Welcome to

10. Network Attacks

KEA Kompetence VF1 Software Security

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

Slides are available as PDF, kramse@Codeberg

10-network-attacks.tex in the repo security-courses

Plan for today

Subjects

- Auditing Application Protocols
- Example protocols and vulnerabilities
- Abstract Syntax Notation (ASN.1) problems
- Domain Name System (DNS) problems
- Firewalls and related issues
- Network Security Monitoring

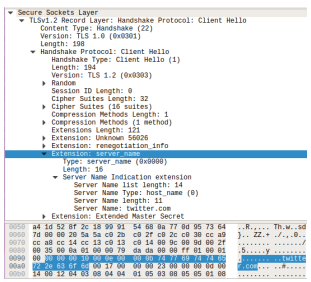
Exercises

- Nmap and SYN flooding exercises

Reading Summary

Browse: *TCP Synfloods - an old yet current problem, and improving pf's response to it* Henning Brauer, BSDCan 2017 http://quigon.bsws.de/papers/2017/bsdcan/

Goals: Introduction to Auditing Application Protocols



Often you dont need to audit the whole protocol in detail

Sometimes people can't tell which protocols, ports and services they use ...

And you need to configure a firewall/network filter

Picture: Wireshark with TLS SNI, recent Exim CVE-2019-15846 was SNI parsing

Plus SYN flooding, and Denial of Service

Goals today: Networking with some pentesting

What is pentest

> A penetration test, informally pen test, is an attack on a computer system that looks for security weaknesses, potentially gaining access to the computer's features and data.[1][2]

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

Penetration testing is a simulation, with good intentions

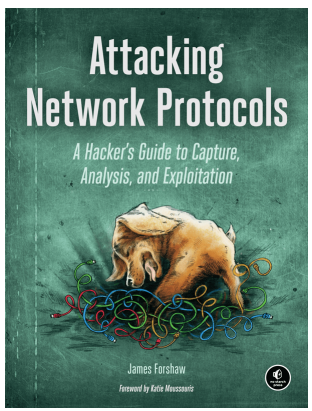
People around the world constantly *test your defenses*

Often better to test at planned times

How to create DDoS simulations, tools and process

I use and recommend Kali Linux as the base for this

Reversing and Attacking Network Protocols



A method with lots detail can be found in the book,
Attacking Network Protocols A Hacker's Guide to Capture, Analysis, and Exploitation
by James Forshaw December 2017, 336 pp. ISBN-13: 9781593277505

https://nostarch.com/networkprotocols

Auditing Application Protocols

- Collect documentation
- Identify Elements of Unknown Protocols
- Use packet sniffers, tcpdump and Wireshark
- Initiate the Connection Several Times
- Replay traffic, can sometimes replay even encrypted traffic, see wireless WEP attacks

Note: We investigate protocols, so we can see what is sent, so we can design *payloads* which create problems for implementations - applications

Reverse Engineer Applications

```
(gdb) disas main
Dump of assembler code for function main:
   0x0000000000000580 <+0>: lea    0x1ed(%rip),%rdi        # 0x774
   0x0000000000000587 <+7>: sub    $0x8,%rsp
   0x000000000000058b <+11>: mov    $0x7fff,%esi
   0x0000000000000590 <+16>: xor    %eax,%eax
   0x0000000000000592 <+18>: callq  0x560 <printf@plt>
   0x0000000000000597 <+23>: lea    0x1ed(%rip),%rdi        # 0x78b
   0x000000000000059e <+30>: mov    $0xffff8000,%esi
   0x00000000000005a3 <+35>: xor    %eax,%eax
   0x00000000000005a5 <+37>: callq  0x560 <printf@plt>
   0x00000000000005aa <+42>: xor    %eax,%eax
   0x00000000000005ac <+44>: add    $0x8,%rsp
   0x00000000000005b0 <+48>: retq
End of assembler dump.
```

- It is possible to debug, disassemble and reverse engineer applications
- Calling socket functions, seeing structs, data types etc.
- Examine strings: HTTP, FTP, SMTP etc. all uses semi-english words GET, EHLO, PASS

Special values

- Examine special values
- What are the defined/used values
- What happens if this is changed? Do they cover values outside of the used ranges? Case/switch constructs
- Use trace functions in the operating system, can capture, analyze and replay sometimes

Buffer Overflow when receiving

- When you see data enter the application, identify functions
- Consider if they use dangerous functions, strcpy and friends
- How much space is available, allocated etc.
- Basic stuff and similar across applications

- Repeat everything we learned about string processing, integeroverflows/underflows etc. Just from the network
- Often trying to abuse will lead to denial of service

- If some rock solid service starts bouncing down and up, maybe look into traffic received.
- This is what honeypots also do

Most Important Network Protocols in TCP/IP

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

UDP User Datagram Protocol

TCP Transmission Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Basically the ones you need to get online on the Internet

Binary Protocols

- Some protocols use binary formats
- Example DNS, which is a complex protocol
- When parsing DNS use standard libraries!
- When attacking DNS applications, use standard libraries! ☺
- DNS is just an example, new protocols may not be implemented in easy to use tools – but someone might have analyzed it or parts already!

Network Authentication

IPMI Authentication Bypass via Cipher 0

Dan Farmer identified a serious failing of the IPMI 2.0 specification, namely that cipher type 0, an indicator that the client wants to use clear-text authentication, actually allows access with any password. Cipher 0 issues were identified in HP, Dell, and Supermicro BMCs, with the issue likely encompassing all IPMI 2.0 implementations. It is easy to identify systems that have cipher 0 enabled using the `ipmi_cipher_zero` module in the Metasploit Framework.

- Sometimes people add network functionality to existing applications – and do this badly
- We have seen applications like IPMI and others
- Is this a network problem, server problem, hardware problem? – **discuss**

Source: https://blog.rapid7.com/2013/07/02/a-penetration-testers-guide-to-ipmi/

Even securiy protocols and software have problems

- ISAKMP example
- IKE(v1) has been critized as being overly complex
- Needed bake-off sessions where vendors meet and tried negotiating
- Searching for CVE ISAKMP show multiple vulnerabilities in various implementations, including firewalls and tcpdump
- Source: AoSSA chapter 16: Network Application Protocols
  The Art of Software Security Assessment Identifying and Preventing Software Vulnerabilities Mark Dowd, John McDonald, Justin Schuh ISBN: 9780321444424

Exercise

Now lets do the exercise

Sniff Your Browser 15min

which is number 33 in the exercise PDF.

