Welcome to

Simulating DDoS Attacks

breaking the firewall infrastructure

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

simulating-ddos-workshop.tex in the repo security-courses

Note: this workshop has run a couple of times, more or less the same.

Contact information

- Henrik Kramselund Jereminsen, internet samurai mostly networks and infosec
- Independent network and security consultant
- 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

What is pentest

> A penetration test, colloquially known as a pen test, pentest or ethical hacking, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system;[1][2] this is not to be confused with a vulnerability assessment.[3] The test is performed to identify weaknesses (also referred to as vulnerabilities), including the potential for unauthorized parties to gain access to the system's features and data,[4][5] as well as strengths,[6] enabling a full risk assessment to be completed.

Source: quote from `https://en.wikipedia.org/wiki/Penetration_test`

Penetration testing is a simulation, with good intentions

People around the world constantly *test your defenses*

Often better to test at planned times

Goal

Don't Panic!

How to create DDoS simulations, tools and process

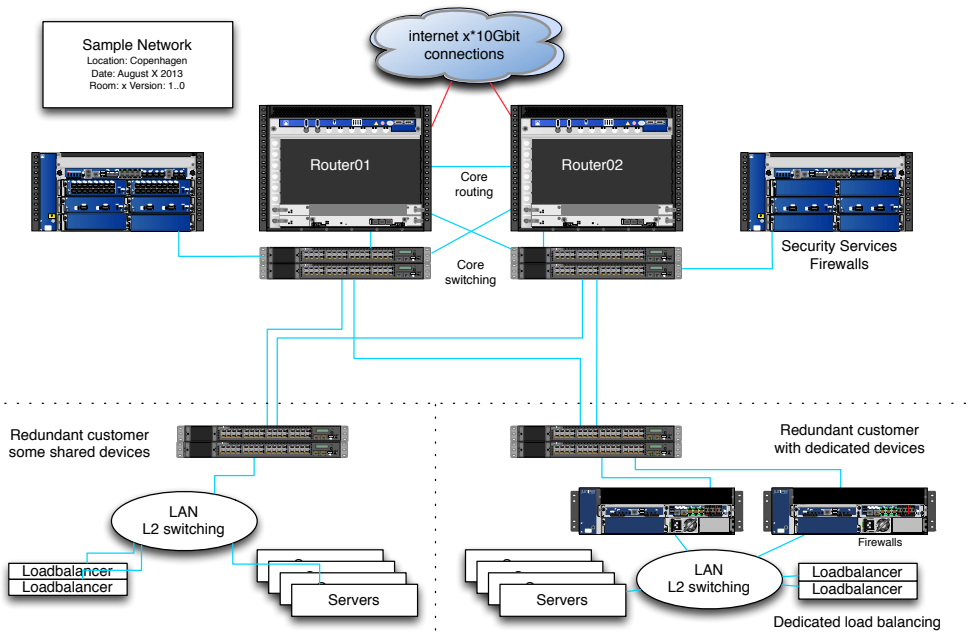
Some actual experience with doing this

Evaluate how good is this, value

I use and recommend Kali Linux as the base for this

If you like, try joining the fun - we will send packets!

Networks today are complex



Conclusion: Do as much as possible with your existing devices

Tuning and using features like stateless router filters works wonders

Kali Linux the new backtrack

Kali `http://www.kali.org/`

Use the documentation on Kali website for installation and management of the operating system

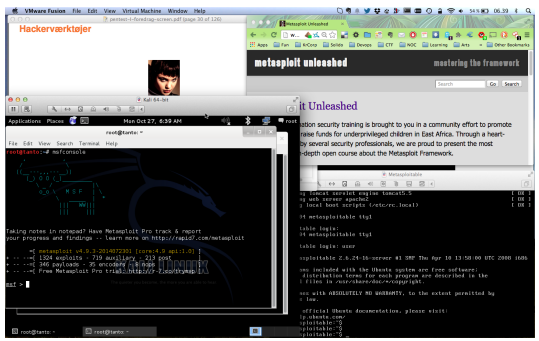
Use the web sites belonging to projects for documentation about tools,
like `https://nmap.org/` and `http://www.hping.org/`

Testing network the legal issues

Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 1 år og 6 måneder straffes den, der uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et informationssystem.

