



Welcome to

2. Network Security Threats

Communication and Network Security 2024

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

Slides are available as PDF, kramse@Codeberg
2-Network-Security-Threats.tex in the repo security-courses

Goals for today



- Introduce network security threats, there are a lot! Dont remember all of them
- Take special note of the prerequisites for the attacks, how can we interrupt attacks
Example ARP spoofing is within a network segment, move some servers into seperate segments ☺
- DDoS has become a problem for more organisations, an advanced concept - lookup when you need this information
- Using the right tools make you more efficient, Nmap can discover and scan a whole network in minutes

Photo by Thomas Galler on Unsplash

Plan for today



Subjects

- Network Security Threats
- ARP spoofing, ICMP redirects, the classics
- Person in the middle attacks
- Network Scanning
- Intro to routing protocols attacks, BGP intro and hijacking
- DDoS and flooding

Exercises

- ARP spoofing, EtherApe and ettercap
- Nmap and Nping
- Pcap-diff
- Flooding with hping3

Reading Summary



Read: PPA chapters 4,5,6 - 66 pages

(Original reading PPA chapter 5,6,7,8,12 - 124 pages)

Skim: Strange Attractors and TCP/IP Sequence Number Analysis

The chapters read for this time, main things:

PPA chapter 4: Working with Captured Packets

- Reading, saving and exporting capture files with packets from Wireshark
- Merging two files with Wireshark
- Finding, Marking, Printing packets
- Capture options including Capture Filters vs Display Filters
- Capture Filters are libpcap/Berkeley Packet Filter (BPF) syntax - same as tcpdump

also more than 20 illustrations in one chapter, not hard to read

Great network security comes from knowing networks!

Reading Summary, continued



The chapters read for this time, main things:

PPA chapter 5: Advanced Wireshark Features

- Networks have end-points and conversations on multiple layers
- Wireshark is advanced, try right-clicking different places
- Name resolution includes low level MAC addresses, and IP - names
- More than 1000 dissectors, but beware some have security issues!

Reading Summary, continued



PPA chapter 6: Packet Analysis on the Command Line

- 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>

Later in this course we will introduce:

Zeek (formerly Bro)[2] is a free and open-source software network analysis framework; it was originally developed in 1994 by Vern Paxson <https://en.wikipedia.org/wiki/Zeek>

Everything you do on the command line can be automated easily



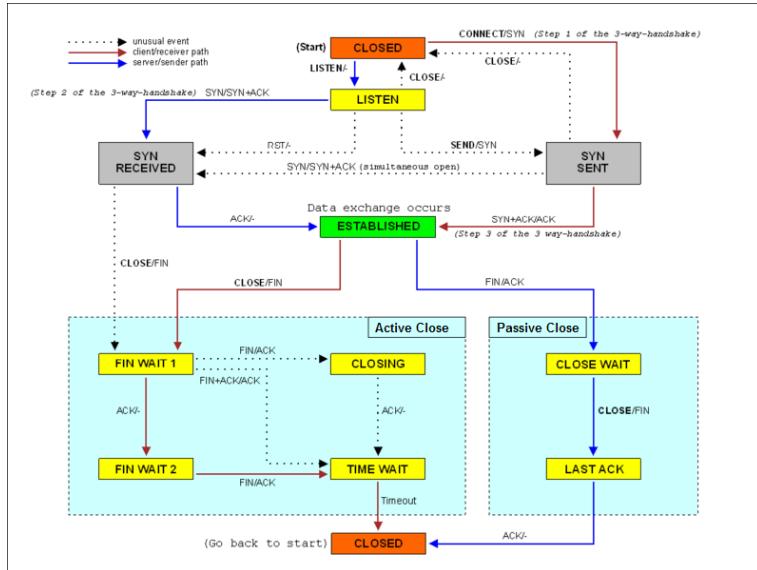
Reading Summary, continued

```
user@Projects:~$ ping -s 1472 -M do 91.102.91.18
PING 91.102.91.18 (91.102.91.18) 1472(1500) bytes of data.
1480 bytes from 91.102.91.18: icmp_seq=1 ttl=244 time=7.43 ms
1480 bytes from 91.102.91.18: icmp_seq=2 ttl=244 time=7.20 ms
...
user@Projects:~$ ping -s 1474 -M do 91.102.91.18
PING 91.102.91.18 (91.102.91.18) 1474(1502) bytes of data.
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500
^C
--- 91.102.91.18 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1025ms
```

PPA chapter 7: Network Layer Protocols, was originally for today!

- What is normal network traffic?
- Great reference chapter for basic protocols
- Plus basic IPv6
- Re PATH MTU etc. Linux MTU 1500 check ping -s 1472 -M do

Reading Summary, continued



PPA chapter 8: Transport Layer Protocols, was originally for today!

- Again the TCP 3-way handshake is described Note: can be done in 4 packets
- Closed TCP returns Reset (RST) packet, closed UDP returns ICMP port unreachable

Reading Summary, continued



Table 12-1: Common Passive Fingerprinting Values

Protocol header	Field	Default value	Platform
IP	Initial time to live	64	NMap, BSD, OS X, Linux
		128	Novell, Windows
		255	Cisco IOS, Palm OS, Solaris
IP	Don't fragment flag	Set	BSD, OS X, Linux, Novell, Windows, Palm OS, Solaris
		Not set	Nmap, Cisco IOS
TCP	Maximum segment size	0	Nmap
		1440–1460	Windows, Novell
		1460	BSD, OS X, Linux, Solaris

(continued)

PPA chapter 12: Packet Analysis for Security, was originally for today!

- Syn scan
- the ARP protocol is inherently insecure
- All attacks have signatures, some more noisy than others

Reading Summary, continued



Strange Attractors and TCP/IP Sequence Number Analysis, Michal Zalewski
<http://lcamtuf.coredump.cx/newtcp/>

Continued from last time TCP/IP Sequence Numbers

- Lets just check out the cool graphs
- Sending lots of packets from attack tools is very possible today

Basic port scanning



What is a port scan

Testing all values possible for port number from 0/1 to 65535

Goal is to identify open ports, listening and vulnerable services

Most often TCP og UDP scan

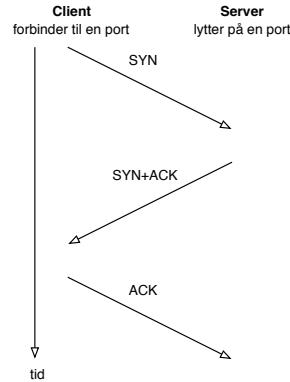
TCP scanning is more reliable than UDP scanning

TCP handshake must respond with SYN-ACK packets

UDP applications respond differently – if they even respond
so probes with real requests may get response, no firewall they respond with ICMP on closed
ports