ASN.1 problems

- Abstract format designed for representing objects in a machine independent format
- Used for various technologies in use on the internet:
- Certificates and key encoding
- Simple Network Management Protocol (SNMP)
- ISAKMP part of IPsec
- Lightweight Directory Access Protocol (LDAP)

Linux Kernel ASN.1

- CVE-2016-0758 Integer overflow in lib/asn1_decoder.c in the Linux kernel before 4.6 allows local users to gain privileges via crafted ASN.1 data.
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-0758
- Linux kernel have about 5 ASN.1 parsers
  https://www.x41-dsec.de/de/lab/blog/kernel_userspace/

Type Length Value TLVs

> TLV sequences are easily searched using generalized parsing functions; New message elements which are received at an older node can be safely skipped and the rest of the message can be parsed. This is similar to the way that unknown XML tags can be safely skipped; TLV elements can be placed in any order inside the message body; TLV elements are typically used in a binary format which makes parsing faster and the data smaller than in comparable text based protocols.

Source: https://en.wikipedia.org/wiki/Type-length-value

- Type Length Value is an encoding used in data communication
- For example in Link Layer Discovery Protocol (LLDP), SSH, RADIUS

Cisco Application Centric Infrastructure, aka Security Device

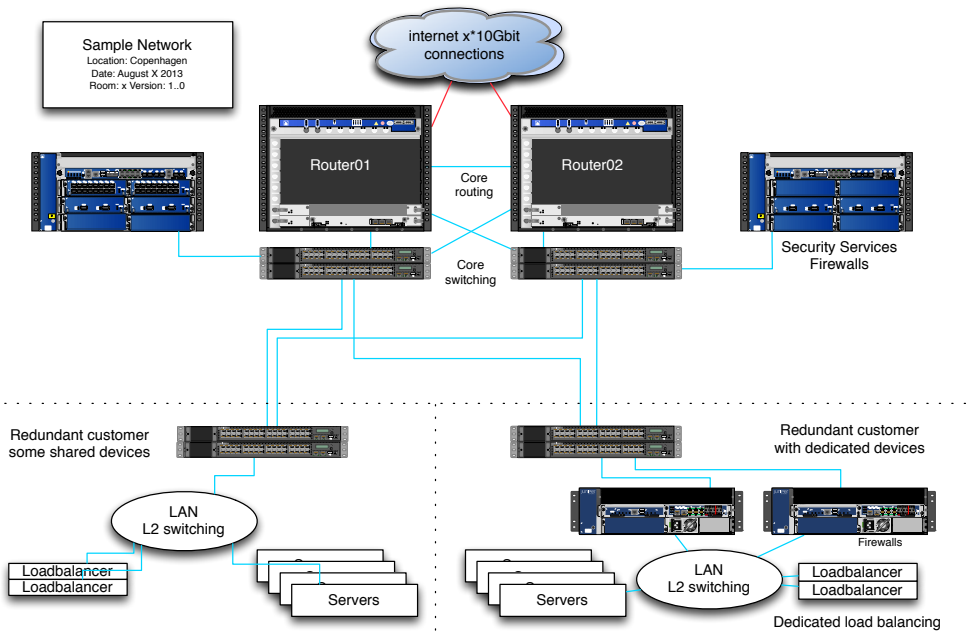
> The first time an APIC gets physically connected to one of the leaf switches of an ACI fabric, it will initiate a configuration process for the switches. The initial packets sent by the APIC are Link Layer Discovery Protocol (LLDP) packets containing information that is used by the leaf switch to initiate the configuration process. The LLDP protocol is used to advertise the identity, capabilities and certain other parameters of the APIC via TypeLength-Value (TLV) fields.

ERNW WHITEPAPER 68, SECURITY ASSESSMENT OF CISCO ACI, 2019

https://static.ernw.de/whitepaper/ERNW_Whitepaper68_Vulnerability_Assessment_Cisco_ACI_signed.pdf

- Cisco Nexus 9000 Series Fabric Switches ACI Mode Fabric Infrastructure VLAN Unauthorized Access Vulnerability (CVE-2019-1890)
- Cisco Nexus 9000 Series Fabric Switches Application Centric Infrastructure Mode **Link Layer Discovery Protocol Buffer Overflow Vulnerability** (CVE-2019-1901)

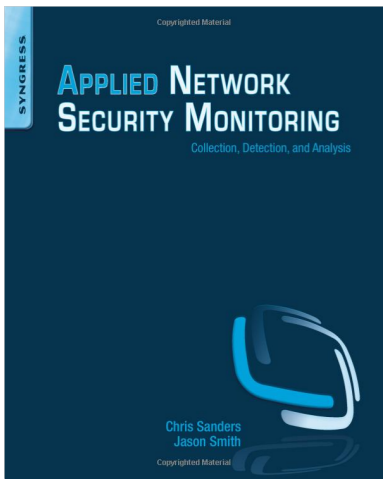
Networks today



Conclusion: networks are complex

Many places where you can block, filter, restrict, limit, avoid, process, change, observe

Book: Applied Network Security Monitoring (ANSM)

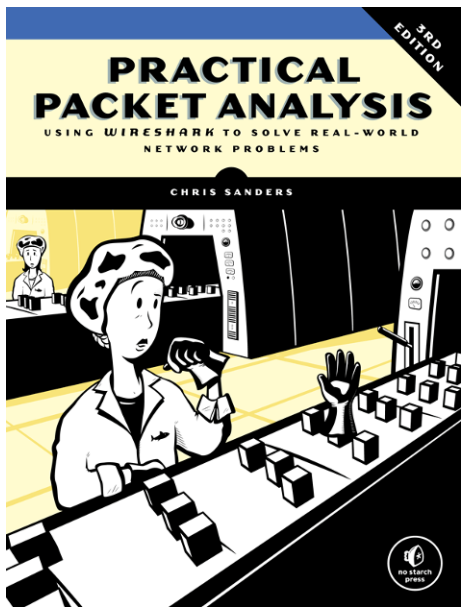


Applied Network Security Monitoring: Collection, Detection, and Analysis 1st Edition

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

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

Internet is Open Standards!

> We reject kings, presidents, and voting.
> We believe in rough consensus and running code.
> – The IETF credo Dave Clark, 1992.