- Danish law about hacking
- Please check with your legal department, or be careful
- We **always** contact network between us and the network to be tested
- Be good netizens

Hackerlab setup



- I recommend getting a hackerlab running on your laptop
- Hardware: modern laptop which has CPU virtualization
  Dont forget to check BIOS settings for virtualization
- Software: your favorite OS: Windows, Mac, Linux
- Virtualization software: VMware, Virtual box, HyperV
- Hacker software: Kali as a Virtual Machine `https://www.kali.org/`
- Tools – we will use two packet generators today hping3 and t50

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 charateristics
- 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`

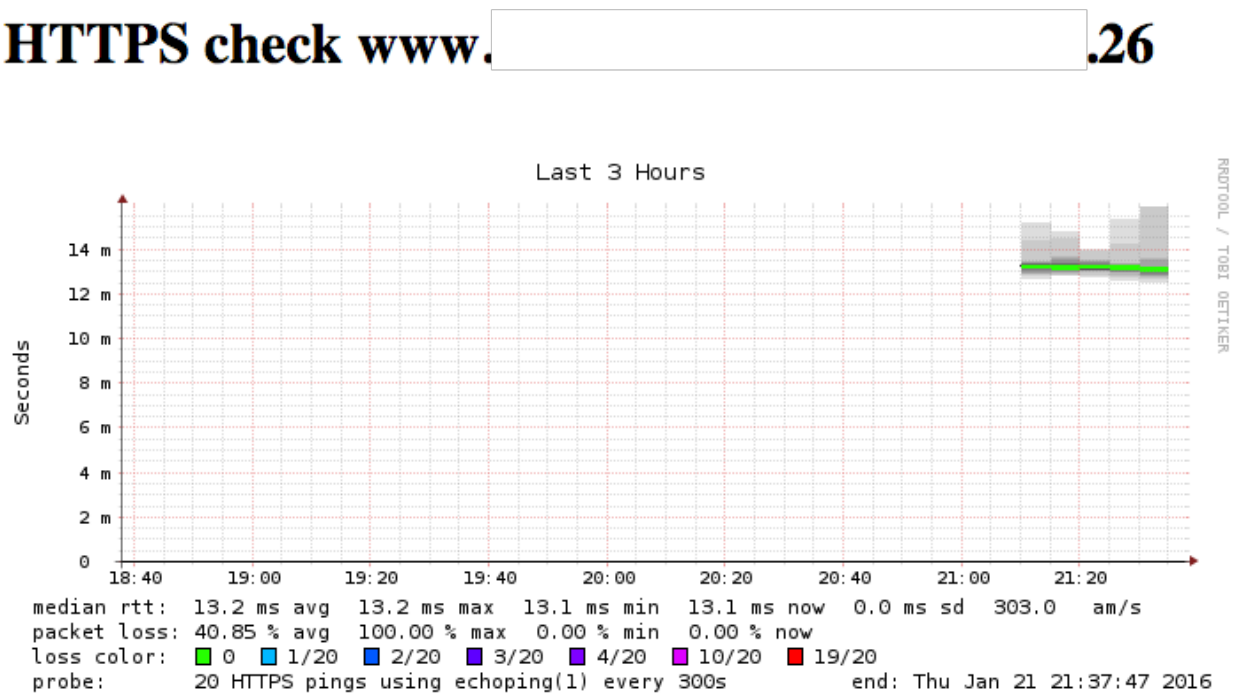
Extremely fast and breaks most firewalls when flooding, easy 800k pps/400Mbps