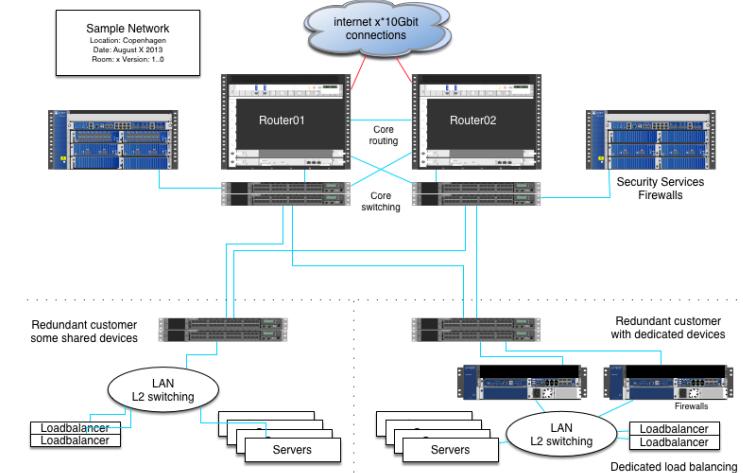
Use the GUI program Zenmap while learning Nmap

TCP three-way handshake



- **TCP SYN half-open** scans
- in the old days systems would only log when a full TCP connection was setup
 - so doing only half open it was a *stealth*-scans
- Today system and IDS intrusion detection can easily monitor for this
- Sending a lot of SYN packets can create a Denial of Service – **SYN-flooding**

Scope: select systems for testing



- Routers in front of critical systems and networks - availability
- Firewalls – are traffic flows restricted
- Mail servers – open for relaying
- Web servers – remote code execution in web systems, data download



Ping and port sweep

Scans across the network are named sweeps

Ping sweeps using ICMP Ping probes

Port sweep trying to find a specific service, like port 80 web

Quite easy to see in network traffic:

- Selecting two IP-addresses not in use
- Should not see any traffic, but if it does, its being scanned
- If traffic is received on both addresses, its a sweep – if they are a bit apart it is even better, like 10.0.0.100 and 10.0.0.200

Pro tip: a Great network intrusion detection engine (IDS), is Suricata suricata-ids.org

what is Nmap today



Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.

Initial release September 1997; +20 years ago

Today a package of programs for Windows, Mac, BSD, Linux, ... source

Flexible, powerful, and free! Includes other tools!

Lets check release notes, 7.70 pt.

<http://seclists.org/nmap-announce/2018/0>

Bonus info: you can help Nmap by submitting fingerprints



Nmap port sweep for web servers

```
root@cornerstone:~# nmap -p80,443 172.29.0.0/24
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:31 CET
Nmap scan report for 172.29.0.1
Host is up (0.00016s latency).
PORT      STATE      SERVICE
80/tcp    open       http
443/tcp   filtered  https
MAC Address: 00:50:56:C0:00:08 (VMware)
```

```
Nmap scan report for 172.29.0.138
Host is up (0.00012s latency).
PORT      STATE      SERVICE
80/tcp    open       http
443/tcp   closed    https
MAC Address: 00:0C:29:46:22:FB (VMware)
```



Nmap port sweep for SNMP port 161/UDP

```
root@cornerstone:~# nmap -sU -p 161 172.29.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:30 CET
Nmap scan report for 172.29.0.1
Host is up (0.00015s latency).
PORT      STATE      SERVICE
161/udp open|filtered snmp
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap scan report for 172.29.0.138
Host is up (0.00011s latency).
PORT      STATE      SERVICE
161/udp closed snmp
MAC Address: 00:0C:29:46:22:FB (VMware)
...
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.18 seconds
```

More reliable to use Nmap script with probes like `--script=snmp-info`

Nmap Advanced OS detection

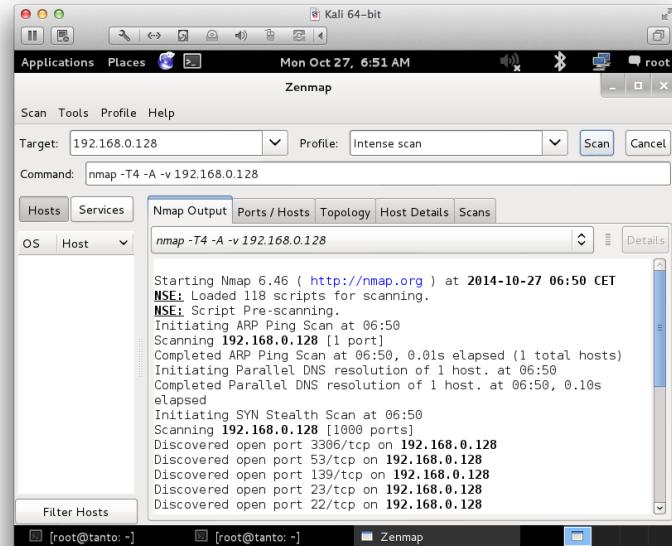


```
root@cornerstone:~# nmap -A -p80,443 172.29.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:37 CET
Nmap scan report for 172.29.0.1
Host is up (0.00027s latency).

PORT      STATE      SERVICE VERSION
80/tcp    open       http      Apache httpd 2.2.26 ((Unix) DAV/2 mod_ssl/2.2.26 OpenSSL/0.9.8zc)
|_http-title: Site doesn't have a title (text/html).
443/tcp   filtered https
MAC Address: 00:50:56:C0:00:08 (VMware)
Device type: media device|general purpose|phone
Running: Apple iOS 6.X|4.X|5.X, Apple Mac OS X 10.7.X|10.9.X|10.8.X
OS details: Apple iOS 6.1.3, Apple Mac OS X 10.7.0 (Lion) - 10.9.2 (Mavericks)
or iOS 4.1 - 7.1 (Darwin 10.0.0 - 14.0.0), Apple Mac OS X 10.8 - 10.8.3 (Mountain Lion)
or iOS 5.1.1 - 6.1.5 (Darwin 12.0.0 - 13.0.0)
OS and Service detection performed.
Please report any incorrect results at http://nmap.org/submit/
```

- Low-level way to identify operating systems, also try/use `nmap -A`
- Send probes and observe responses, lookup in table of known OS and responses
- Techniques known since at least: *ICMP Usage In Scanning* Version 3.0, Ofir Arkin, 2001

Portscan using Zenmap GUI



Zenmap included in the full Nmap package <https://nmap.org>

Note: Zenmap is based on old Python2! Use the Kali Kaboxer apt install zenmap-kbx



What happens now?

Think like a hacker

Recon phase – gather information reconnaissance

- Traceroute, Whois, DNS lookups
- Ping sweep, port scan
- OS detection – TCP/IP and banner grabbing
- Service scan – rpcinfo, netbios, ...
- telnet/netcat interact with services

Exercise



Now lets do the exercise

⚠ Nping check ports 10 min

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

Exercise



Now lets do the exercise

i Try pcap-diff 15 min

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

Exercise



Now lets do the exercise

⚠ Discover active systems ping sweep 10 min

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

Exercise



Now lets do the exercise

⚠ Execute nmap TCP and UDP port scan 20 min

which is number **19** in the exercise PDF.

Exercise

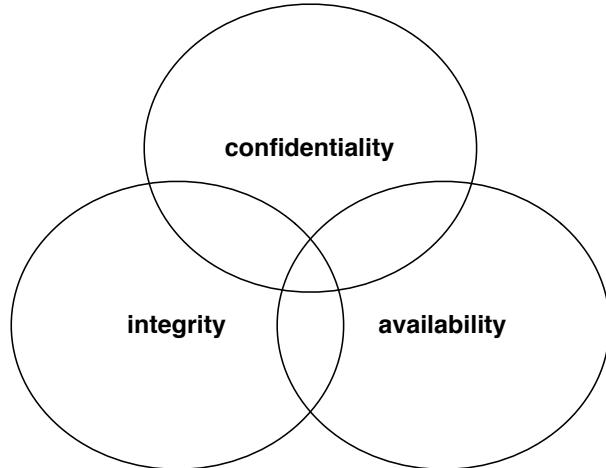


Now lets do the exercise

⚠ Perform nmap OS detection 10 min

which is number **20** in the exercise PDF.