Request for comments - RFC - er en serie af dokumenter

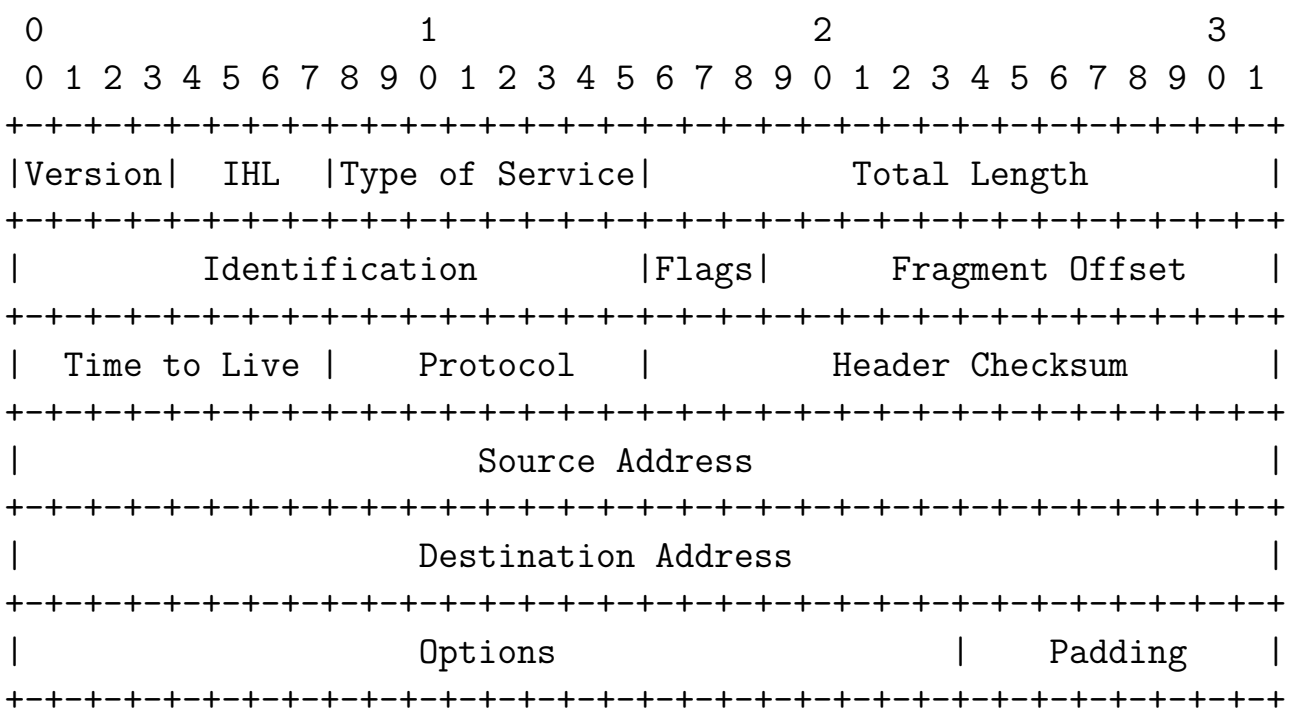
RFC, BCP, FYI, informational
de første stammer tilbage fra 1969

Ændres ikke, men får status Obsoleted når der udkommer en nyere version af en standard

Standards track:
Proposed Standard $\rightarrow$ Draft Standard $\rightarrow$ Standard

Åbne standarder = åbenhed, ikke garanti for sikkerhed

IPv4 packets – header - RFC-791



Example Internet Datagram Header

DNS problems

> The Domain Name System (DNS) [32][33] provides for a distributed database mapping host names to IP addresses. An intruder who interferes with the proper operation of the DNS can mount a variety of attacks, including denial of service and password collection. There are a number of vulnerabilities.

Source: *Security Problems in the TCP/IP Protocol Suite*, S.M. Bellovin 1989
https://www.cs.columbia.edu/~smb/papers/ipext.pdf

We have a lot of the same problems in DNS today

Plus some more caused by middle-boxes, NAT, DNS size, DNS inspection

- DNS must allow both UDP and TCP port 53
- Your DNS servers must have updated software, which use updated protocol behaviour
  see DNS flag day https://dnsflagday.net/
- DNS is unencrypted

Recommended Software – Unbound and NSD

Unbound is a validating, recursive, caching DNS resolver. It is designed to be fast and lean and incorporates modern features based on open standards.

To help increase online privacy, Unbound supports DNS-over-TLS which allows clients to encrypt their communication. In addition, it supports various modern standards that limit the amount of data exchanged with authoritative servers.

https://www.nlnetlabs.nl/projects/unbound/about/

My preferred local DNS server

Also check out uncensored DNS and his DNS over TLS setup!
Even has pinning information available:
https://blog.censurfridns.dk/blog/32-dns-over-tls-pinning-information-for-unicastcensurfridnsdk/

DNS over TLS vs DNS over HTTPS - DNS encryption

Protocols exist that encrypt DNS data, like dnscrypt which is not RFC standard https://dnscrypt.info/ https://en.wikipedia.org/wiki/DNSCrypt

Today we have competing standards:

Specification for DNS over Transport Layer Security (TLS) (DoT), RFC 7858 MAY 2016 https://en.wikipedia.org/wiki/DNS_over_TLS

DNS Queries over HTTPS (DoH) RFC 8484

DNS problems

- Failure to Deal with Invalid Label Lengths
- Insufficient Destination Lengths Check
- Insufficient Source Length Checks
- Pointer Values Not Verified In Packet
- Special Pointer Values
- Length Variables
- These are from the book: AoSSA chapter 16: Network Application Protocols
  The Art of Software Security Assessment Identifying and Preventing Software Vulnerabilities Mark Dowd, John McDonald, Justin Schuh ISBN: 9780321444424

Labels and pointers within packets save bytes, but make it more complex!

Firewalls and related issues

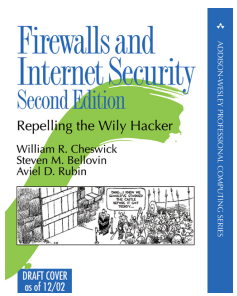
> In computing, a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.[1] A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.[2]

Source: Wikipedia

- https://en.wikipedia.org/wiki/Firewall_(computing)
- http://www.wilyhacker.com/ Cheswick chapter 2 PDF *A Security Review of Protocols: Lower Layers*
- Network layer, packet filters, application level, stateless, stateful

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

Firewall historik

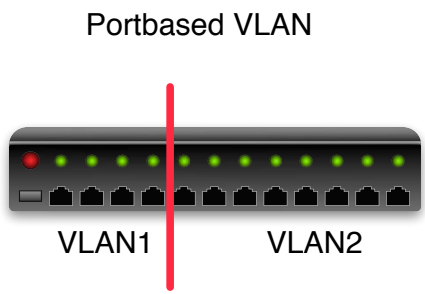


Firewalls har været kendt siden starten af 90'erne

Første bog *Firewalls and Internet Security* udkom i 1994 men kan stadig anbefales, læs den på http://www.wilyhacker.com/

2003 kom den i anden udgave *Firewalls and Internet Security* William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin, Addison-Wesley, 2nd edition

Together with Firewalls - VLAN Virtual LAN



Nogle switche tillader at man opdeler portene

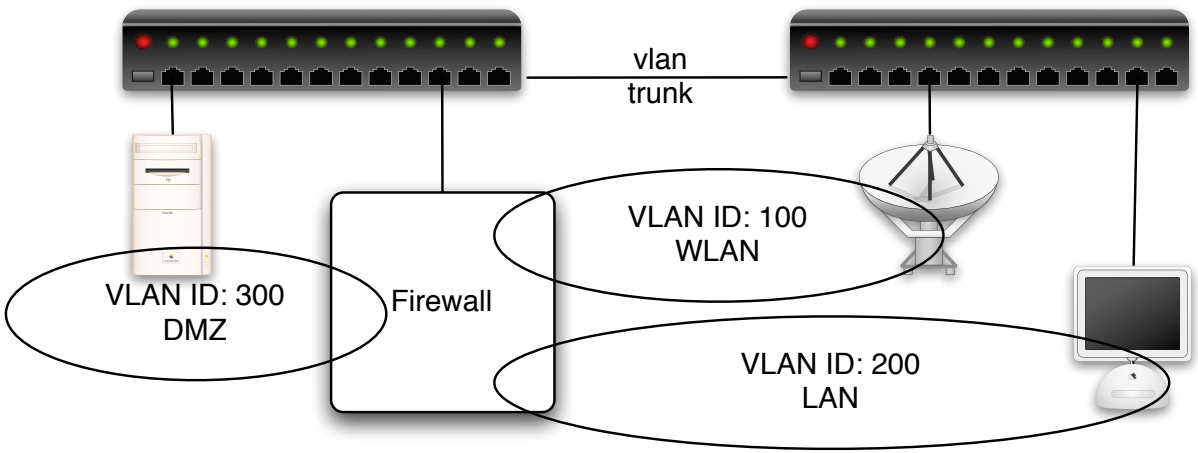
Denne opdeling kaldes VLAN og portbaseret er det mest simple