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

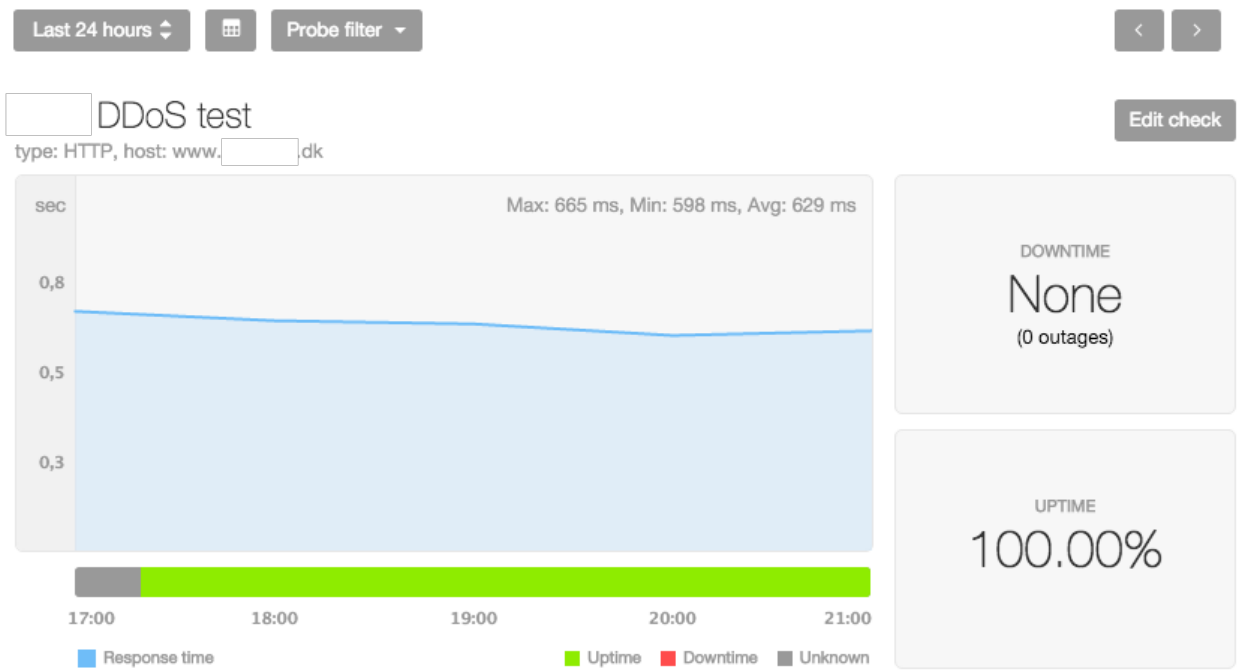
Ohh we lost our VPN into the environment, ohh the fw console is dead

Before testing: Smokeping



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!

Pro tip: use an environment variable to hold IP addresses. Then you won't suddenly mistype and start to attack 10.0.4.0/24 when you wanted to attack 10.0.42.0/24.