Confidentiality Integrity Availability



We want to protect something

Confidentiality - data holdes hemmelige

Integrity - data ændres ikke uautoriseret

Availability - data og systemet er tilgængelige når de skal bruges



Unencrypted data protocols

Examples

- TFTP use UDP and is unencrypted
- TFTP still used for configuration files and firmwares
- FTP sends data in cleartext

USER username og

PASS password

- DNS sending unencrypted on UDP and TCP
Proposals for encrypted DNS over TCP and DNS over HTTPS being worked on

Stop using FTP on the internet! Use DNS over HTTPS (DoH) or DNS over TLS (DoT)

Person in the middle attacks



ARP spoofing, ICMP redirects, the classics

Used to be called Man in The Middle MiTM

- ICMP redirect
- ARP spoofing
- Wireless listening and spoofing higher levels like airpwn-ng <https://github.com/ICSec/airpwn-ng>

Usually aimed at unencrypted protocols

Today we only talk about getting the data, not how to perform higher level attacks

ICMP redirect



Routere understøtter ofte ICMP Redirect

Med ICMP Redirect kan man til en afsender fortælle en anden vej til destination

Den angivne vej kan være smartere eller mere effektiv

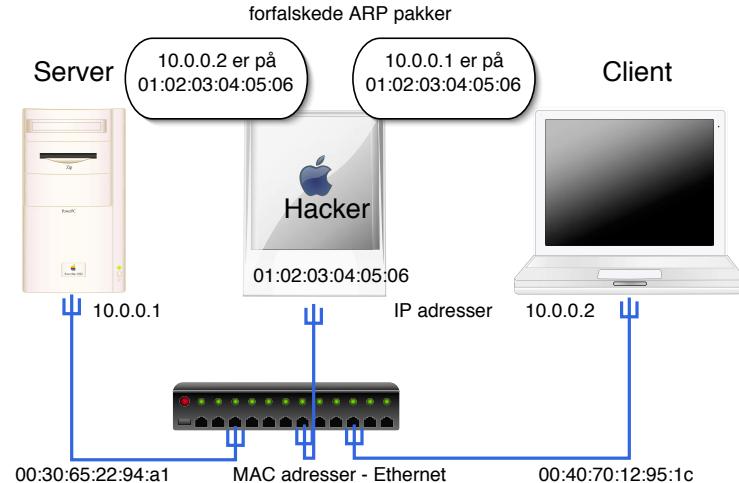
Det er desværre uheldigt, idet der ingen sikkerhed er

Idag bør man ikke lytte til ICMP redirects, ej heller generere dem

Det svarer til ARP spoofing, idet trafik omdiriges



Hvordan virker ARP spoofing?



Hackeren sender forfalskede ARP pakker til de to parter

De sender derefter pakkerne ud på Ethernet med hackerens MAC adresse som modtager - som får alle pakkerne

Forsvar mod ARP spoofing



Hvad kan man gøre?

låse MAC adresser til porte på switcher

låse MAC adresser til bestemte IP adresser

Efterfølgende administration!

Adskilte netværk - brug IEEE 802.1q VLANs

arpwatch er et godt bud - overvåger ARP

bruge protokoller som ikke er sårbare overfor opsamling

Lufthavns wifi

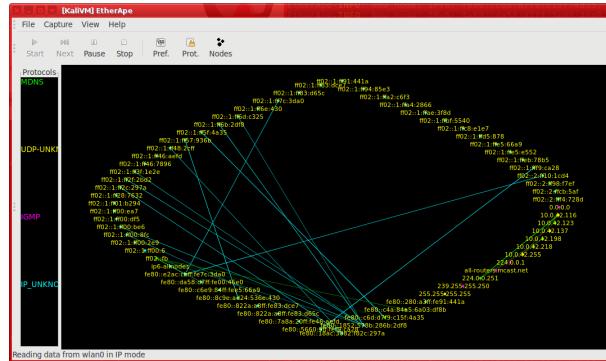


Åbne trådløse netværk er dejlige, vi bruger dem allesammen.

```
http://wifi.aal.dk/fs/customwebauth/login.html?  
switch_url=http://wifi.aal.dk/login.html&ap_mac=70:db:98:73:e5:a0&  
client_mac=30:10:b3:XX:YY:ZZ&wlan=AALfree&redirect=www.gstatic.com/generate_204
```

- Når du forbinder til netværket, bruger din enhed sin MAC adresse
- Denne indeholder en OUI som er den første halvdel af de 48-bit
- Dette ID er gemt i din enhed, fra fabrikken, kan sjældent ændres
- Alle i nærheden kan se denne MAC, og dermed din enheds unikke hardwareadresse.
- Kendere ved at man kan skifte sin MAC midlertidigt, og det gør telefoner ofte når de scanner efter netværk idag - hvis de overhovedet scanner
- Søg på macchanger <https://pkg.kali.org/pkg/macchanger>

Demo Attacks fun with nodes



EtherApe is a graphical network monitor for Unix modeled after etherman. Featuring link layer, IP and TCP modes, it displays network activity graphically. Hosts and links change in size with traffic. Color coded protocols display.

How do we find nodes to perform ARP spoofing?

The main page for the tool is: <https://etherape.sourceforge.io/>

Exercise



Now lets do the exercise

EtherApe 10 min

which is number **21** in the exercise PDF.

Exercise

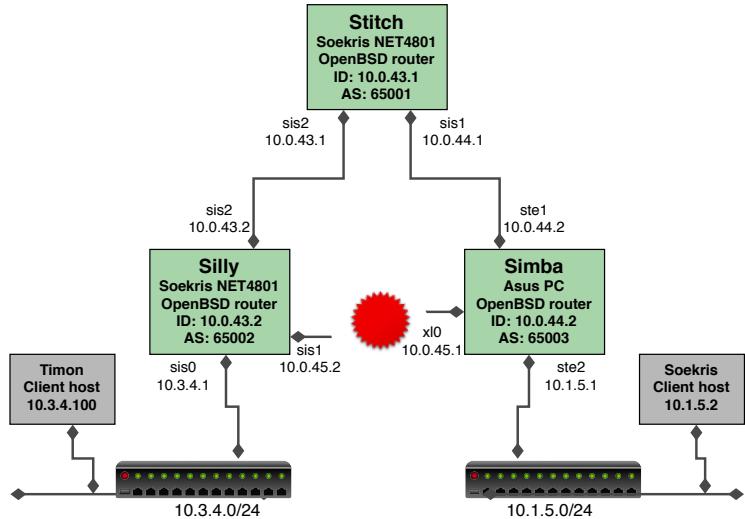


Now lets do the exercise

i ARP spoofing and ettercap 20min

which is number **22** in the exercise PDF.