Port 1-4 er et LAN

De resterende er et andet LAN

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

IEEE 802.1q

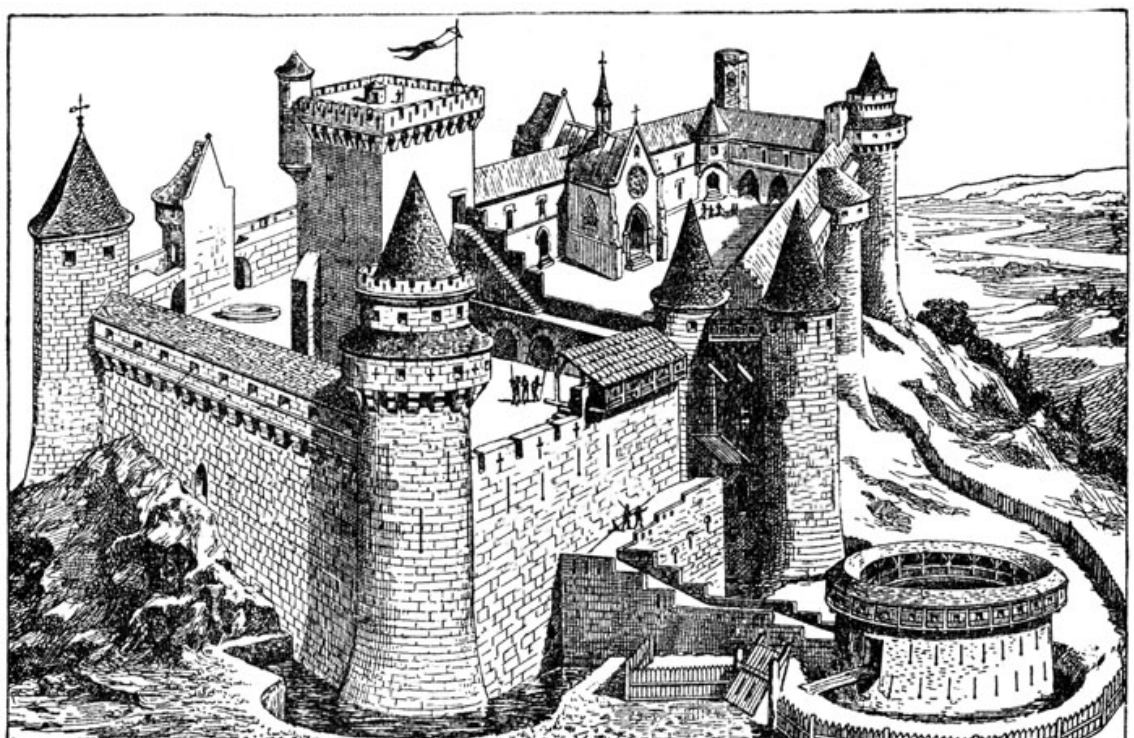


Med 802.1q tillades VLAN tagging på Ethernet niveau

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

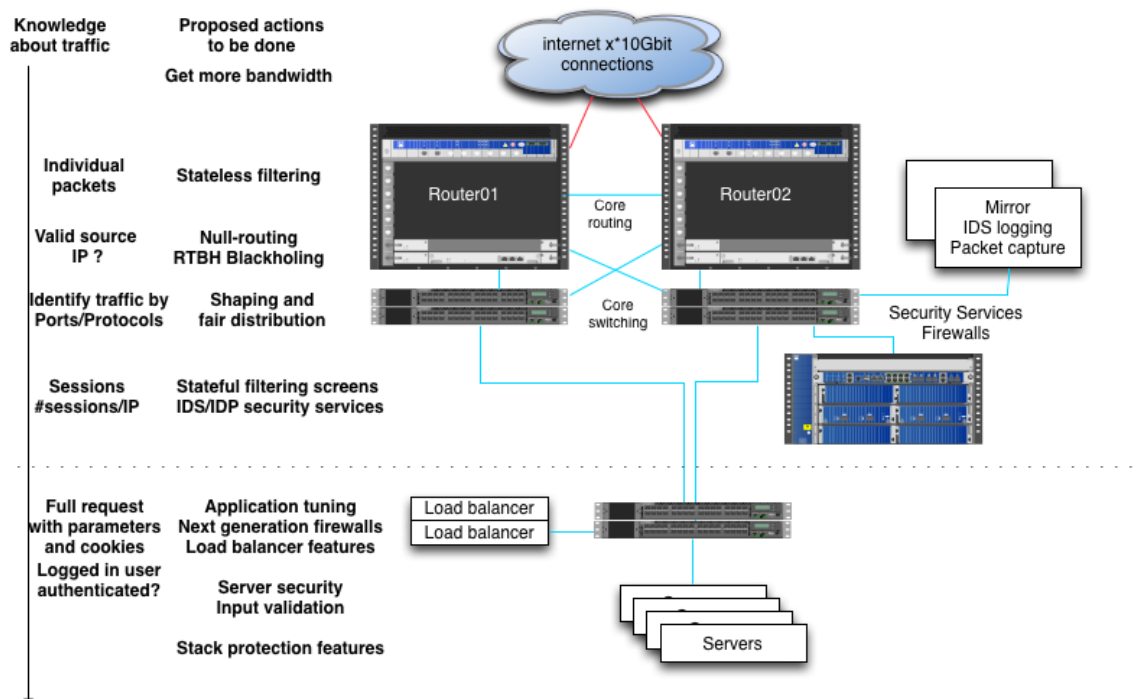
VLAN trunking giver mulighed for at dele VLANs ud på flere switches

Defense in depth



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

Firewall er ikke alene



Forsvaret er som altid - flere lag af sikkerhed!

Modern Firewalls

Basalt set et netværksfilter - det yderste fæstningsværk

Indeholder typisk:

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

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

Sample rules from OpenBSD PF

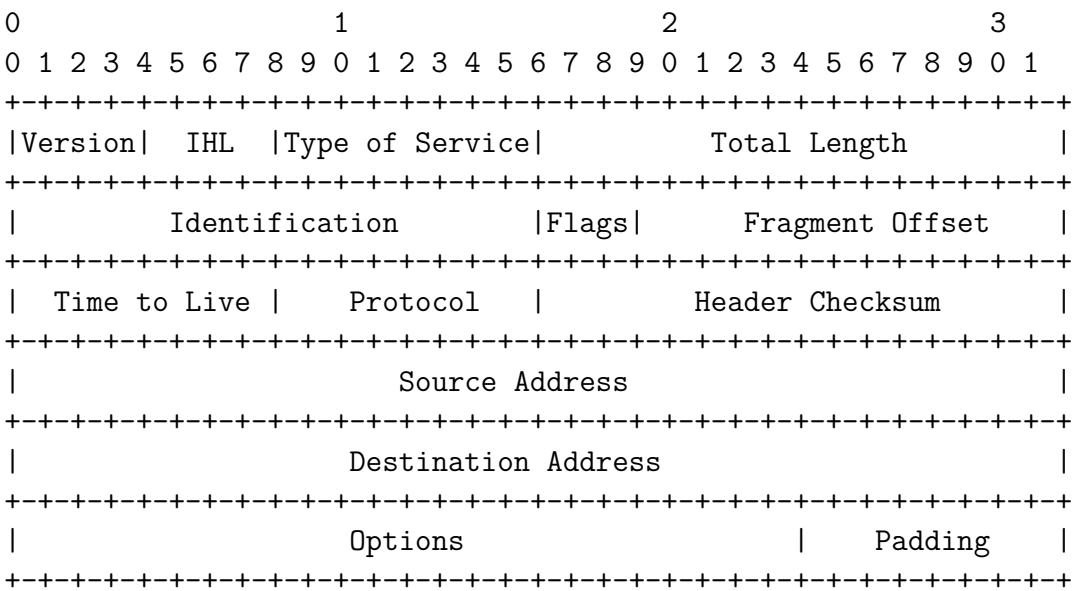
```
# hosts and networks
router="217.157.20.129"
webserver="217.157.20.131"
homenet=" 192.168.1.0/24, 1.2.3.4/24 "
wlan="10.0.42.0/24"
wireless=wi0
set skip lo0
# things not used
spoofed=" 127.0.0.0/8, 172.16.0.0/12, 10.0.0.0/16, 255.255.255.255/32 "

block in all # default block anything
# egress and ingress filtering - disallow spoofing, and drop spoofed
block in quick from $spoofed to any
block out quick from any to $spoofed

pass in on $wireless proto tcp from { $wlan $homenet } to any port = 22
pass in on $wireless proto tcp from any to $webserver port = 80

pass out
```

Packet filtering



Packet filtering er firewalls der filtrerer på IP niveau

Idag inkluderer de fleste stateful inspection

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

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

Executive Summary

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

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

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

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

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

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 is a presentation about DDoS protection with 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 workshop teaching people 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 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

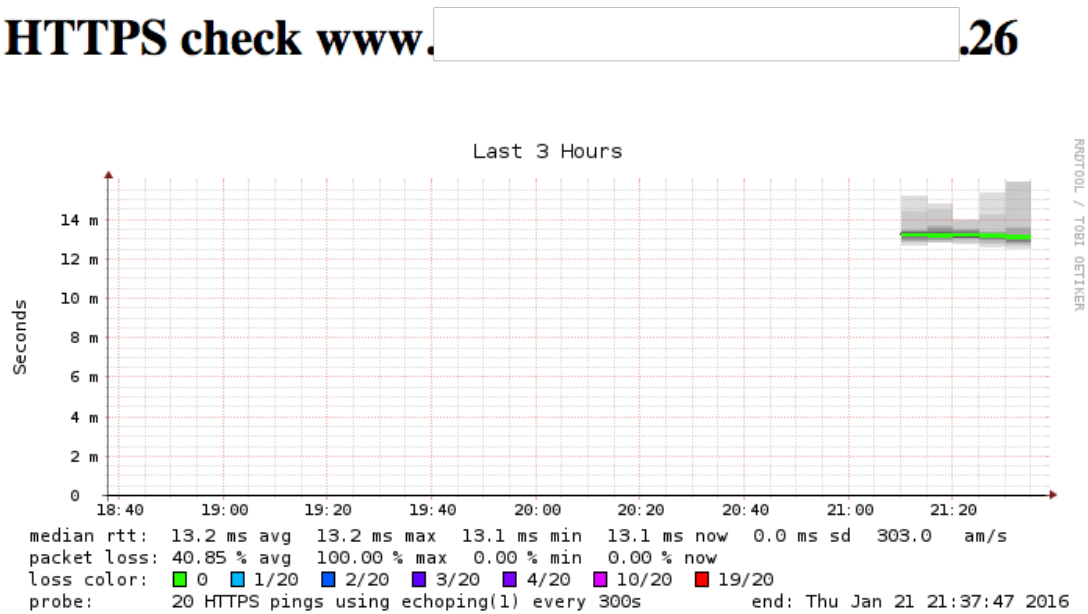
Extremely fast and breaks most firewalls when flooding, easy 800k pps/400Mbps

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

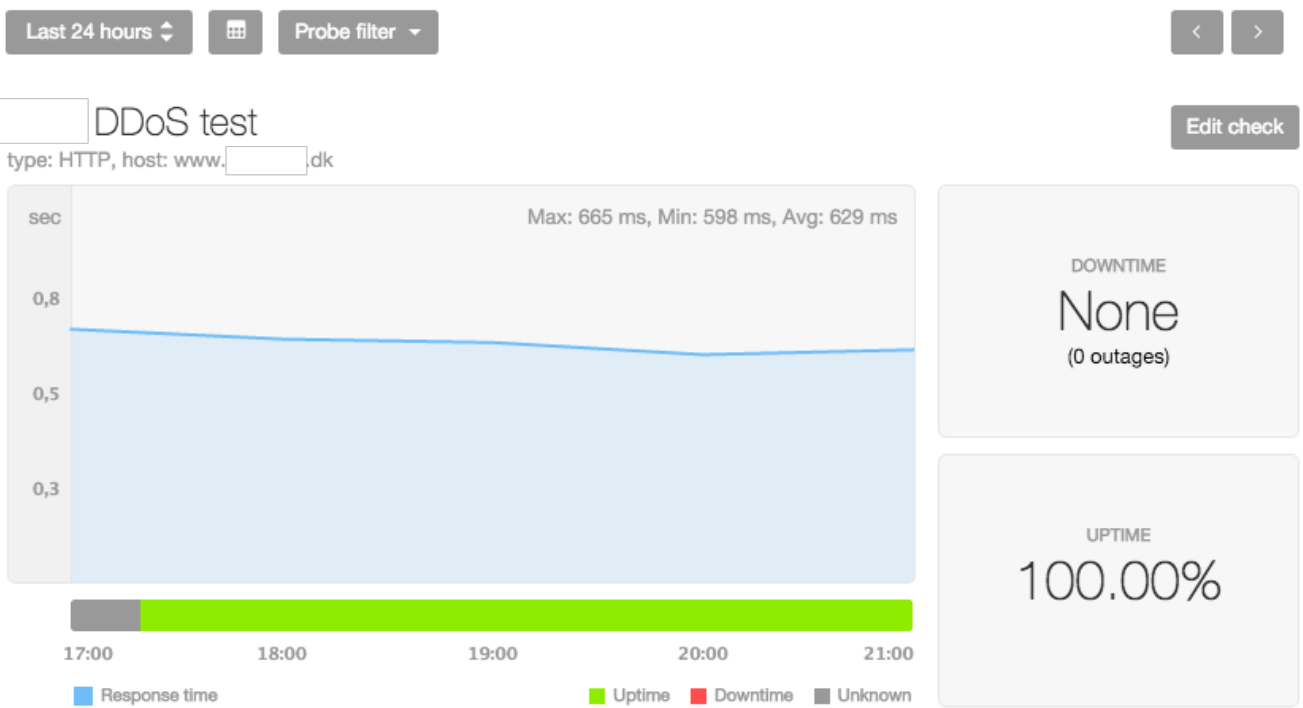
Ohh we lost our VPN into the environment, ohh the fw console is dead

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!

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

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 speeeeeeeeeeeed ☺

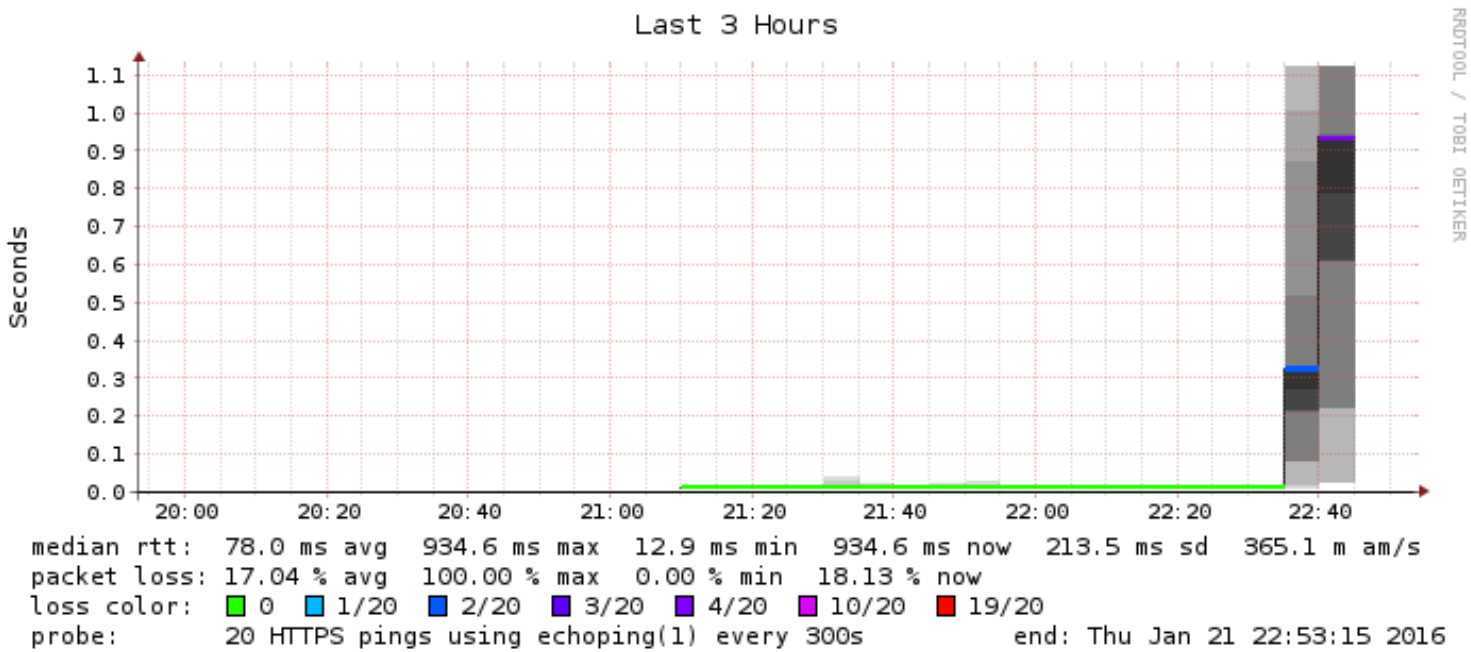
TCP testing use a port which is allowed through the network, often 80/443

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

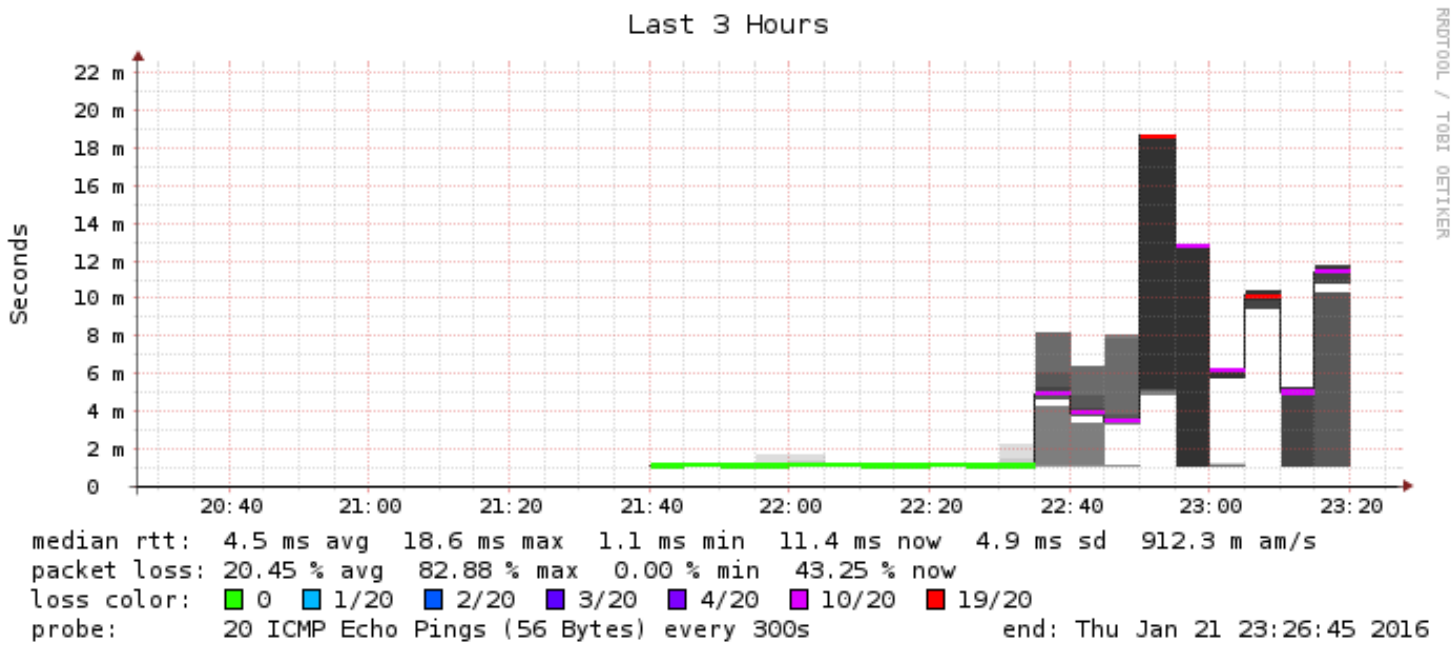
```
hping3 -q -c 1000000 -i u60 -S -p 22 $ROUTER_IP
hping3 -q -c 1000000 -i u60 -S -p 179 $ROUTER_IP
```

Rocky Horror Picture Show - 1



Really does it break from 50.000 pps SYN attack?

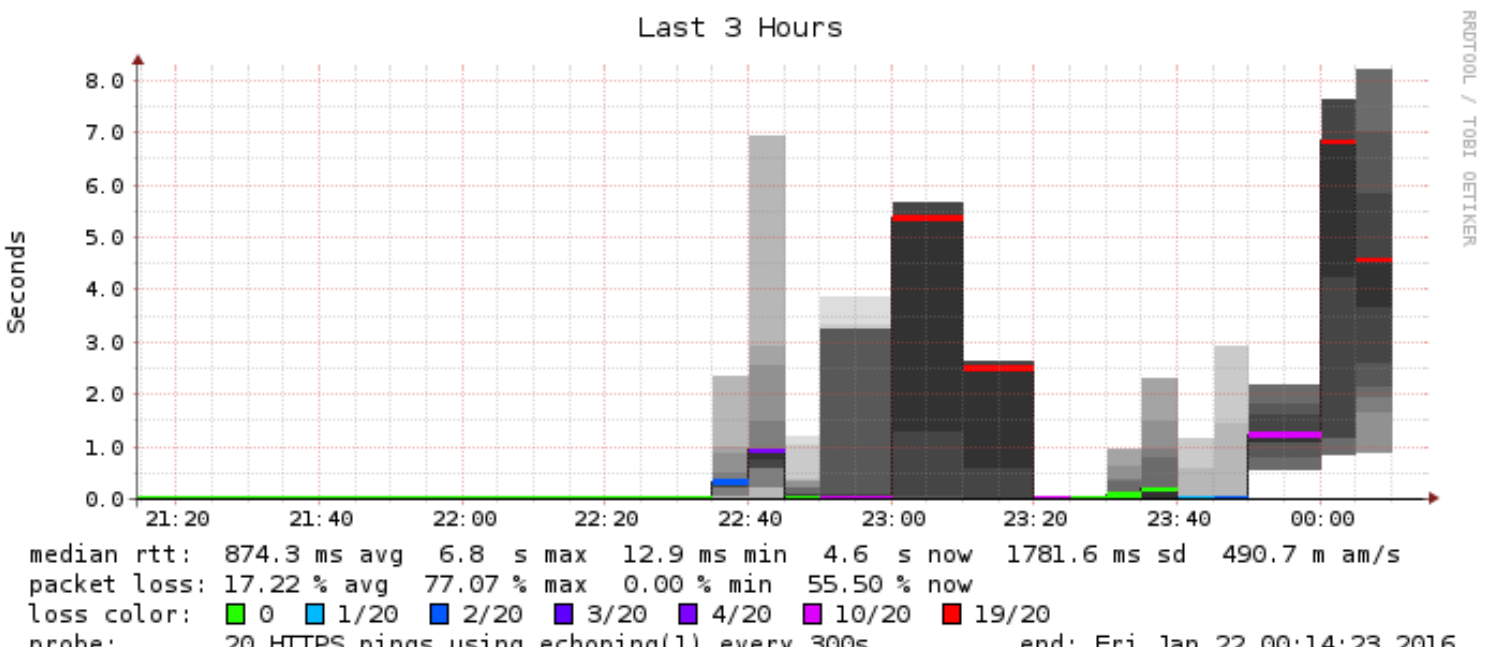
Rocky Horror Picture Show - 2



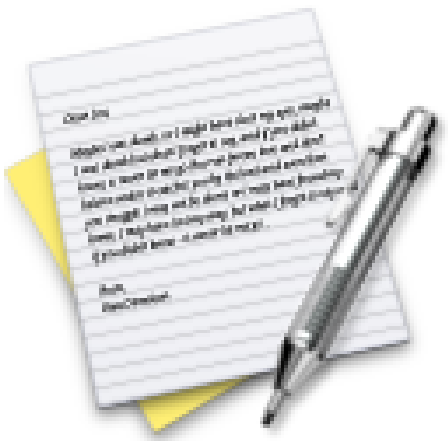
Oh no 500.000 pps UDP attacks work?

Rocky Horror Picture Show - 3

Oh no spoofing attacks work?



Exercise

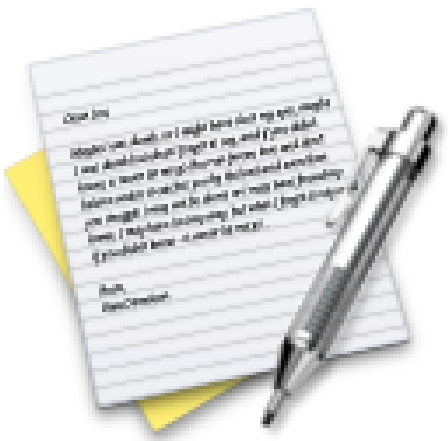


Now lets do the exercise

⚠ Execute nmap TCP and UDP port scan 20 min

which is number **34** in the exercise PDF.

Exercise

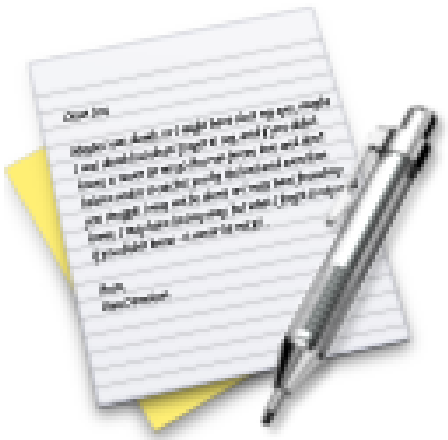


Now lets do the exercise

⚠ Discover active systems ping and port sweep 15 min

which is number **35** in the exercise PDF.

Exercise



Now lets do the exercise

❶ TCP SYN flooding 30min

which is number **36** in the exercise PDF.

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

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

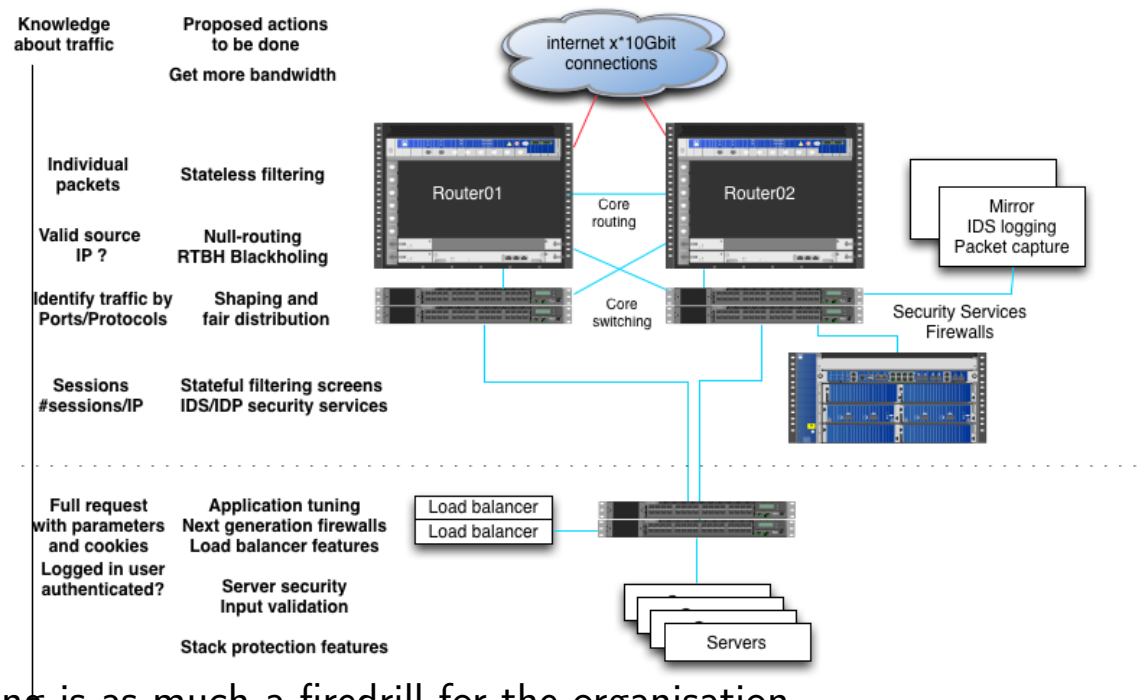
Conclusion

.

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

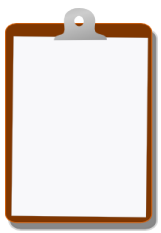


And dont forget that DDoS testing is as much a firedrill for the organisation

Strict stateless filtering

```
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;
}
```

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