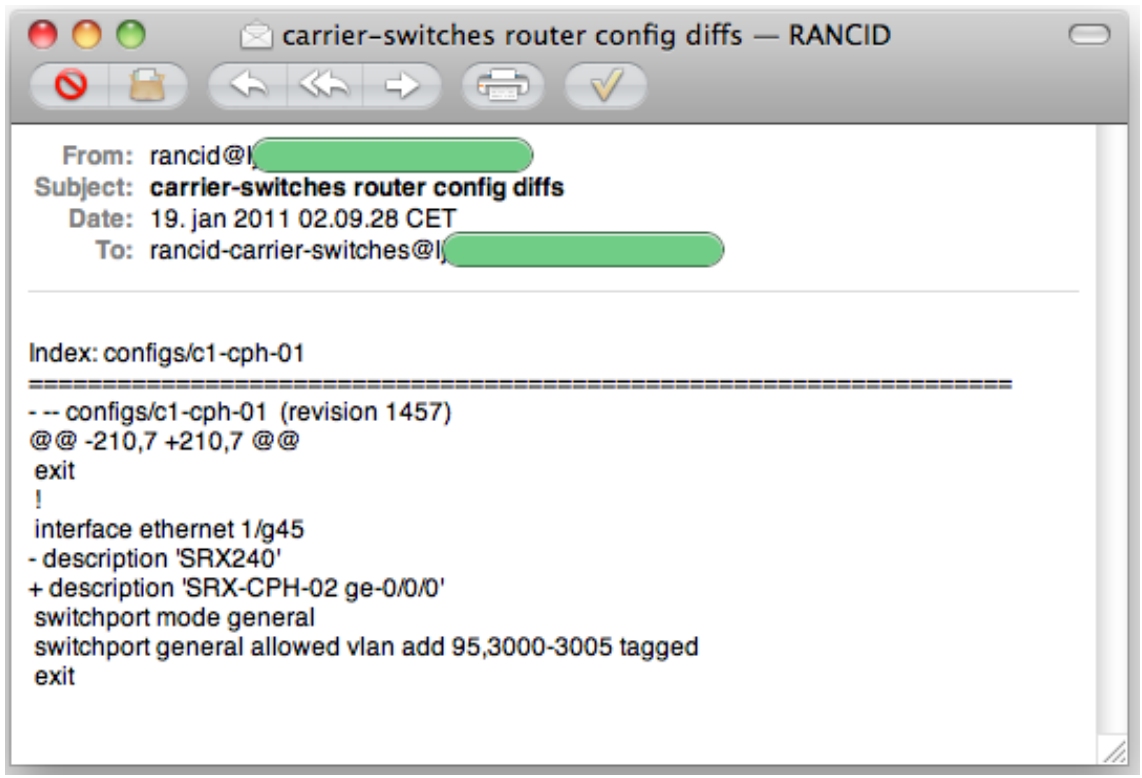
Example for Juniper can be found at:
https://www.juniper.net/documentation/en_US/junos/topics/example/snmpv3-configuration-junos-nm.html

Config example: SNMP

```
snmp {
    description "CORE-SW-02";
    location "Teknikrum, Graested, Denmark";
    contact "noc@zencurity.com";
    community yourcommunitynotmine {
        authorization read-only;
        clients {
            10.1.1.1/32;
            10.1.2.2/32;
        }
    }
}
```

SNMPv2 example

RANCID output



Step 1: configure devices properly

You should always configure your devices properly

Use Secure Shell (SSH)

Configure NTP

Turn on SNMP, probably SNMPv3

Turn on LLDP Link Layer Discovery Protocol – vendor-neutral

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

Configure centralized syslog

Updated firmware, HTTPS and SSH only etc. the usual stuff

Having a minimum template of configuration options is a great starting point

Then use Oxidized and LibreNMS

Oxidized

```
gem install oxidized
gem install oxidized-script oxidized-web
```

- Oxidized is written in Ruby as a replacement for RANCID (Perl)
- Very easy to get running
- Fetches and imports configurations into Git repository, or others
- RESTful API, Web Interface, and command line tools
- https://github.com/ytti/oxidized

Oxidized configuration

```
~/.config/oxidized/router.db
  router01.example.com:ios
  switch01.example.com:procurve
  router02.example.com:ios
```

Easy to configure new device types, added Clavister firewalls once

Config example: Dell switch LLDP

```
interface ethernet 1/xg17
mtu 9216
lldp transmit-tlv port-desc sys-name sys-desc sys-cap
lldp transmit-mgmt
exit
```

LLDP trick using tcpdump

```
[hlk@ljh ~]$ sudo tcpdump -i eth0 ether proto 0x88cc
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
.... wait for it ....
11:03:55.395064 00:1c:23:80:49:ff (oui Unknown) > 01:80:c2:00:00:0e (oui Unknown),
ethertype Unknown (0x88cc), length 60:
0x0000:  0207 0400 1c23 8049 fd04 0705 312f 302f  .....#.I....1/0/
0x0010:  3331 0602 0078 0000 0000 0000 0000 0000  31...x..........
0x0020:  0000 0000 0000 0000 0000 0000 0000       ..............

1 packets captured
3 packets received by filter
0 packets dropped by kernel
```

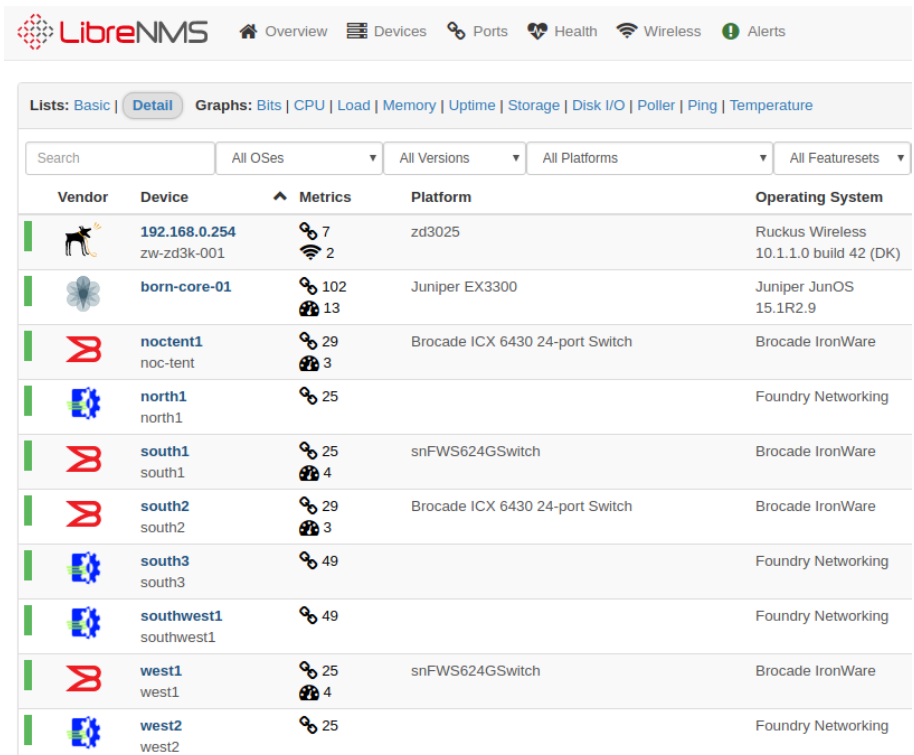
I know **for sure** that this server is in Unit 1 port 31!