Intro to routing protocols attacks



Networks grow and static configuration gets cumbersome, hard to maintain

Static routing does not scale, and we also require redundancy

Typical techs used RSTP on Layer 2 switching and Dynamic routing Layer 3 IP level

BGP intro



What is BGP Border Gateway Protocol

Dynamic routing protocol, BGPv4 used on whole internet

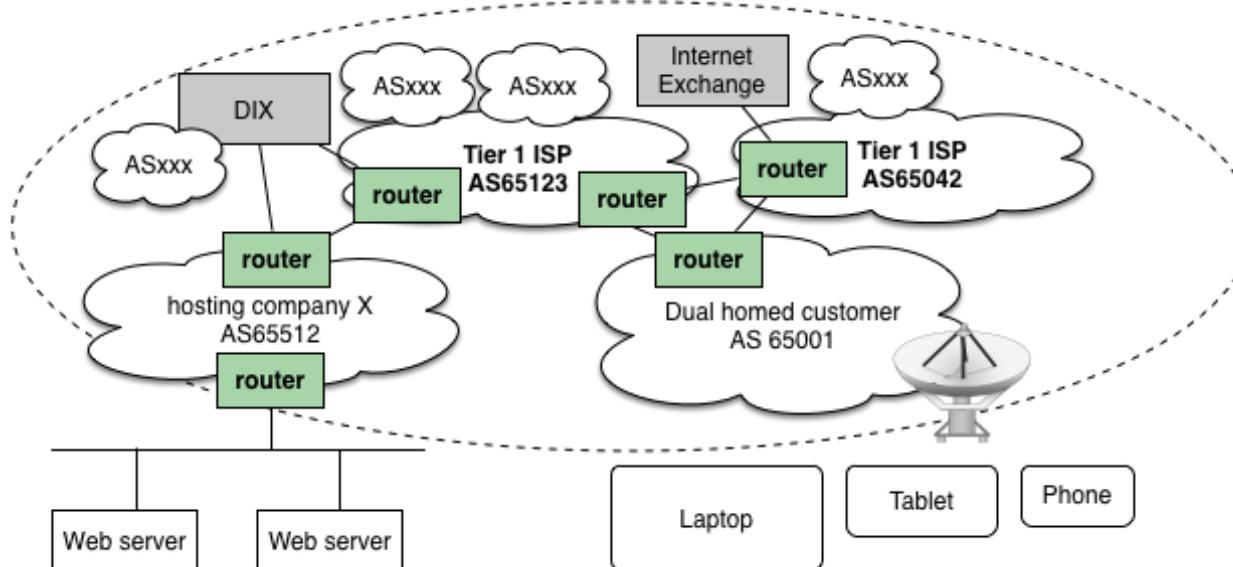
Networks identified using AS numbers ASNs

Autonomous System (AS) can be small or very big, world wide

BGP version 4 RFC-4271 uses TCP connections *peering*

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

Hosting and internet providers

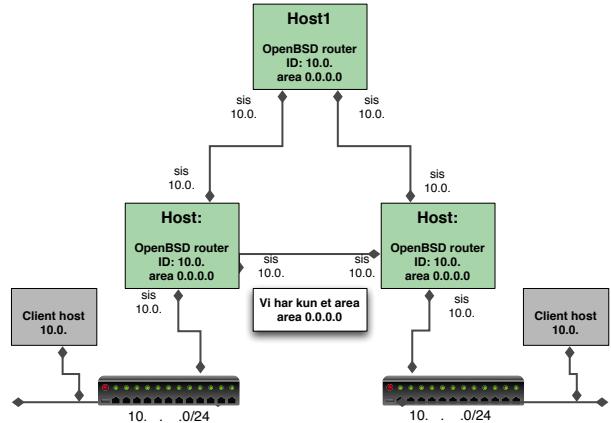


- BGP networks are used for all of the Internet
- New standards like Resource Public Key Infrastructure (RPKI) are underway

OSPF Open Shortest Path First



OpenOSPFD netværk



- Dynamic routing protocol for internal routing updates
- OSPF version 3 RFC-2740
- OSPF uses neither TCP or UDP, but protocol ID 89
- OSPF uses a metric/cost per link and calculates on each node
- http://en.wikipedia.org/wiki/Open_Shortest_Path_First

Network Security Threats



Low level and Network Layer Attacks

- "Yersinia is a network tool designed to take advantage of some weakness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems."
evil I2 tools - STP, CDP, DTP, DHCP, HSRP, IEEE 802.1Q, IEEE 802.1X, ISL, VTP
<https://github.com/tomac/yersinia>
- IP based creating strange fragments, overlapping, missing, SMALLL with fragroute/fragrouter
- LAND - same destination and source address
- THC-IPV6 - attacking the IPV6 protocol suite

Note: Evil repeats itself, like doing ARP poisoning across MPLS

Attackers are very creative!

Loki - ikke alt er layer 7



At the beginning LOKI was made to combine some stand-alone **command line tools**, like the `bgp_cli`, the `ospf_cli` or the `ldp_cli` and to give them a **user friendly, graphical interface**. In the meantime LOKI is more than just the combination of the single tools, it gave its modules the opportunity to base upon each other (like **combining ARP-spoofing from the ARP module with some man-in-the-middle actions, rewriting MPLS-labels for example**) and even inter operate with each other.

(bold by me)

<https://www.c0decafe.de/loki.html> Loki

<http://www.packetstan.com/2011/02/running-loki-on-backtrack-4-r2.html>

Resource Public Key Infrastructure RPKI



- 1997 - AS7007 mistakenly (re)announces 72,000+ routes (becomes the poster-child for route filtering).
- 2008 - ISP in Pakistan accidentally announces IP routes for YouTube by blackholing the video service internally to their network.
- 2017 - Russian ISP leaks 36 prefixes for payments services owned by Mastercard, Visa, and major banks.
- 2018 - BGP hijack of Amazon DNS to steal crypto currency.

Source: <https://blog.cloudflare.com/rpki/>

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

Authenticated routing protocols passwords, secrets etc.

Use RPKI



Routinator is RPKI Relying Party software, also known as an RPKI Validator. It is designed to have a small footprint and great portability.

Routinator connects to the Trust Anchors of the five Regional Internet Registries (RIRs) — APNIC, AFRINIC, ARIN, LACNIC and RIPE NCC — downloads all of the certificates and ROAs in their repositories and validates the signatures. It can feed the validated information to hardware routers supporting Route Origin Validation such as Juniper, Cisco and Nokia, as well as serving software solutions like BIRD and OpenBGPD. Alternatively, Routinator can output the validated data in a number of useful formats, such as CSV, JSON and RPSL.