Running Attacks with hping3

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

Expected output:

```
# 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 are tested
- Keep a log handy, write notes and start the session with `script ddos-date-customer.log`
- Check once in a while if you have some process running, using `ps auxw | grep hping3`
- Run multiple instances of the tools. One process might generate 800.000 pps, while two may double it. Though 10 processes might not be 10 times exactly

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?

Main difference are that attackers are free to switch attack types and mix them. While we try specifically to keep using one type, to see the worst and which ones that hurt the most.

I also start at the bottom, and work my way up – while an attacker may begin attacking HTTP/HTTPS directly.

Pop quiz: Experiences from testing

How much bandwidth can big danish companies handle?

- A) 10-100Mbps
- B) 100Mbps -1Gbit
- C) Up to 5Gbit easily

How much abuse in pps can big danish companies handle?

- A) 10.000 - 50.000 pps
- B) 50 - 500k pps
- C) Up to 5 million pps

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 1000000 --flood -1 $CUST_IP &`
- UDP flood: `hping3 -q -c 1000000 --flood -2 $CUST_IP &`

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

Use flooding with caution, runs max speeeeeeeeeeeed ☺

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

Test-cases / Scenarios, continued Small Fragments

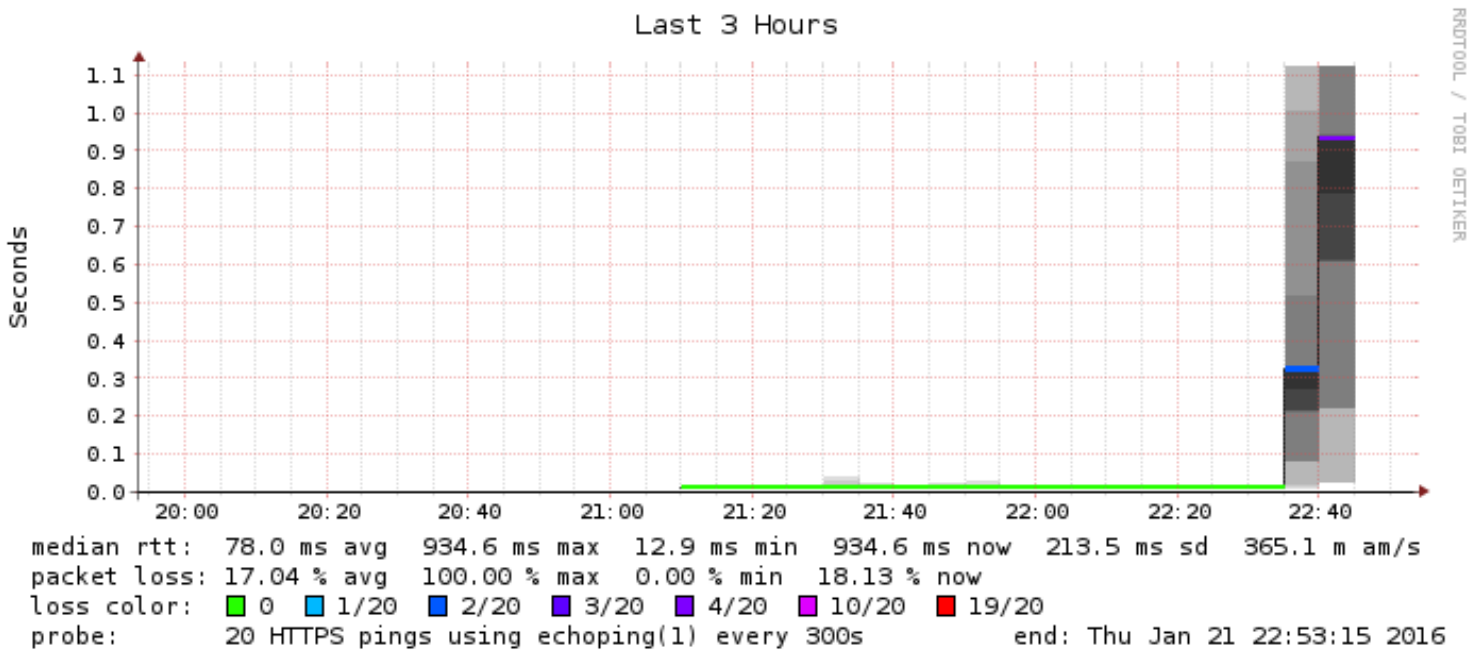
Using the built-in option -f for hping