LibreNMS a fully featured network monitoring system

- Homepage: https://www.librenms.org/
- **Suggested method:** https://docs.librenms.org/Installation/
- How basic information about devices are presented, from devices when added - and nothing more.
  See how to add a device and add your own. https://docs.librenms.org/Support/Adding-a-Device/
- How SNMP location is used to categorize devices and provice maps, see
  https://docs.librenms.org/Extensions/World-Map/
- How protocols like LLDP allow LibreNMS to make maps, see
  https://docs.librenms.org/Extensions/Network-Map/
- How port description can be used for describing ports
  https://docs.librenms.org/Extensions/Interface-Description-Parsing/

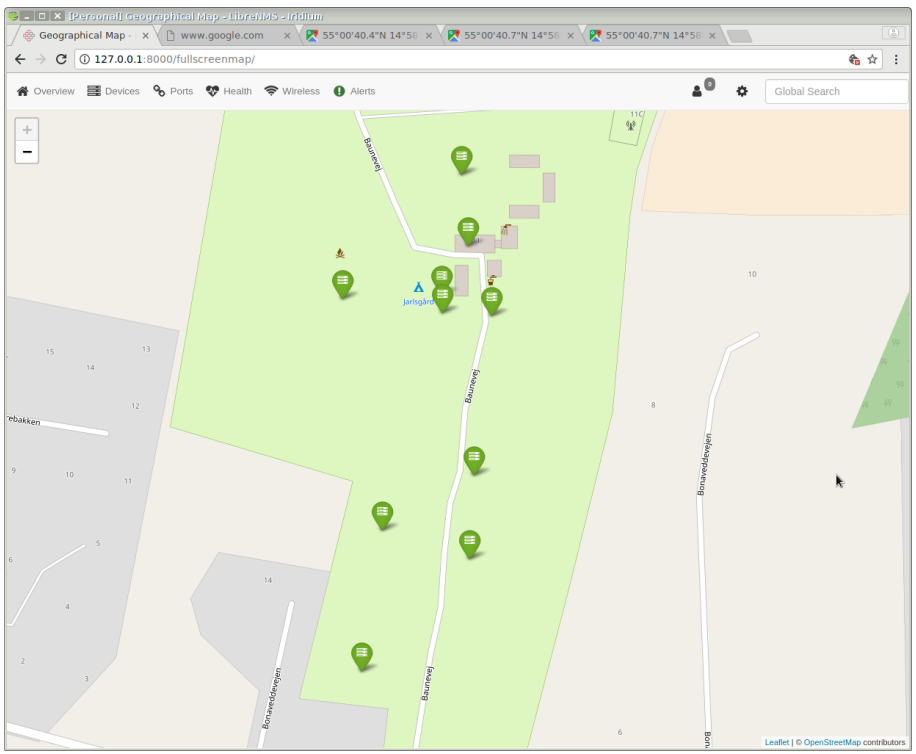
Most of this happens with very little effort. Just **configure devices consistently** and they will be presented nicely.

LibreNMS Automatic discovery

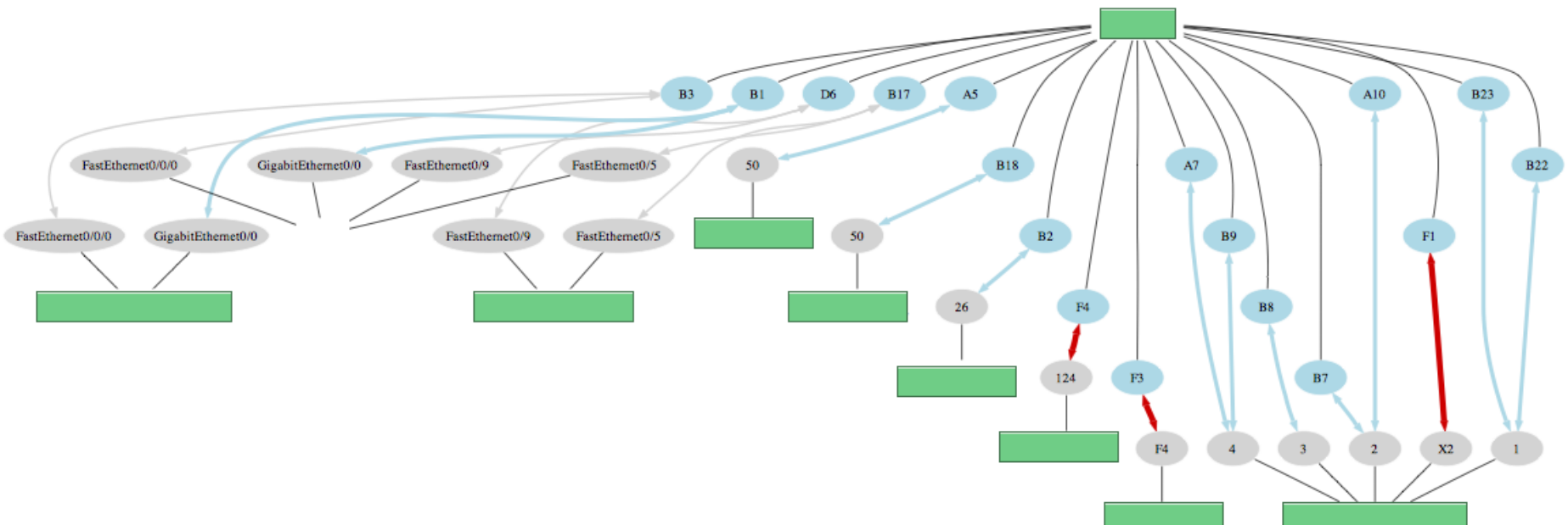


Automatically discover your entire network using CDP, FDP, LLDP, OSPF, BGP, SNMP and ARP.