Quote from <https://www.nlnetlabs.nl/projects/rpki/routinator/>

Update your records in the Whois system, read about RIPE here:

<https://www.ripe.net/manage-ips-and-asns/resource-management/certification>

The Spamhaus Don't Route Or Peer Lists

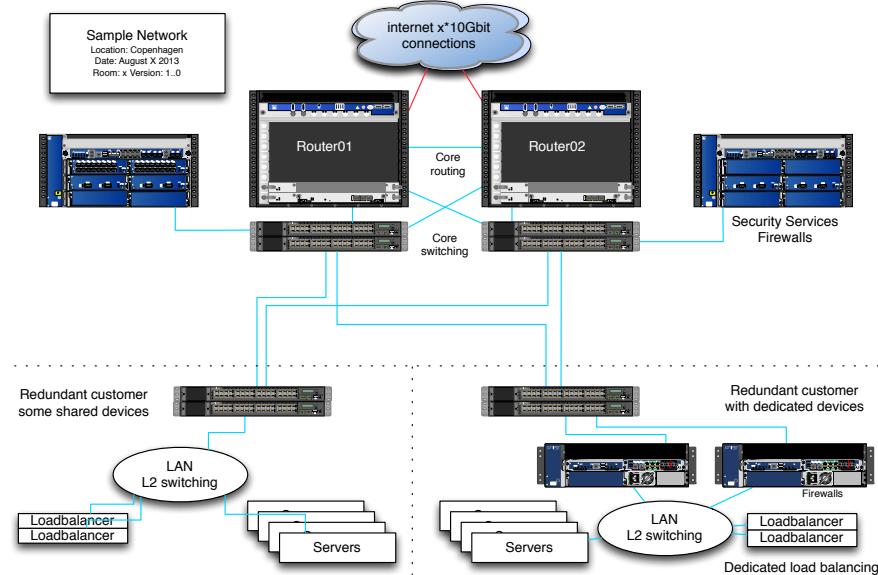


The Spamhaus Don't Route Or Peer Lists

DROP (Don't Route Or Peer) and EDROP are advisory "drop all traffic" lists, consisting of stolen 'hijacked' netblocks and netblocks controlled entirely by criminals and professional spammers. DROP and EDROP are a tiny subset of the SBL designed for use by firewalls and routing equipment.

<http://www.spamhaus.org/drop/>

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.

Availability and Network flooding attacks



- SYN flood is the most basic and very common on the internet towards 80/tcp and 443/tcp
- ICMP and UDP flooding are the next targets
- Supporting litterature is TCP Synfloods - an old yet current problem, and improving pf's response to it, Henning Brauer, BSDCan 2017
- All of them try to use up some resources
- Memory space in specific sections of the kernel, TCP state, firewalls state, number of concurrent sessions/connections
- interrupt processing of packets - packets per second
- CPU processing in firewalls, pps
- CPU processing in server software
- Bandwidth - megabits per second mbps

There are multiple resources about DDoS protection with more low level technical measures to implement at

<https://github.com/kramse/security-courses/tree/master/presentations/network/introduction-ddos-testing>

Simulating DDoS packets



A minimal introduction how to produce DDoS simulation traffic - usefull for testing their own infrastructures.

We will have a server connected on a switch with multiple 1Gbit port for attackers. Attackers will be connected through 1Gbit ports using USB Ethernet - we have loaners.

Work together to produce enough to take down this server!

WHILE attack is ongoing there will be both the possibility to monitor traffic, monitor port, and decide on changes to prevent the attacks from working.



Common DDoS attack types

We will work through common attack types, like:

- TCP SYN flooding
- TCP other flooding
- UDP flooding NTP, etc.
- ICMP flooding
- Misc - stranger attacks and illegal combinations of flags etc.

then we will discuss which changes to environment could be implemented.

You will go away from this with tools for producing packets, hping3 and some configurations for protecting - PF rules, switch rules, server firewall rules.



hping3 packet generator

```
usage: hping3 host [options]
-i --interval  wait (uX for X microseconds, for example -i u1000)
--fast        alias for -i u10000 (10 packets for second)
--faster      alias for -i u1000 (100 packets for second)
--flood       sent packets as fast as possible. Don't show replies.
```

...

hping3 is fully scriptable using the TCL language, and packets can be received and sent via a binary or string representation describing the packets.

- Hping3 packet generator is a very flexible tool to produce simulated DDoS traffic with specific characteristics
- Home page: <http://www.hping.org/hping3.html>
- Source repository <https://github.com/antirez/hping>

My primary DDoS testing tool, easy to get specific rate pps



t50 packet generator

```
root@cornerstone03:~# t50 -?
T50 Experimental Mixed Packet Injector Tool 5.4.1
Originally created by Nelson Brito <nbrito@sekure.org>
Maintained by Fernando Mercês <fernando@mentebinaria.com.br>
```

```
Usage: T50 <host> [/CIDR] [options]
```

Common Options:

```
--threshold NUM      Threshold of packets to send      (default 1000)
--flood              This option supersedes the 'threshold'
```

...

6. Running T50 with '--protocol T50' option, sends ALL protocols sequentially.

```
root@cornerstone03:~# t50 -? | wc -l
264
```

- T50 packet generator, another high speed packet generator can easily overload most firewalls by producing a randomized traffic with multiple protocols like IPsec, GRE, MIX
home page: <http://t50.sourceforge.net/resources.html>

Extremely fast and breaks most firewalls when flooding, easy 800k pps/400Mbps

Stress testing and DDoS



```
[Projects] Terminal - hlk@penguin01:
File Edit View Terminal Tabs Help
root@penguin01:/home/hlk/projects/MoonGen# ./build/MoonGen ./examples/pinguinping-02.lua 10.0.49.1 -a 10.1.2.3 -r 1000 -S -A -F -U -P -R

EAL: Detected 16 lcore(s)
EAL: No free hugepages reported in hugepages-1048576kB
EAL: Probing VFIO support...
EAL: PCI device 0000:01:00.0 on NUMA socket -1
EAL:   Invalid NUMA socket, default to 0
EAL:     probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:01:00.1 on NUMA socket -1
EAL:   Invalid NUMA socket, default to 0
EAL:     probe driver: 8086:10fb net_ixgbe

Device 0: 00:25:90:32:9f:F2 (Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection)
Device 1: 00:25:90:32:9f:F3 (Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection)
PMD: ixgbe_dev_link_status_print(): Port 0: Link Up - speed 0 Mbps - half-duplex

TCP mode get TCP packet
ETH 00:25:90:32:9f:F2 > 00:00:00:00:00:00 type 0x0800 [IP4]
IP4 10.1.2.3 > 10.0.49.1: 4 ihl 5 tos 0 len 46 id 0 flags 0 frag 0 ttl 64 proto 0x06 (TCP) cksum 0x0000 [-]
TCP 52049 > 80 seq 1 ack 0 offset 0x5 reserved 0x00 flags 0x3f [URG|ACK|PSH|RST|SYN|FIN] win 10 cksum 0x0000 urg 0 []
    0000 0000 0000 0025 0032 0ff2 0000 4500
    002e 0000 0000 4006 0000 0a01 0203 0a00
    3101 cb51 0050 0000 0001 0000 0000 503f
    000a 0000 0000 0000 0000 0000 0000 0000

[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 994 Mbit/s (1304 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
[Device: id=0] TX: 1.95 Mpps, 1000 Mbit/s (1312 Mbit/s with framing)
```

- PenguinPing packet generator, my high speed packet generator home page: <https://pinguinping.org>
- First versions are only about 230 lines of Lua code and implement basic command line to replace hping3
- Built on top of MoonGen/libmoon <https://github.com/emmericp/MoonGen>

Extremely fast and allows easy customization

Process: monitor, attack, break, repeat



- Pre-test: Monitoring setup - from multiple points
- Pre-test: Perform full Nmap scan of network and ports
- Start small, run with delays between packets
- Turn up until it breaks, decrease delay - until using --flood
- Monitor speed of attack on your router interface pps/bandwidth
- Give it maximum speed