```
-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
```

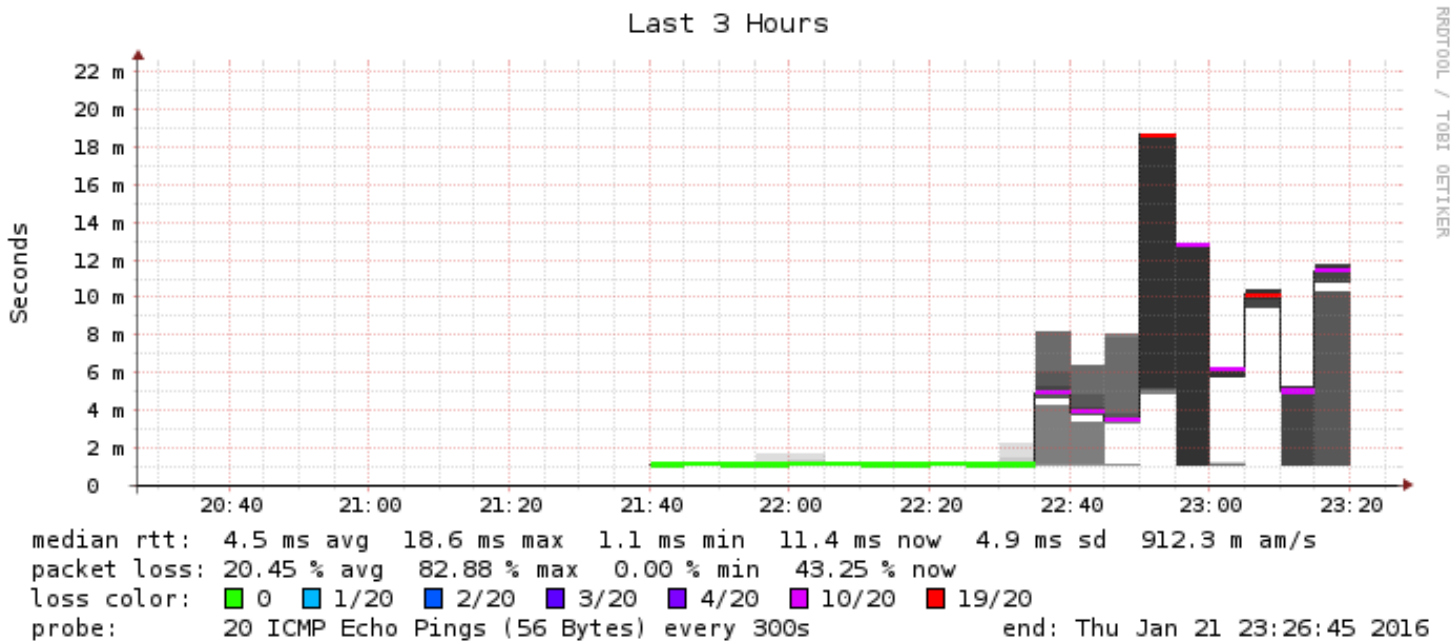
Similar process with bad checksum and Bad fragment offset

Rocky Horror Picture Show - 1



Really does it break from 50.000 pps SYN attack?

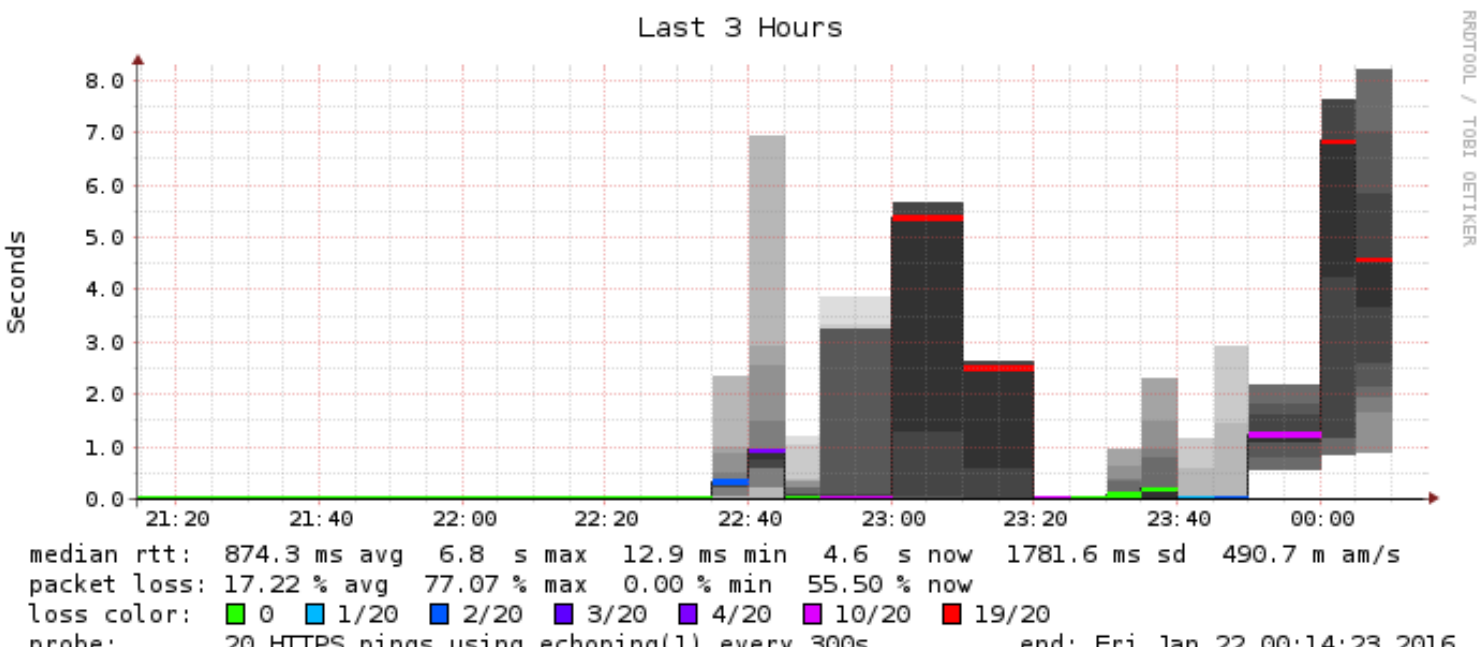
Rocky Horror Picture Show - 2



Oh no 500.000 pps UDP attacks work?

Rocky Horror Picture Show - 3

Oh no spoofing attacks work?



Experiences from testing

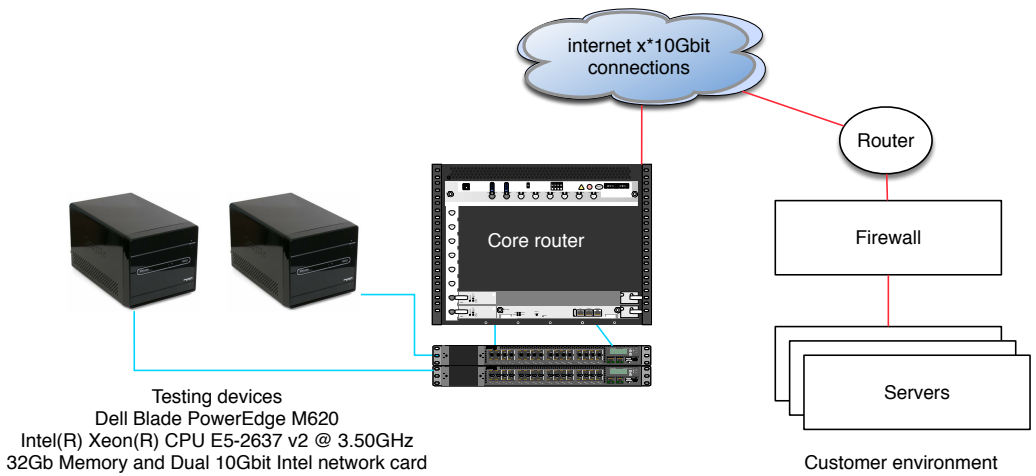
How much bandwidth can big danish companies handle!
100Mbps -1Gbit

How much abuse in pps can big danish companies handle!
B) **50.000 - 500k pps** TCP attacks
B) **500.000 - 1mill pps** UDP or ICMP attacks
Ohhh and often we can spoof using their addresses in the first test