LibreNMS Geo Location



LLDP spaghetti?



LLDP is needed!

LibreNMS wireless clients

Centralized management SSH, Jump hosts

> A jump server, jump host or jumpbox is a computer on a network used to access and manage devices in a separate security zone. The most common example is managing a host in a DMZ from trusted networks or computers.

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

Advantage, you can configure this server/system to only allow Key based logins – eliminate the possibility for brute-force attacks succeeding

OpenSSH client config with jump host

My recommended SSH client settings, put in `$HOME/.ssh/config`:

```
Host *
    ServerAliveInterval=30
    ServerAliveCountMax=30
    NoHostAuthenticationForLocalhost yes
    HashKnownHosts yes
    UseRoaming no

Host jump-01
  Hostname 10.1.2.3
  Port 12345678

Host fw-site-01 10.1.2.5
  User hlk
  Port 34
  Hostname 10.1.2.5
  ProxyCommand ssh -q -a -x jump-01 -W %h:%p
```

I configure fw using both hostname and IP,
then I can use name, and any program using IP get this config too

Ansible

From my course materials:
Ansible is great for automating stuff, so by running the playbooks we can get a whole lot of programs installed, files modified - avoiding the Vi editor.

- Easy to read, even if you don't know much about YAML
- https://www.ansible.com/ and https://en.wikipedia.org/wiki/Ansible_(software)
- Great documentation
  https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html

Why Ansible

Platform options Ansible:

```
CloudEngine OS, CNOS, Dell OS6, Dell OS9 Dell OS10, ENOS, EOS, ERIC_ECCLI, EXOS,
FRR, ICX, IOS, IOS-XR, IronWare, Junos OS, Meraki, Pluribus NETVISOR, NOS, NXOS,
RouterOS, SLX-OS, VOSS, VyOS, WeOS 4

plus routers based on Linux, OpenBSD, FreeBSD etc.
```

One management system with many uses, free to download and use

- Generic configuration management – and you end up running support systems, network near systems
- Ansible for Network Automation
  https://docs.ansible.com/ansible/latest/network/index.html
- Allows you to install, configure and run your network management systems – like LibreNMS, Nipap

Ansible and Junos – Juniper Networks devices

> Juniper Networks provides support for using Ansible to manage devices running Junos OS. Starting in Ansible Release 2.1, Ansible natively includes a number of core modules that can be used to manage devices running Junos OS. In addition, Juniper Networks provides the Juniper.junos Ansible role, which is hosted on the Ansible Galaxy website and includes additional modules for Junos OS.

Source:

https://www.juniper.net/documentation/en_US/junos-ansible/topics/task/installation/junos-ansible-server-installing.html

- Minimal configuration changes: `user@host# set netconf ssh`

How cool is that, centralized configuration management with source code!

Ansible Dependencies

Adding a few extra for Junos

```
user@NMS:~/projects/network-automation$ sudo pip3 install ncclient
user@NMS:~/projects/network-automation$ sudo pip3 install jxmlease
```

- Ansible based on Python, only need Python installed https://www.python.org/
- Often you use Secure Shell for connecting to servers https://www.openssh.com/
- Easy to configure SSH keys, for secure connections
- Can use sudo or doas commands for root access when needed

How Ansible Works: inventory files

List your hosts in one or multiple text files:

```
user@NMS:~/projects/network-automation$ cat hosts
[all:vars]
#ansible_ssh_port=34443

[switches]
ex2200-camp ansible_ssh_host=10.0.42.5

[switches:vars]
ansible_connection=netconf
ansible_netconf_user=ansible
ansible_netconf_pass=henrik42
ansible_ssh_user=root
ansible_ssh_pass=juniper123
```

- Inventory files specify the hosts we work with – only Junos config shown here
- Real inventory for a site with development and staging may be 500 lines

How Ansible Works: ad hoc commands

Using the inventory file you can run commands with Ansible:

```
ansible -m ping new-server
ansible -a "date" new-server
ansible -m shell -a "grep a /etc/something" new-server
```

- Running commands on multiple servers is easy now
- This alone has value in itself
- Checking settings on servers
- Making small changes to servers
- Then playbooks can automate your work!

Ansible Ping

```
user@NMS:~/projects/network-automation$ ansible -i hosts  -m ping ex2200-camp
ex2200-camp | SUCCESS =>
    "ansible_facts":
        "discovered_interpreter_python": "/usr/bin/python3.9"
    ,
    "changed": false,
    "ping": "pong"
```

- Checking connectivity with ping is always a good start

How Ansible Works: Playbooks

The benefit comes with tasks listed in playbooks – example playbook content, installing software using APT:

```
apt:
    name: "{{ packages }}"
    vars:
      packages:
        - nmap
        - curl
        - iperf
        ...
```

Running it:

```
cd kramse-labs/suricatazeek
ansible-playbook -v 1-dependencies.yml 2-suricatazeek.yml 3-elasticstack.yml
```

Installs a full server with Elasticsearch, Kibana, Zeek and Suricata in 10 minutes

"YAML (a recursive acronym for "YAML Ain't Markup Language") is a human-readable data-serialization language."https://en.wikipedia.org/wiki/YAML

Get configuration with playbook

```
# Filename: get_config.yaml

- name: Demonstration of the get_config Ansible module
  gather_facts: false
  hosts: all
  tasks:
    - name: Execute the get_config RPC
      netconf_get:
        display: json
      register: result
    - name: Print the configuration as JSON
      debug:
        var: result.output
```

- Next step would be to convert this running config into a full Ansible setup
- The Ansible modules are documented nicely!

https://docs.ansible.com/ansible/latest/collections/junipernetworks/junos/junos_config_module.html

Python and YAML – Git

- We need to store configurations
- Run playbooks
- Problem: Remember what we did, when, how
- Solution: use git for the playbooks
- Not the only version control system, but my preferred one

How Ansible Works: typical execution

```
ansible-playbook -i hosts.cph1 -K infrastructure-firewalls.yml -t pf.conf --check --diff

ansible-playbook -i hosts.cph1 -K infrastructure-firewalls.yml -t pf.conf

ansible-playbook -i hosts.cph1 -K infrastructure-nagios.yml -t config-only

ansible-playbook -i smartboxes -K create-pf-conf.yml -l smartbox-xxx-01
```

- Pro tip: check before you push out changes to production networks ☺
- Check will see if something needs changing
- Diff will show the changes about to be made
- Having configuration in Git improves things a lot!