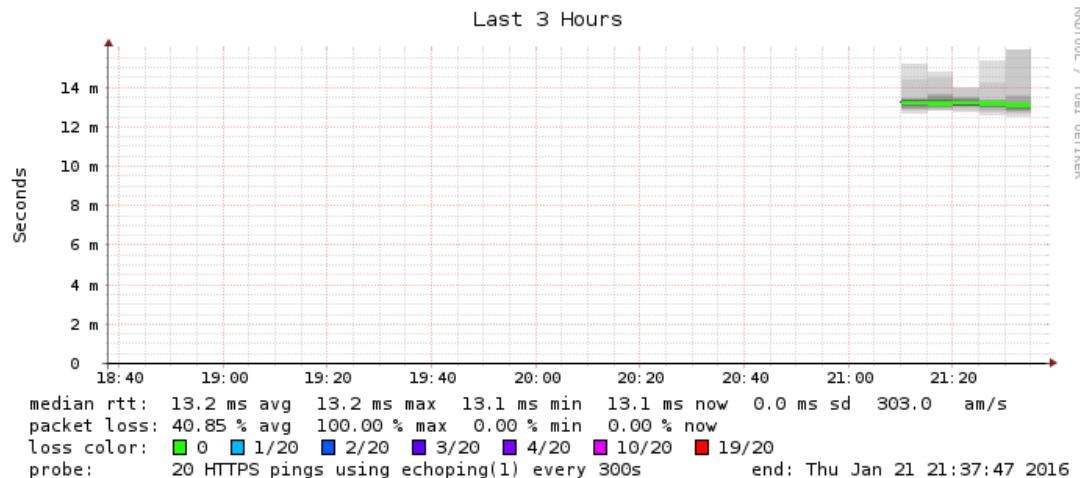
```
hping3 --flood -1 and hping3 --flood -2
```
- Have a common chat with network operators/customer to talk about symptoms and things observed
- Any information resulting from testing is good information

Ohh we lost our VPN into the environment, ohh the fw console is dead

Before testing: Smokeping



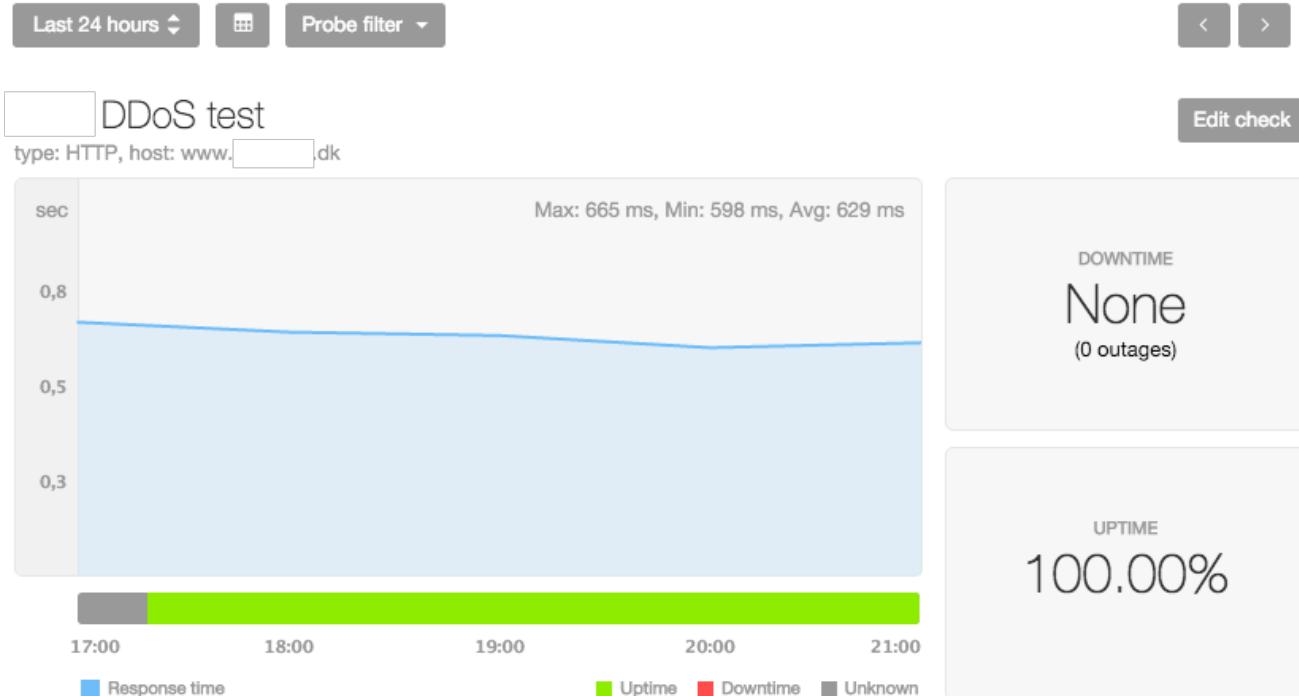
HTTPS check www. .26



Before DDoS testing use Smokeping software



Before testing: Pingdom



Another external monitoring from Pingdom.com



Running full port scan on network

```
# export CUST_NET="192.0.2.0/24"  
# nmap -p 1-65535 -A -oA full-scan $CUST_NET
```

Performs a full port scan of the network, all ports

Saves output in "all formats" normal, XML, and grepable formats

Goal is to enumerate the ports that are allowed through the network.

Note: This command is pretty harmless, if something dies, then it is
vulnerable to normal traffic - and should be fixed!



Running Attacks with hping3

```
# export CUST_IP=192.0.2.1
# date;time hping3 -q -c 1000000 -i u60 -S -p 80 $CUST_IP

# date;time hping3 -q -c 1000000 -i u60 -S -p 80 $CUST_IP
Thu Jan 21 22:37:06 CET 2016
HPING 192.0.2.1 (eth0 192.0.2.1): S set, 40 headers + 0 data bytes

--- 192.0.2.1 hping statistic ---
1000000 packets transmitted, 999996 packets received, 1% packet loss
round-trip min/avg/max = 0.9/7.0/1005.5 ms

real    1m7.438s
user    0m1.200s
sys     0m5.444s
```

Dont forget to do a killall hping3 when done ☺

Recommendations During Test



Run each test for at least 5 minutes, or even 15 minutes

Some attacks require some build-up before resource run out

Take note of any change in response, higher latency, lost probes

If you see a change, then re-test using the same parameters, or a little less first

We want to know the approximate level where it breaks

If you want to change environment, then wait until all scenarios tested

Exercise



Now lets do the exercise

i TCP SYN flooding 30min

which is number **23** in the exercise PDF.

Exercise booklet contains some bonus exercises, feel free to try them at home



Comparable to real DDoS?

Tools are simple and widely available but are they actually producing same result as high-powered and advanced criminal botnets. We can confirm that the attack delivered in this test is, in fact, producing the traffic patterns very close to criminal attacks in real-life scenarios.

- We can also monitor logs when running a single test-case
- Gain knowledge about supporting infrastructure
- Can your syslog infrastructure handle 800.000 events in < 1 hour?