Even the DDoS protection services are a bit too small, can handle perhaps only 10G and also multiple times admins lost access to network, VPN, log overflow etc.

Note: attackers can send full 10Gbit 14mill pps from Core i7 with 3 cores ...

Demo time



I will show the setup on my laptop while doing DDoS testing

Setup terminals: Kali and router

Run some tests against my target

Improvements seen after testing

Turning off unneeded features - free up resources

Tuning sesions, 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

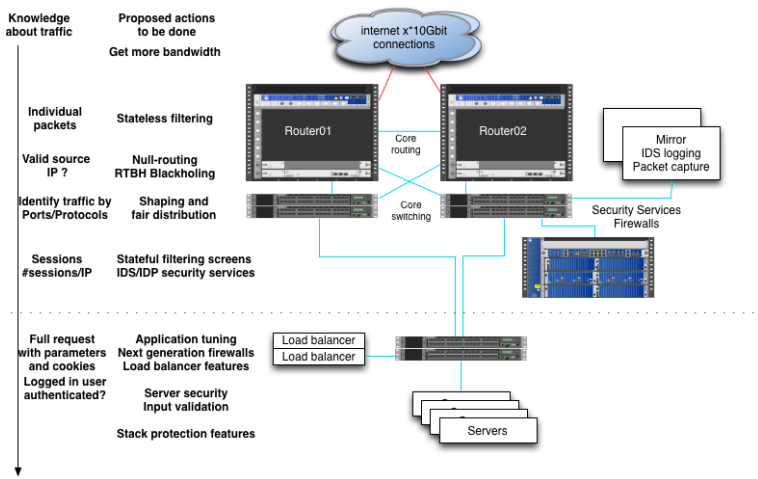
...

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



.

- 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

Questions?

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

You are always welcome to send me questions later via email

Email: hkj@zencurity.dk Mobile: +45 2026 6000

Extras if needed or questions arise

Preventing spoofed packets

Check your filtering see

- BCP38 is RFC2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing
  There is a whole wiki dedicated to this: `http://www.bcp38.info/index.php/Main_Page`
- `https://en.wikipedia.org/wiki/Ingress_filtering`
- `https://en.wikipedia.org/wiki/Reverse-path_forwarding`
- `https://www.openbsd.org/faq/pf/filter.html#antispoof`

Good MANRS

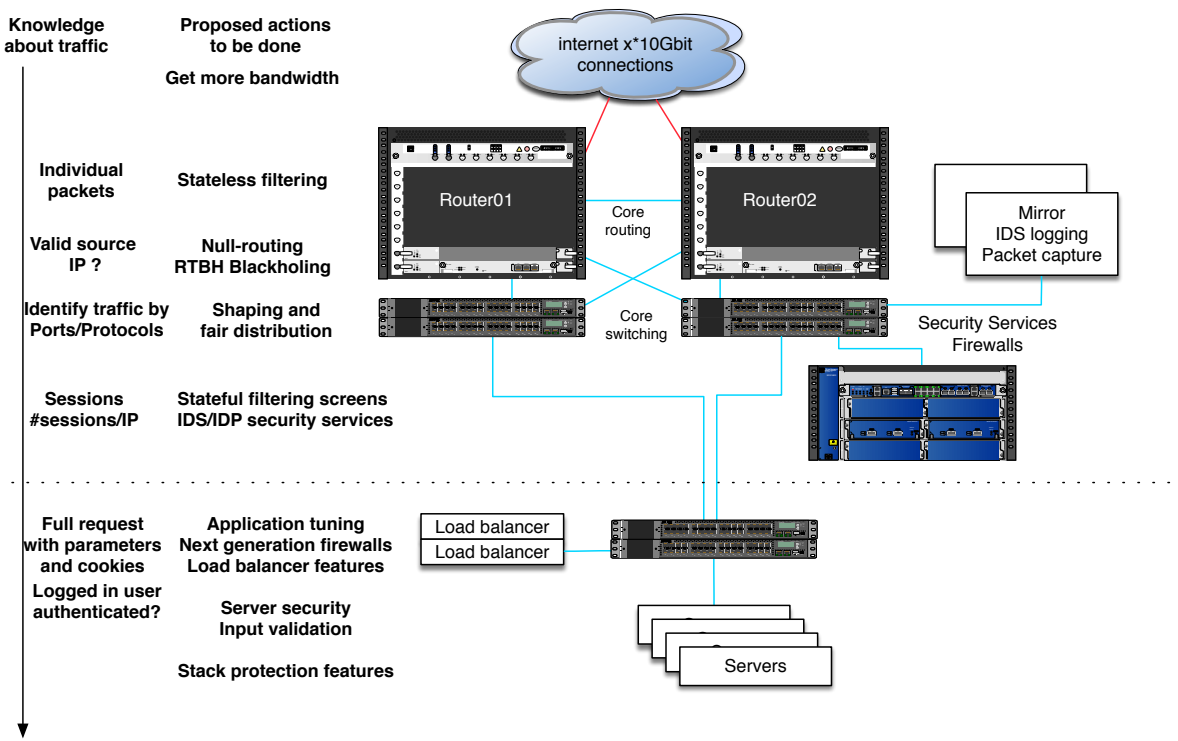
Also if you are a network operator, look into RPKI and MANRS

> 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.

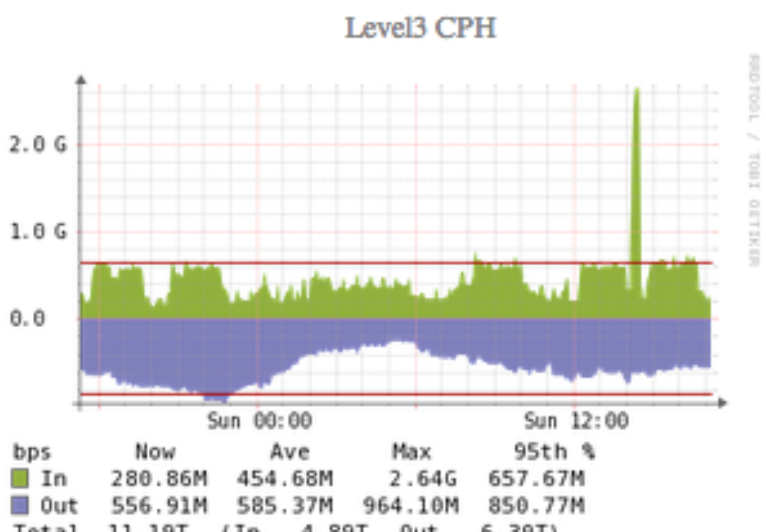
`https://www.manrs.org/`
and read this
`https://www.manrs.org/wp-content/uploads/2018/09/MANRS_PDF_Sep2016.pdf`

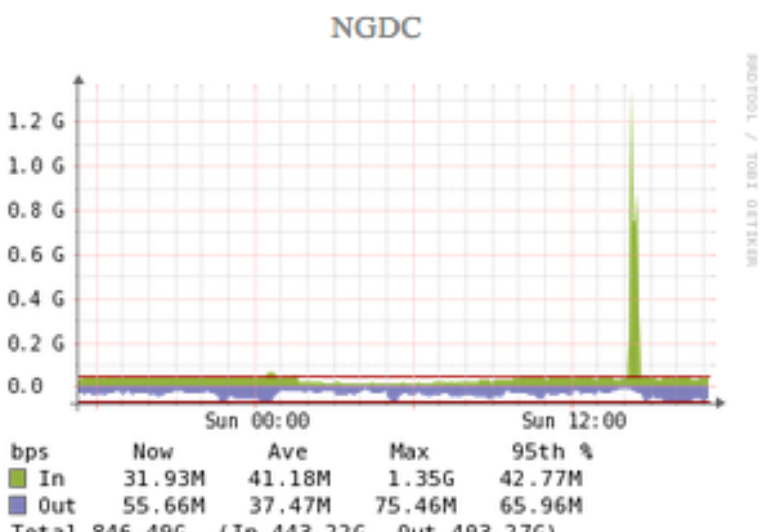
RPKI helps to secure routing: `https://en.wikipedia.org/wiki/Resource_Public_Key_Infrastructure`

Defense in depth - multiple layers of security



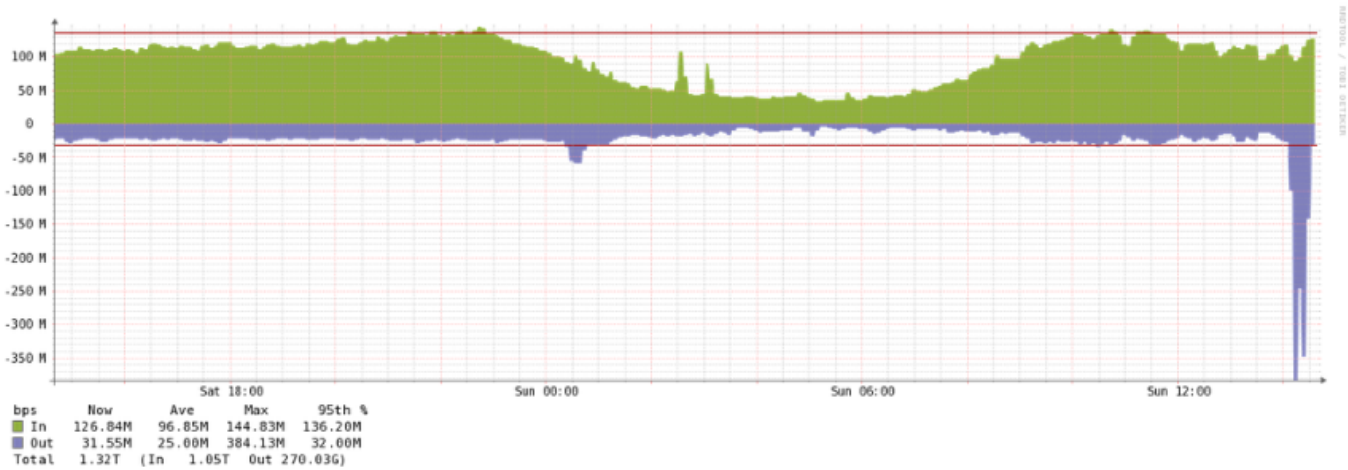
DDoS traffic before filtering





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

Stateless firewall filter throw stuff away

```
hlk@MX-CPH-02> show configuration firewall filter all | no-more
/* This is a static sample, perhaps better to use BGP flowspec and RTBH */
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 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 {
        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