Get ready, Up and running with Ansible

Prequisites for Ansible - you need a Linux machine:

- python language - Ansible uses this
- ssh keys - remote login without passwords
- Sudo - allow regular users to do superuser tasks
- Recommended tool: `ssh-copy-id` for getting your key on new server
- Recommended Change: `sshd_config` - no passwords allowed, no bruteforce
- Recommended to use: jump hosts/ProxyCommand in `ssh_config`
- Highly recommended: Git and/or github for version control

Official docs:
https://docs.ansible.com/ansible/latest/installation_guide/index.html

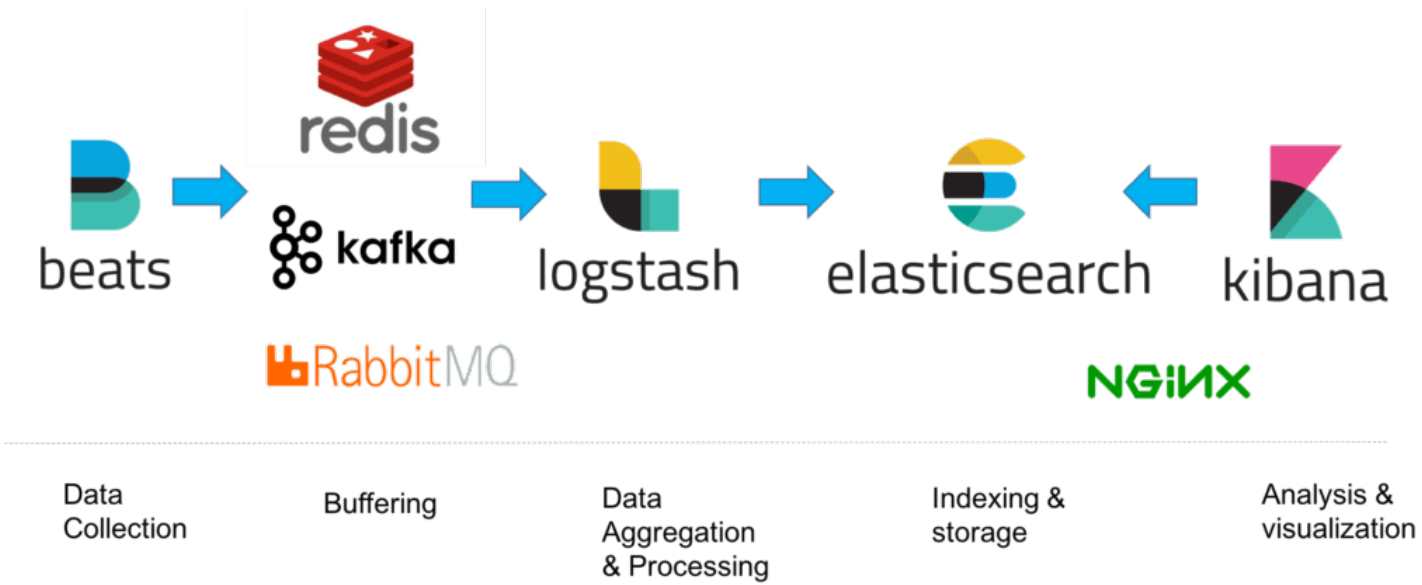
Ansible configuration management

```
- apt: name= item  state=latest
  with_items:
        - unzip
        - elasticsearch
        - logstash
        - redis-server
        - nginx
- lineinfile: "dest=/etc/elasticsearch/elasticsearch.yml state=present
  regexp='network.host: localhost' line='network.host: localhost'"
- name: Move elasticsearch data into /data
  command: creates=/data/elasticsearch mv /var/lib/elasticsearch /data/
- name: Make link to /data/elasticsearch
  file: state=link src=/data/elasticsearch path=/var/lib/elasticsearch
```

Example playbooks:

`git clone` https://github.com/kramse/ansible-workshop

Logging



- The world needs answers
- You can only present answers, if you got them
- Using logging you can investigate what happened

Netflow – Lets move to detailed monitoring and logging

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, IP Type of Service
- Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols

Today we can use Netflow version 9 or IPFIX with more fields available, or 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

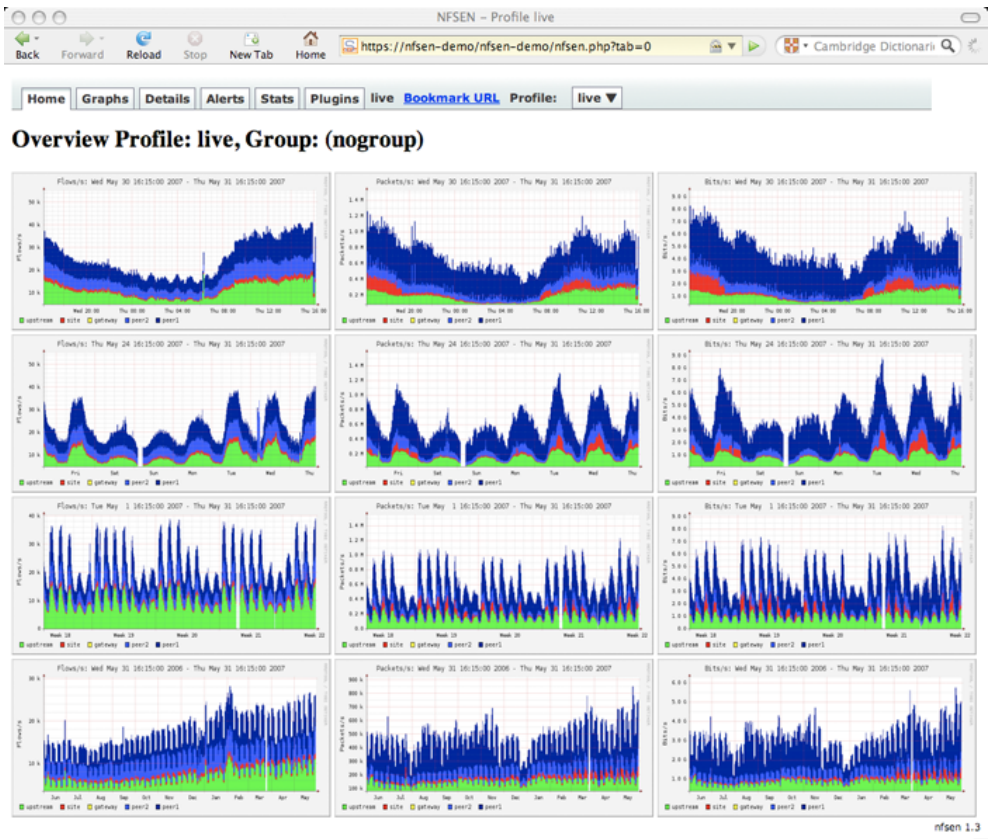
NFSen is an old but free application http://nfsen.sourceforge.net/

Source:

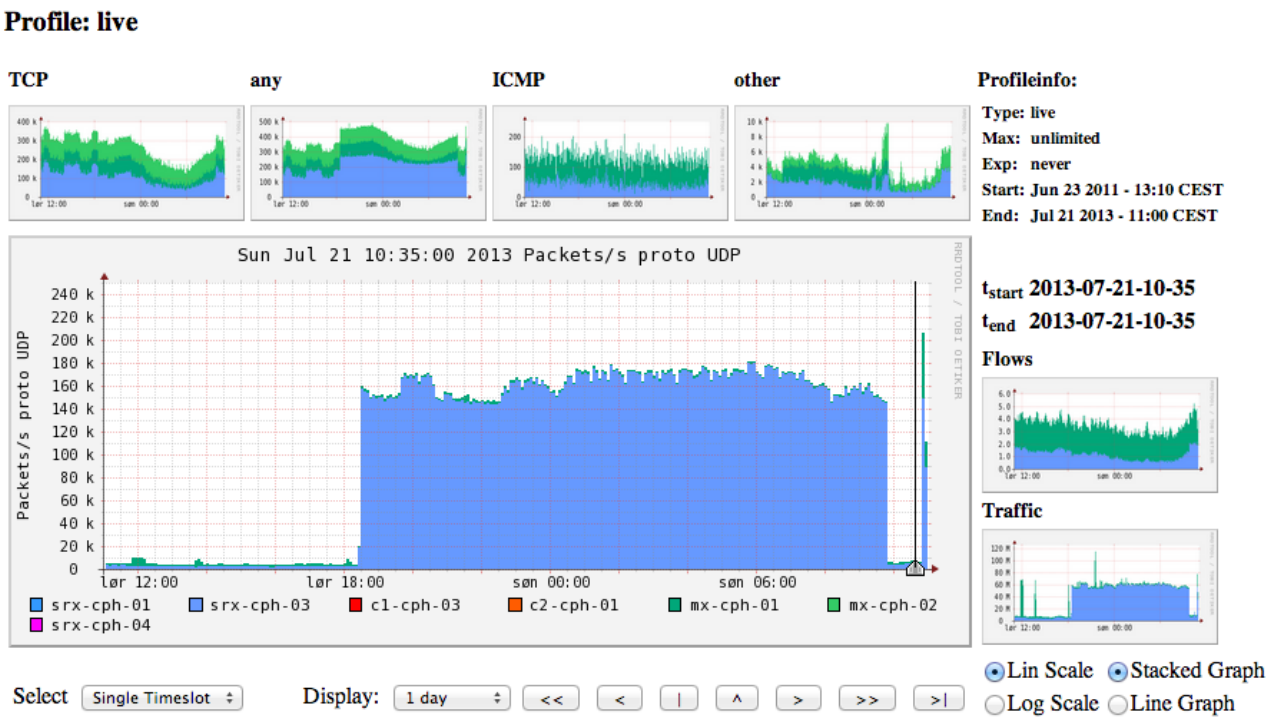
https://en.wikipedia.org/wiki/NetFlow

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

Netflow using NFSen

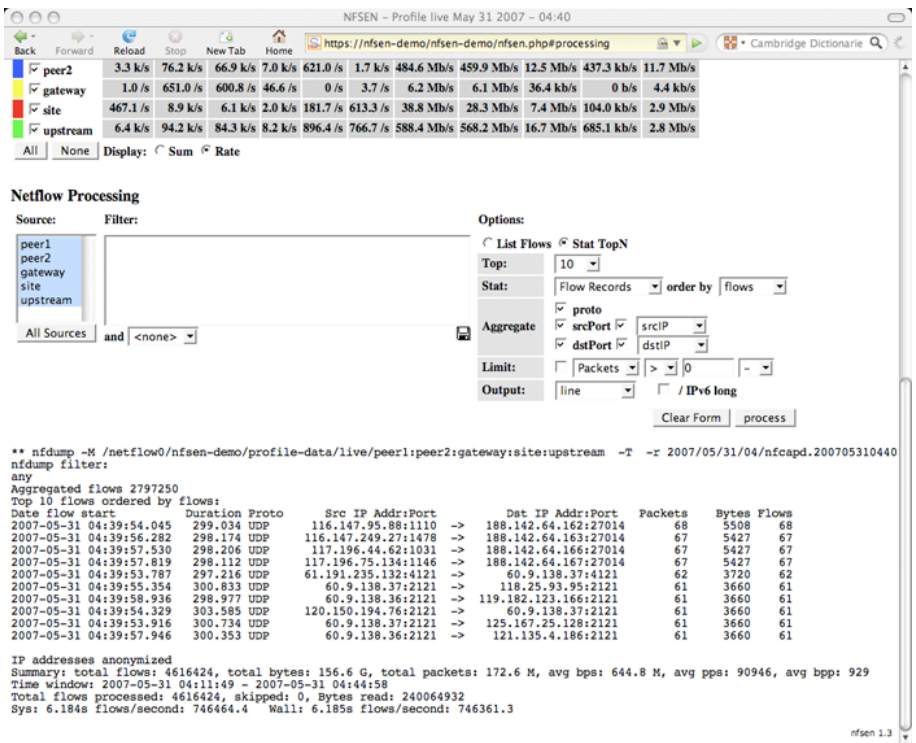


Netflow NFSen



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

Netflow processing from the web interface



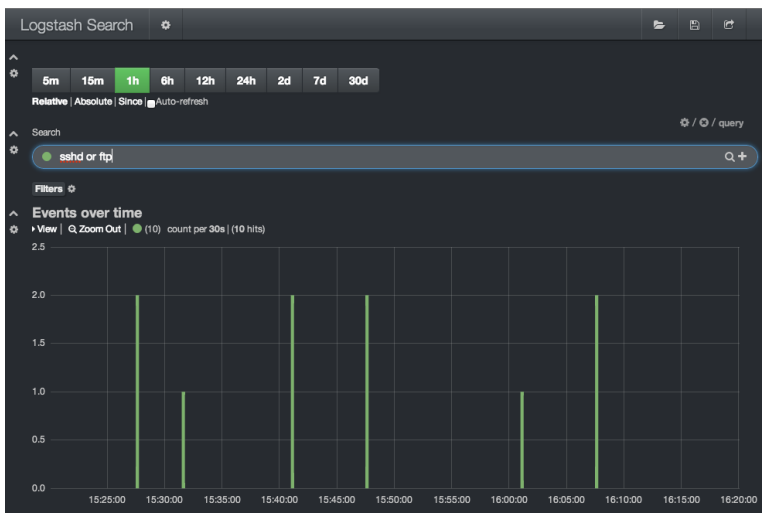
Bringing the power of the command line forward

ElastiFlow

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
PS I havent tried it in real life, yet – but a friend has

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!

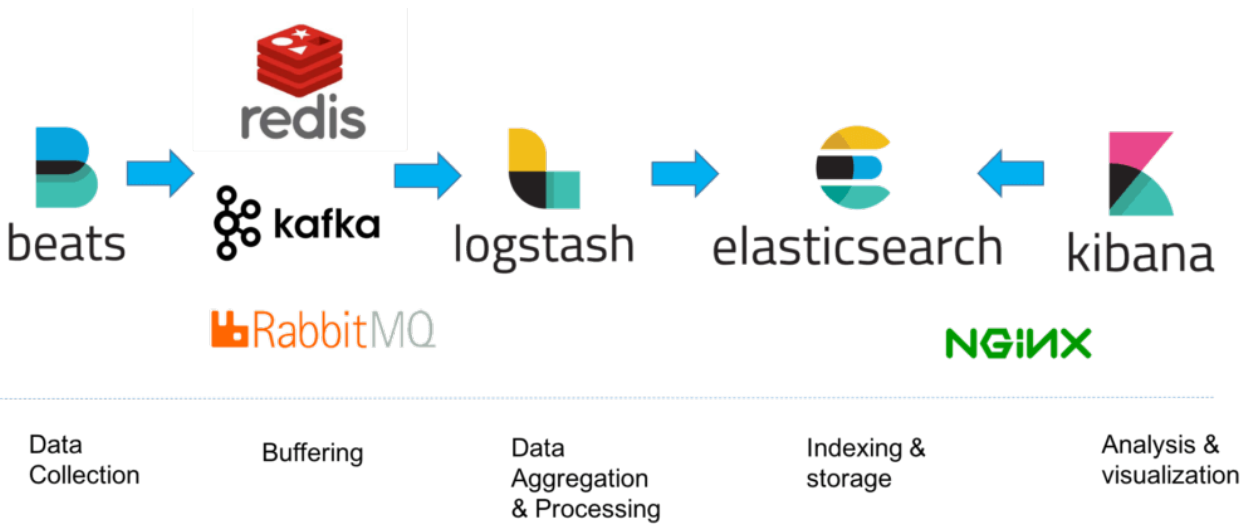
Other popular examples include Graylog and Grafana

Big Data tools: Elasticsearch and Kibana

Elasticsearch is an open source distributed, RESTful search and analytics engine capable of solving a growing number of use cases. Source: https://www.elastic.co

- We are all Devops now, even security people!
- Highly recommended for a lot of data visualisation
- Non-programmers can create, save, and share dashboards

Architecture



- Real production environments often add some buffering in between
- Allows the ingestion to become more smooth, no lost messages

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

SIEM and logging systems

> You could **buy a bunch of expensive gear**, point it all to a **log management** or a **security incident and event management (SIEM)** system, and let it automatically tell you what it knows. Some incident response teams may start this way, but unfortunately, many never evolve. **Working only with what the SIEM tells you**, versus what you have configured it to tell you **based on your contextual data**, will invariably fail. Truly demonstrating **value** from security monitoring and incident response **requires a major effort**. As with all projects, **planning** is the most important phase. **Designing** an approach that works best for you requires significant effort up front, but offers a great payout later in the form of **meaningful incident response and overall improved security**.

Source: Crafting the InfoSec Playbook, 4 A Data-Centric Approach to Security Monitoring (bold by me) by Jeff Bollinger, Brandon Enright, and Matthew Valites ISBN: 9781491949405

- I recommend pre-filtering, see what noise your devices *would send*
  "Collecting only relevant data can have a direct impact on reducing costs as well."
- Same is recommended in CIP page 50: Just the Facts
- Normalization – "Data normalization for the purposes of log mining is the process by which a portion, or field, of a log event is transformed into its canonical form."

Metadata – enrichment

Metadata	Metacategory: Data
The DNS lookup occurred at a certain time.	Timestamp: 278621182
The internal host sent a DNS PTR request.	Network protocol: DNS PTR
The internal host had a hostname.	Location: Desktop subnet
	Source IP Address: 1.1.1.2
	Hostname: windowspc22.company.com
The internal host resolved an external host.	Location: External
	Destination IP Address: 255.123.215.3
	Hostname: dgf7adfnkjhh.com
The external host was hosted by a dynamic DNS provider.	Network: Shady DDnS Provider Inc.
	ASN: SHADY232
	Reputation: Historically risky network
The remote hostname appeared randomly generated.	Hostname: dgf7adfnkjhh.com
	Category: Unusual, nonlinguistic

Source: picture from Crafting the InfoSec Playbook, CIP

Metadata + Context

Summary, what to log

CIP 7 Tools of the Trade, need to know NetFlow, DNS, and HTTP proxy logs in the real-world

- Defense in Depth – we will never catch everything
- Log Management: The Security Event Data Warehouse
- Intrusion Detection Isn't Dead
- DNS, the One True King – Logging and analyzing DNS transactions, Blocking DNS requests or responses
- HTTP Is the Platform: Web Proxies – Web proxies allow you to solve additional security problems
- Rolling Packet Capture – In a perfect world, we would have full packet capture everywhere

Detection Capabilities

Security incidents happen, but what happens. One of the actions to reduce impact of incidents are done in preparing for incidents.

> *Preparation* for an attack, establish procedures and mechanisms for detecting and responding to attacks

Preparation will enable easy **identification** of affected systems, better **containment** which systems are likely to be infected, **eradication** what happened – how to do the **eradication** and **recovery**.

Strategy for implementing identification and detection

We recommend that the following strategy is used for implementing identification and detection.

We have the following recommendations and actions points for logging:

- ☐ Enable system logging from servers
- ☐ Enable system logging from network devices
- ☐ Centralize logging
- ☐ Add search facilities and dashboards
- ☐ Perform system audits manually or automatically
- ☐ Setup notification and notification procedures

Extended Sources

When a basic logging infrastructure is setup, it can be expanded to increase coverage, by adding more sources:

- DNS query logging – will enable multiple cases to be resolved, example malware identification and tracing, when was a malware domain queried, when was the first infection
- Session data from Firewalls, Netflow – traffic patterns can be investigated and both attacks and cases like exfiltration can likely be seen

Hint: Take the sources available first, make a proof-of-concept, expand coverage

Data Analysis Skills

Although we could spend an entire book creating an exhaustive list of skills needed to be a good security data scientist, this chapter covers the following skills/domains that a data scientist will benefit from knowing within information security:

- Domain expertise—Setting and maintaining a purpose to the analysis
- Data management—Being able to prepare, store, and maintain data
- Programming—The glue that connects data to analysis
- Statistics—To learn from the data
- Visualization—Communicating the results effectively

It might be easy to label any one of these skills as the most important, but in reality, the whole is greater than the sum of its parts. Each of these contributes a significant and important piece to the workings of security data science.

Source: *Data-Driven Security: Analysis, Visualization and Dashboards* Jay Jacobs, Bob Rudis
ISBN: 978-1-118-79372-5 February 2014 https://datadrivensecurity.info/ - short DDS

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

Automated packet sniffing tools

The Zeek Network Security Monitor

Zeek – Network Security Monitor https://zeek.org

Suricata – open source, mature, fast and robust network threat detection https://suricata-ids.org/

ntopng – High-speed web-based traffic analysis https://www.ntop.org/

Maltrail – Malicious traffic detection system https://github.com/stamparm/MalTrail

Slide included as a reference for Network SECURITY Monitoring

Side note: Zeek Security Monitor handles formats differently

Zeek has files formatted with a header:

```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id
        rtt     query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name      AA
        TC      RD      RA      Z       answers ttls    rejected

1538982372.416180 CD12Dc1SpQm42QW4G3 10.xxx.0.145 57476 10.x.y.141 53 udp 20383
0.045021 www.dr.dk 1 C_INTERNET 1 A 0 NOERROR F F T T 0
   www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93 60.000000,20409.000000,20.000000 F
```

Note: this show ALL the fields captured and dissected by Zeek, there is a nice utility program bro-cut which can select specific fields:

```
root@NMS-VM:/var/spool/bro/bro# cat dns.log | bro-cut -d ts query answers | grep dr.dk
2018-10-08T09:06:12+0200 www.dr.dk www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93
```

Can also just use JSON now via Filebeat

Questions?

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

You are always welcome to send me questions later via email

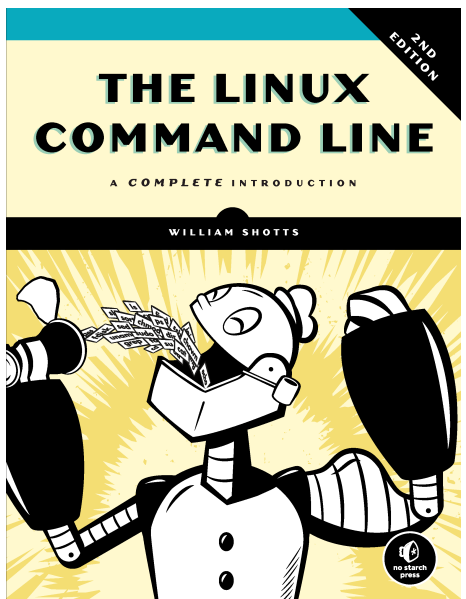
Mobile: +45 2026 6000

Email: hlk@zencurity.com

Recommended further reading

- Campus Network Security: High Level Overview , Network Startup Resource Center https://nsrc.org/workshops/2018/myren-nsrc-cndo/networking/cndo/en/presentations/Campus_Security_Overview.pdf
- Campus Operations Best Current Practice, Network Startup Resource Center https://nsrc.org/workshops/2018/tenet-nsrc-cndo/networking/cndo/en/presentations/Campus_Operations_BCP.pdf
- Mutually Agreed Norms for Routing Security (MANRS) https://www.manrs.org/wp-content/uploads/2018/09/MANRS_PDF_Sep2016.pdf
- RFC2827: Network Ingress Filtering: Defeating Denial of Service Attacks https://tools.ietf.org/html/rfc2827

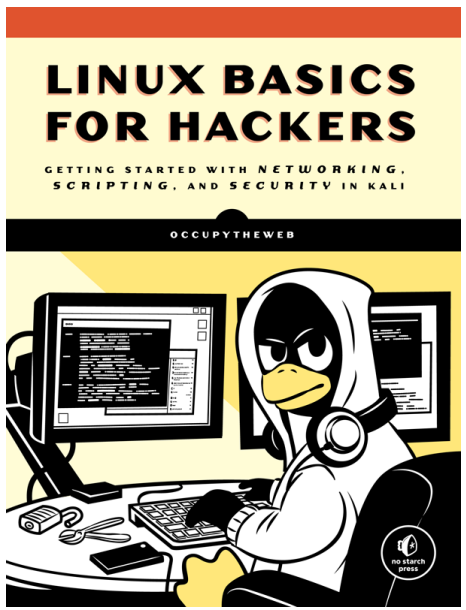
Book: The Linux Command Line



The Linux Command Line, 2nd Edition A Complete Introduction by William Shotts

Print: https://nostarch.com/tlcl2
Download – internet edition https://sourceforge.net/projects/linuxcommand

Book: Linux Basics for Hackers (LBhf)



Linux Basics for Hackers Getting Started with Networking, Scripting, and Security in Kali by OccupyTheWeb December 2018, 248 pp. ISBN-13: 9781593278557

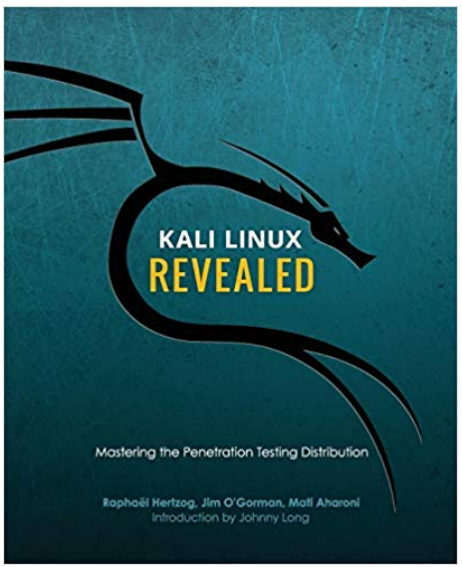
https://nostarch.com/linuxbasicsforhackers Explains how to use Linux

Book: The Debian Administrator's Handbook (DEB)

The Debian Administrator's Handbook, Raphaël Hertzog and Roland Mas
https://debian-handbook.info/ - shortened DEB

Not curriculum but explains how to use Debian Linux

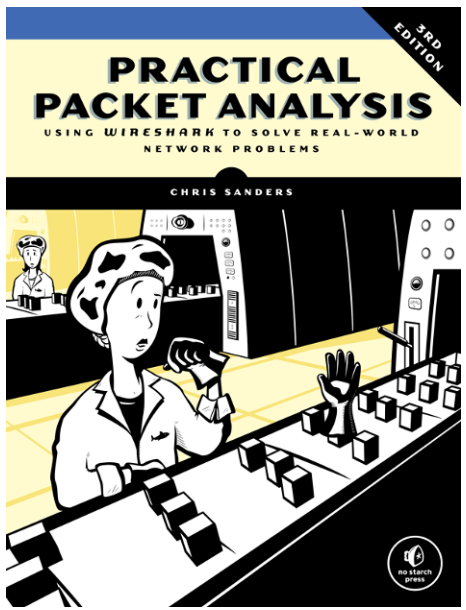
Book: Kali Linux Revealed (KLR)



Kali Linux Revealed Mastering the Penetration Testing Distribution

https://www.kali.org/download-kali-linux-revealed-book/
Explains how to install Kali Linux

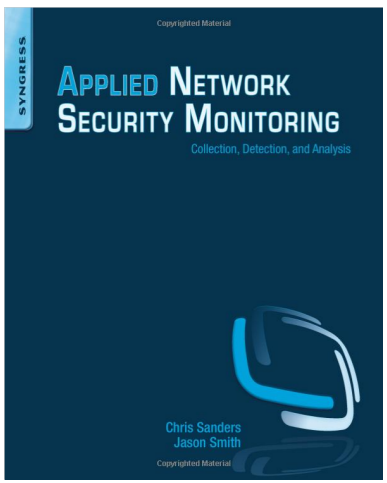
Book: Practical Packet Analysis (PPA)



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

Book: Applied Network Security Monitoring (ANSM)

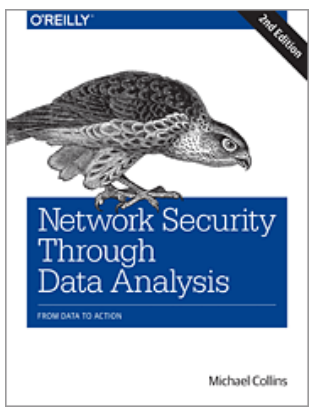


Applied Network Security Monitoring: Collection, Detection, and Analysis 1st Edition

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

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

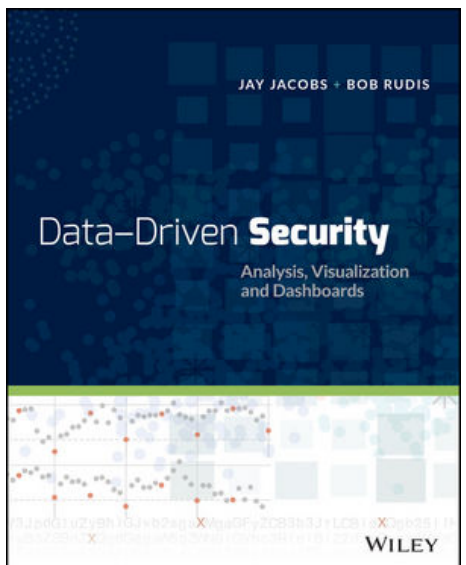
Network Security Through Data Analysis



Low page count, but high value! Recommended.

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

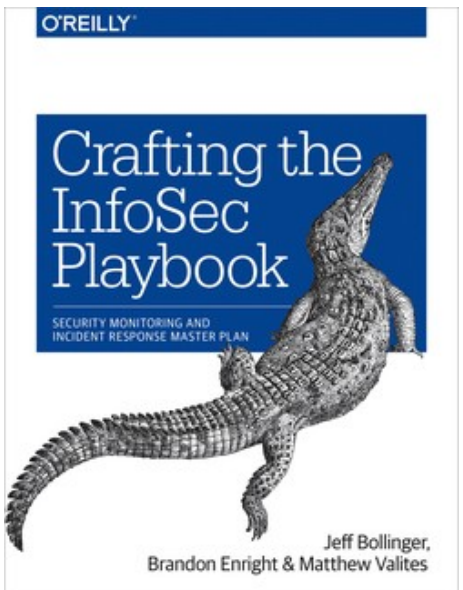
Data-Driven Security: Analysis, Visualization and Dashboards



Data-Driven Security: Analysis, Visualization and Dashboards Jay Jacobs, Bob Rudis
ISBN: 978-1-118-79372-5 February 2014 https://datadrivensecurity.info/ - short DDS

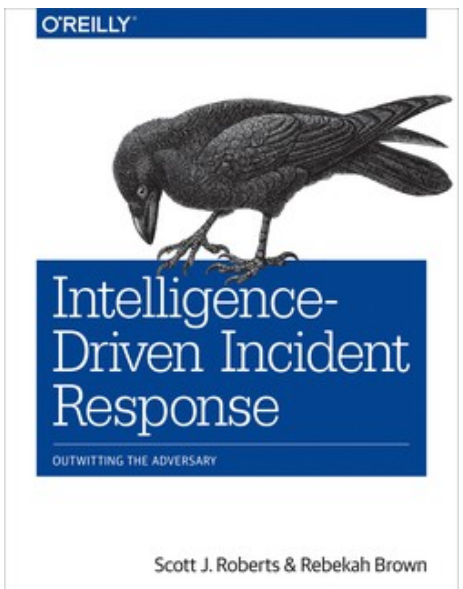
Our main book for this course. We will read a lot from this one.

Crafting the InfoSec Playbook



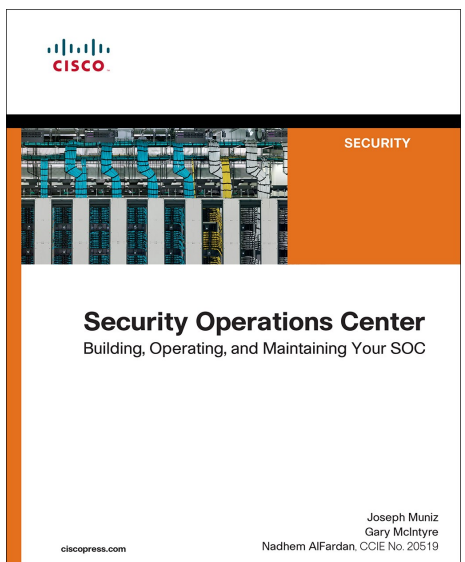
Crafting the InfoSec Playbook: Security Monitoring and Incident Response Master Plan
by Jeff Bollinger, Brandon Enright, and Matthew Valites ISBN: 9781491949405 - short CIP

Intelligence-Driven Incident Response



Intelligence-Driven Incident Response
Scott Roberts ISBN: 9781491934944 - short IDI

Security Operations Center



Security Operations Center: Building, Operating, and Maintaining your SOC
ISBN: 9780134052014 Joseph Muniz - short SOC