Running the tools



A basic test would be:

- TCP SYN flooding
- TCP other flags, PUSH-ACK, RST, ACK, FIN
- ICMP flooding
- UDP flooding
- Spoofed packets src=dst=target ☺
- Small fragments
- Bad fragment offset
- Bad checksum
- Be creative
- Mixed packets - like t50 --protocol T50
- Perhaps esoteric or unused protocols, GRE, IPSec



Test-cases / Scenarios

The minimal run contains at least these:

- SYN flood: hping3 -q -c 1000000 -i u60 -S -p 80 \$CUST_IP &
- SYN+ACK: hping3 -q -c 1000000 -i u60 -S -A -p 80 \$CUST_IP &
- ICMP flood: hping3 -q -c --flood -1 \$CUST_IP &
- UDP flood: hping3 -q -c --flood -1 \$CUST_IP &

Vary the speed using the packet interval -i u60 up/down

Use flooding with caution, runs max speeeeeeeeeeed ☺

TCP testing use a port which is allowed through the network, often 80/443

Focus on attacks which are hard to block, example TCP SYN must be allowed in

Also if you found devices like routers in front of environment

```
hping3 -q -c 1000000 -i u60 -S -p 22 $ROUTER_IP
```

```
hping3 -q -c 1000000 -i u60 -S -p 179 $ROUTER_IP
```



Test-cases / Scenarios, continued Spoof Source

Spoofed packets src=dst=target ☺

Flooding with spoofed packet source, within customer range

-a --spoof hostname

Use this option in order to set a fake IP source address, this option ensures that target will not gain your real address.

```
hping3 -q --flood -p 80 -S -a $CUST_IP $CUST_IP
```

Preferably using a test-case you know fails, to see effect

Still amazed how often this works

BCP38 anyone!



Test-cases / Scenarios, continued Small Fragments

Using the built-in option -f for hping

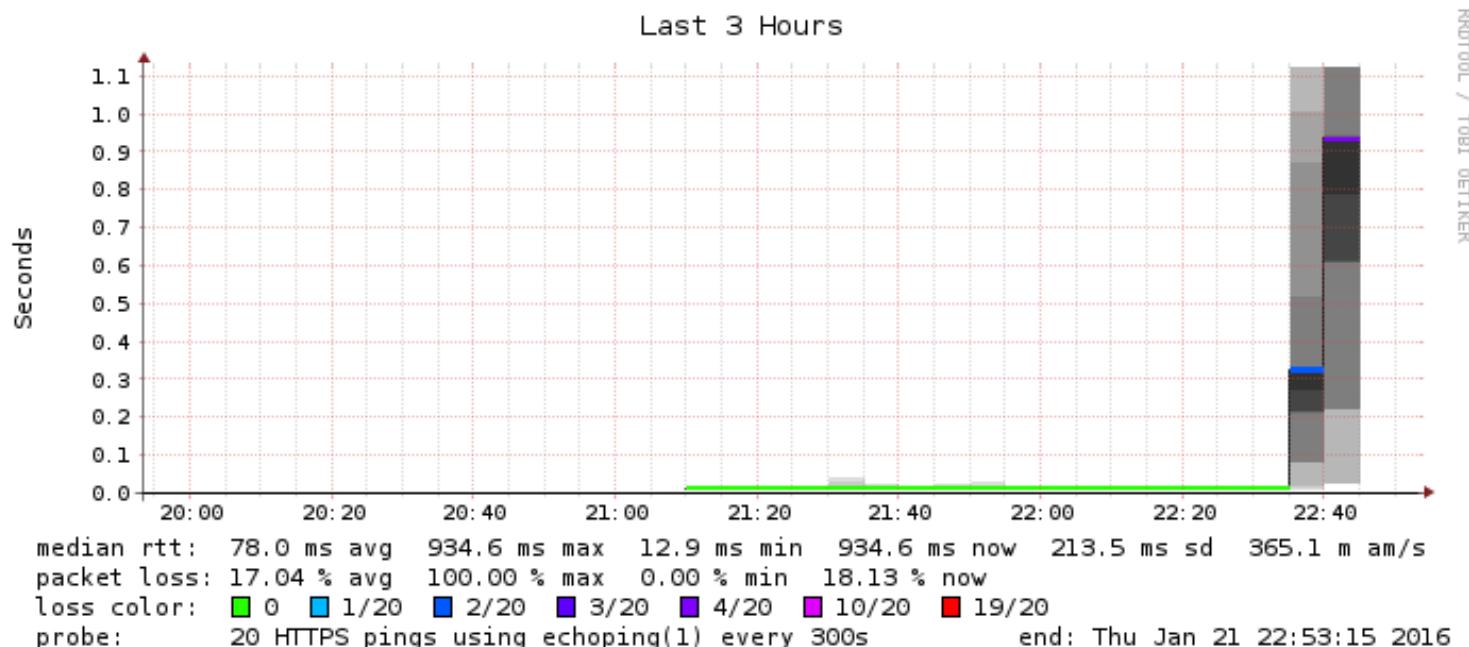
-f --frag

Split packets in more fragments, this may be useful in order to test IP stacks fragmentation performance and to test if some packet filter is so weak that can be passed using tiny fragments (anachronistic). Default '**virtual mtu' is 16 bytes.** see also --mtu option.

hping3 -q --flood -p 80 -S -f \$CUST_IP

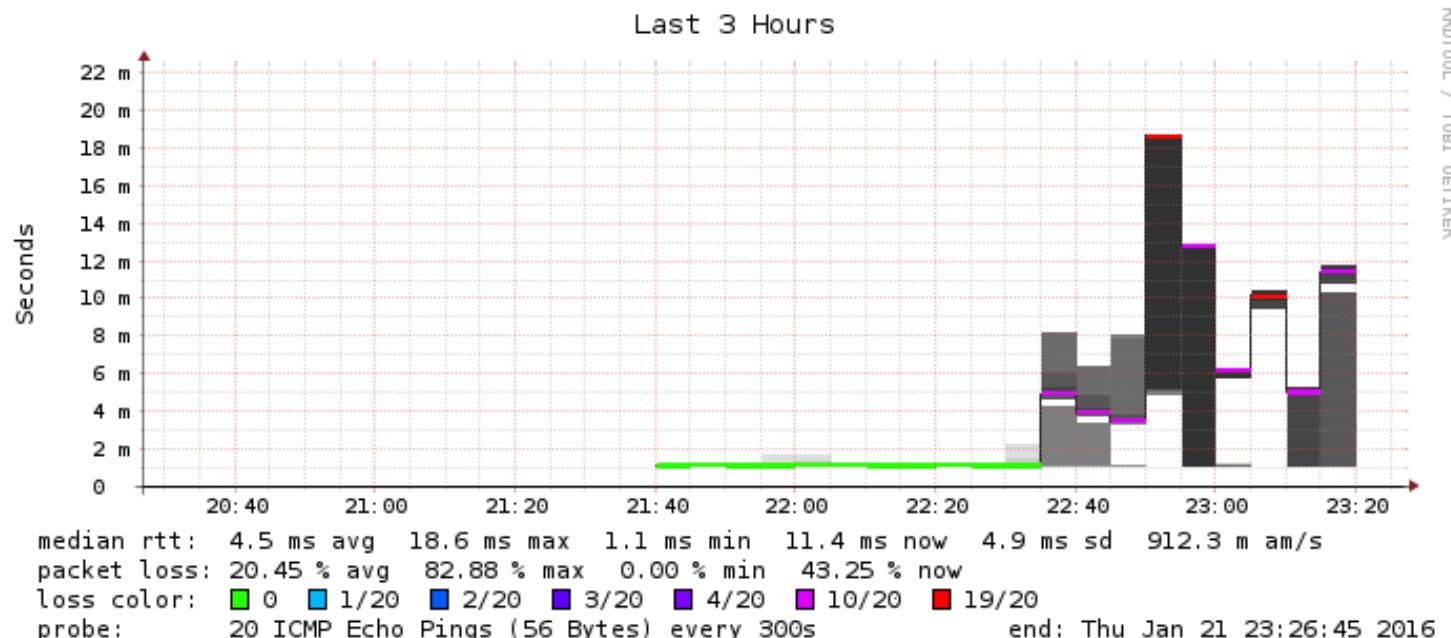
Similar process with bad checksum and Bad fragment offset

Rocky Horror Picture Show - 1



Really does it break from 50.000 pps SYN attack?

Rocky Horror Picture Show - 2

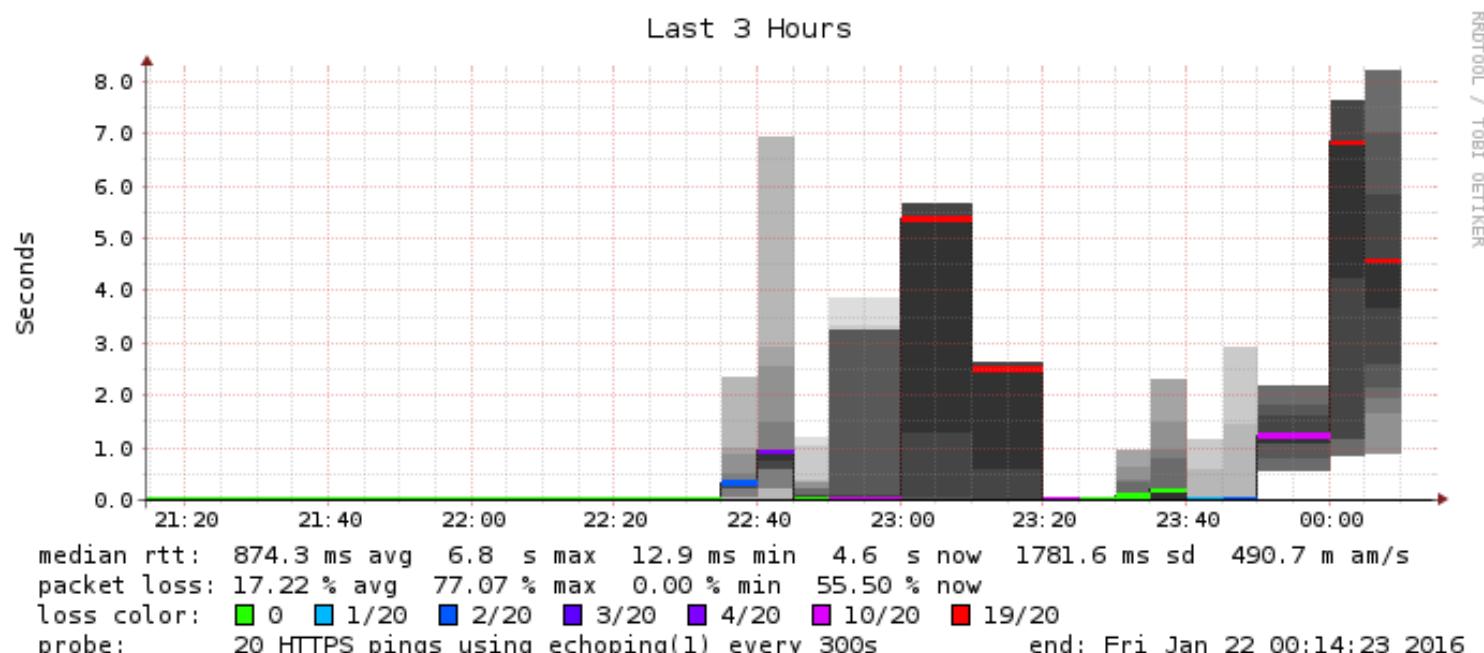


Oh no 500.000 pps UDP attacks work?

Rocky Horror Picture Show - 3



Oh no spoofing attacks work?





Improvements seen after testing

Turning off unneeded features - free up resources

Tuning sessions, max sessions src / dst

Tuning firewalls, max sessions in half-open state, enabling services

Tuning network, drop spoofed src from inside net ☺

Tuning network, can follow logs, manage network during attacks

...

And organisation has better understanding of DDoS challenges

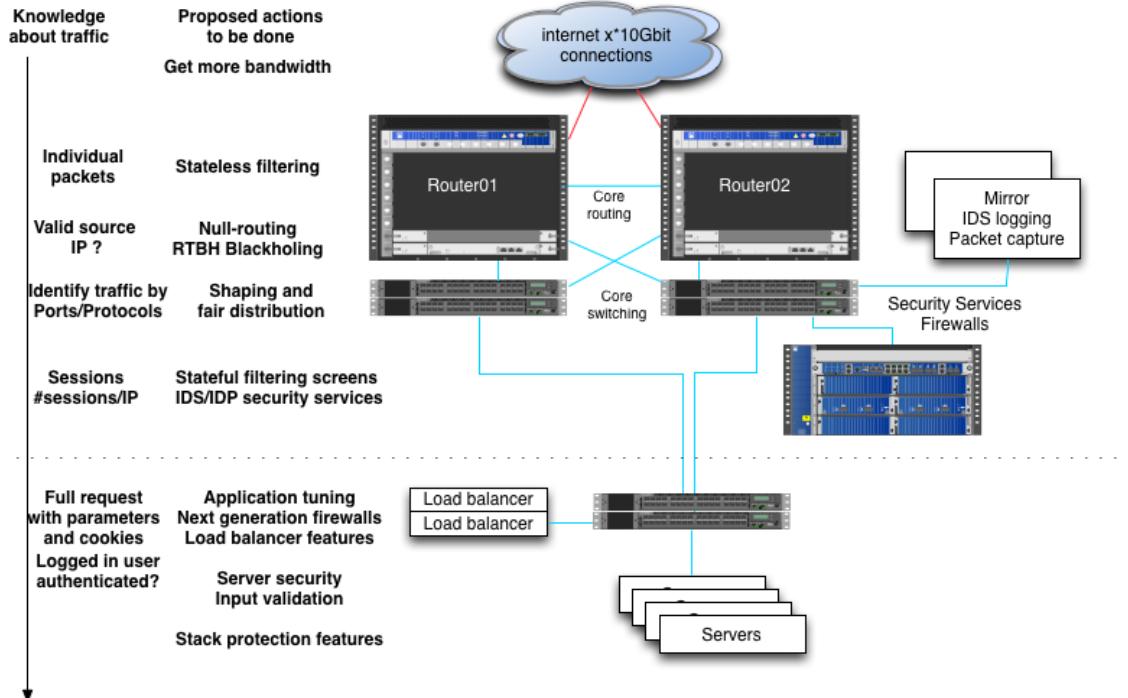
Including vendors, firewall consultants, ISPs etc.

After tuning of **existing devices/network** improves results 10-100 times

Conclusion DDoS and network attacks



You really should try testing
Investigate your existing devices
all of them, RTFM, upgrade firmware
Choose which devices does which
part - discard early to free resources
for later devices to dig deeper



And dont forget that DDoS testing is as much a firedrill for the organisation

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools