# Communication and Network Security exercises

Henrik Lund Kramshoej
hlk@zencurity.com

February 18, 2019

# Contents



# Preface

This material is prepared for use in *Communication and Network Security workshop* and was prepared by Henrik Lund Kramshoej, http://www.zencurity.com . It describes the networking setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github
Look for communication-and-network-security-exercisesin the repo security-courses.

These exercises are expected to be performed in a training setting with network connected systems. The exercises use a number of tools which can be copied and reused after training. A lot is described about setting up your workstation in the repo

https://github.com/kramse/kramse-labs

## Prerequisites

This material expect that participants have a working knowledge of TCP/IP from a user perspective. Basic concepts such as web site addresses and email should be known as well as IP-addresses and common protocols like DHCP.

Have fun and learn

# Introduction to networking

## IP - Internet protocol suite

It is extremely important to have a working knowledge about IP to implement secure and robust infrastructures. Knowing about the alternatives while doing implementation will allow the selection of the best features.

## ISO/OSI reference model

A very famous model used for describing networking is the ISO/OSI model of networking which describes layering of network protocols in stacks.

This model divides the problem of communicating into layers which can then solve the problem as smaller individual problems and the solution later combined to provide networking.

Having layering has proven also in real life to be helpful, for instance replacing older hardware technologies with new and more efficient technologies without changing the upper layers.

In the picture the OSI reference model is shown along side with the Internet Protocol suite model which can also be considered to have different layers.

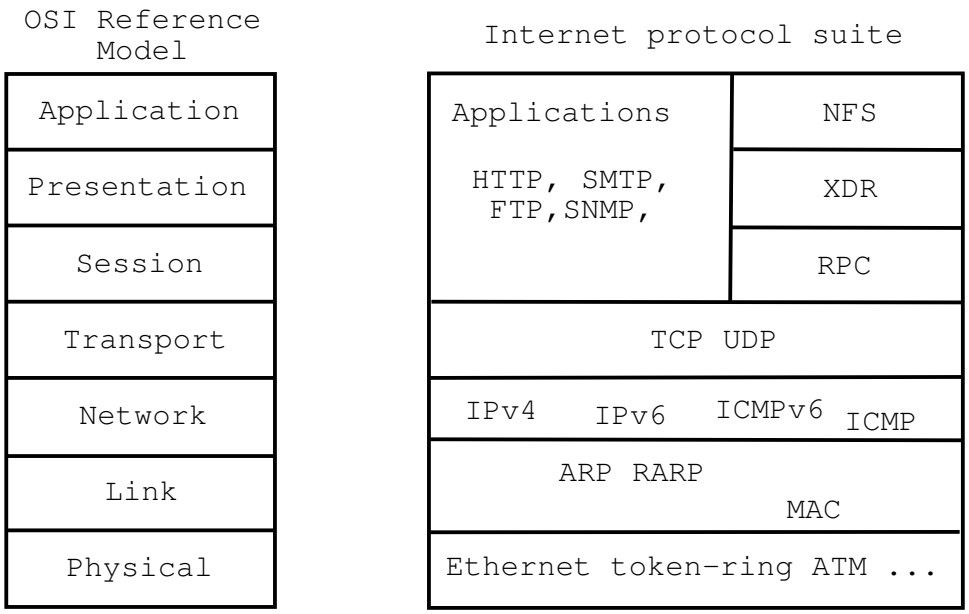



Figure 1: OSI og Internet Protocol suite

## Exercise content

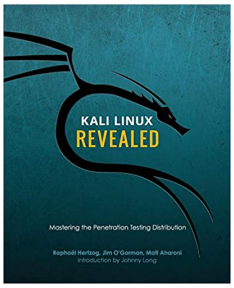
Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective
- **Purpose:** What is to be the expected outcome and goal of doing this exercise
- **Suggested method:** suggest a way to get started
- **Hints:** one or more hints and tips or even description how to do the actual exercises
- **Solution:** one possible solution is specified
- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

# Exercise 1

## Download Kali Linux Revealed (KLR) Book 10 min




*Kali Linux Revealed Mastering the Penetration Testing Distribution*

**Objective:**
We need a Kali Linux for running tools during the course. This is open source, and the developers have released a whole book about running Kali Linux.

This is named Kali Linux Revealed (KLR)

**Purpose:**
We need to install Kali Linux in a few moments, so better have the instructions ready.

**Suggested method:**
Create folders for educational materials. Go to https://www.kali.org/download-kali-linux-revealed-book/ Read and follow the instructions for downloading the book.

**Solution:**
When you have a directory structure for download for this course, and the book KLR in PDF you are done.

**Discussion:**
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux is a free pentesting platform, and probably worth more than $10.000

The book KLR is free, but you can buy/donate, and I recommend it.

# Exercise 2

## Check your Kali VM, run Kali Linux 30 min


**Objective:**
Make sure your virtual machine is in working order.

We need a Kali Linux for running tools during the course.

**Purpose:**
If your VM is not installed and updated we will run into trouble later.

**Suggested method:**
Go to https://github.com/kramse/kramse-labs/

Read the instructions for the setup of a Kali VM.

**Hints:**
If you allocate enough memory and disk you wont have problems.

**Solution:**
When you have a updated virtualisation software and Kali Linux, then we are good.

**Discussion:**
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Kali Linux includes many hacker tools and should be known by anyone working in infosec.

# Exercise 3

## Bonus: Check your Debian VM 10 min


**Objective:**
Make sure your virtual Debian 9 machine is in working order.

We need a Debian 9 Linux for running a few extra tools during the course.

**This is a bonus exercise - one is needed per team that want to try these tools. Tools which need Debian are Zeek and Suricata.**

**Purpose:**
If your VM is not installed and updated we will run into trouble later.

**Suggested method:**
Go to https://github.com/kramse/kramse-labs/

Read the instructions for the setup of a Kali VM.

**Hints:**

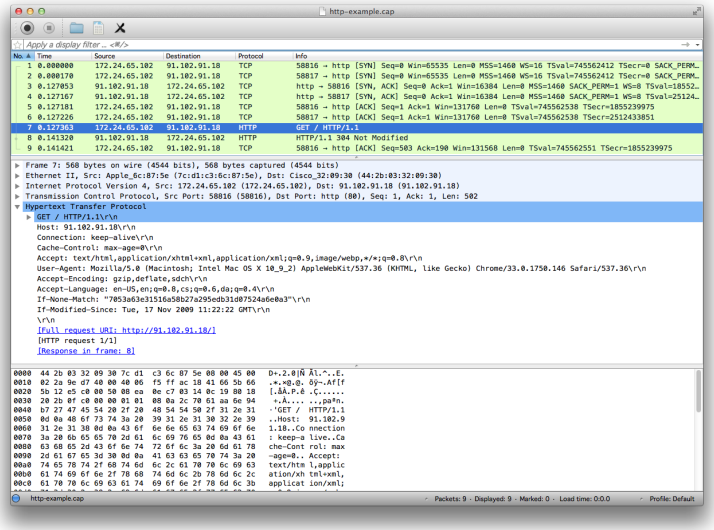
**Solution:**
When you have a updated virtualisation software and Kali Linux, then we are good.

**Discussion:**
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

# Exercise 4

## Wireshark and Tcpdump 15 min



**Objective:**
Try the program Wireshark locally your workstation, or tcpdump

You can run Wireshark on your host too, if you want.

**Purpose:**
Installing Wireshark will allow you to analyse packets and protocols

Tcpdump is a feature included in many operating systems and devices to allow packet capture and saving network traffic into files.

**Suggested method:**
Run Wireshark or tcpdump from your Kali Linux

The PPA book page 41 describes Your First Packet Capture.

**Hints:**
PCAP is a packet capture library allowing you to read packets from the network. Tcpdump uses libpcap library to read packet from the network cards and save them. Wireshark is a graphical application to allow you to browse through traffic, packets and protocols.

Both tools are already on your Kali Linux, or do: `apt-get install tcpdump wireshark`

**Solution:**
When Wireshark is installed sniff some packets. We will be working with both live traffic and saved packets from files in this course.

If you want to capture packets as a non-root user on Debian, then use the command to add a Wireshark group:

```
sudo dpkg-reconfigure wireshark-common
```

and add your user to this:

```
sudo gpasswd -a $USER wireshark
```

Dont forget to logout/login to pick up this new group.

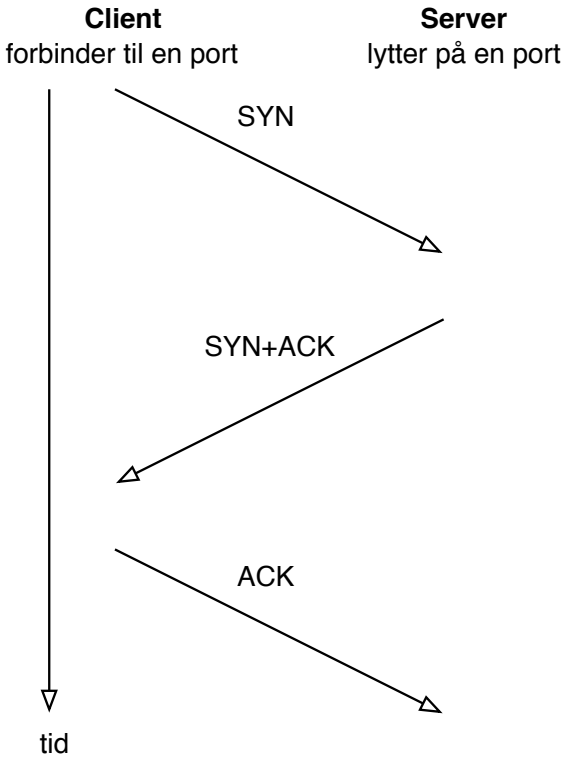
**Discussion:**
Wireshark is just an example other packet analyzers exist, some commercial and some open source like Wireshark

We can download a lot of packet traces from around the internet, we might use examples from
https://www.bro.org/community/traces.html

# Exercise 5

## Capturing TCP Session packets 10 min




**Objective:**
Sniff TCP packets and dissect them using Wireshark

**Purpose:**
See real network traffic, also know that a lot of information is available and not encrypted.

Note the three way handshake between hosts running TCP. You can either use a browser or command line tools like cURL while capturing

```
curl http://www.zencurity.com
```

**Suggested method:**
Open Wireshark and start a capture
Then in another window execute the ping program while sniffing

or perform a Telnet connection while capturing data

**Hints:**
When running on Linux the network cards are usually named eth0 for the first Ethernet and wlan0 for the first Wireless network card. In Windows the names of the network cards are long and if you cannot see which cards to use then try them one by one.

**Solution:**
When you have collected some TCP sessions you are done.

**Discussion:** Is it ethical to collect packets from an open wireless network?

Also note the TTL values in packets from different operating systems

# Exercise 6

## Whois databases 15 min

**Objective:**
Learn to lookup data in the global Whois databases

**Purpose:**
We often need to see where traffic is coming from, or who is responsible for the IP addresses sending attacks.

**Suggested method:**
Use a built-in command line, like: `host www.zencurity.dk` to look up an IP address and then `whois` with the IP address.

**Hints:**
Another option is to use web sites for doing Whois lookups https://apps.db.ripe.net/db-web-ui/#/query or their RIPEStat web site which can give even more information https://stat.ripe.net/

**Solution:**
When you can find our external address and look it up, you are done.

**Discussion:**
Whois databases are global and used for multiple purposes, the ones run by the Regional Internet Registries ARIN, RIPE, AfriNIC, LACNIC og APNIC have information about IP addresses and AS numbers allocated.

# Exercise 7

## Using ping and traceroute 10 min

**Objective:**
Be able to do initial debugging of network problems using commands ping and traceroute

**Purpose:**
Being able to verify connectivity is a basic skill.

**Suggested method:**
Use `ping` and `traceroute` to test your network connection - can be done one Windows and UNIX.

**Hints:**

```
$ ping 10.0.42.1
PING 10.0.42.1 (10.0.42.1) 56(84) bytes of data.
64 bytes from 10.0.42.1: icmp_seq=1 ttl=62 time=1.02 ms
64 bytes from 10.0.42.1: icmp_seq=2 ttl=62 time=0.998 ms
^C
--- 10.0.42.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.998/1.012/1.027/0.034 ms
```

Dont forget that UNIX ping continues by default, press ctrl-c to break.

Do the same with traceroute.

**Solution:**
Run both programs to local gateway and some internet address by your own choice.

**Discussion:**
Note the tool is called tracert on Windows, shortened for some reason.

ICMP is the Internet Control Message Protocol, usually used for errors like host unreachable. The ECHO request ICMP message is the only ICMP message that generates another.

The traceroute programs send packets with low Time To Live (TTL) and receives ICMP messages, unless there is a problem or a firewall/filter. Also used for mapping networks.

**Bonus:**

Whats the difference between:

- **traceroute** and **traceroute -I**
- NB: traceroute -I is found on UNIX - traceroute using ICMP pakker
- Windows tracert by default uses ICMP
- Unix by default uses UDP, but can use ICMP instead.
- Lots of traceroute-like programs exist for tracing with TCP or other protocols

# Exercise 8

## DNS and Name Lookups 10 min

**Objective:**
Be able to do DNS lookups from specific DNS server

**Purpose:**
Try doing DNS lookup using different programs

**Suggested method:**
Try the following programs:

- nslookup - UNIX and Windows, but not recommended
  `nslookup -q=txt -class=CHAOS version.bind. 0`
- dig - syntax @server domain query-type query-class
  `dig @8.8.8.8 www.example.com`
- host - syntaks host [-l] [-v] [-w] [-r] [-d] [-t querytype] [-a] host [server]
  `host www.example.com 8.8.8.8`

**Hints:**
Dig is the one used by most DNS admins, I often prefer the host command for the short output.

**Solution:**
Shown inline, above.

**Discussion:**
The nslookup program does not use the same method for lookup as the standard lookup libraries, results may differ from what applications see.

What is a zone transfer, can you get one using the host command?

Explain forward and reverse DNS lookup.

# Exercise 9

## Nping check ports 10 min

**Objective:**
Show the use of Nping tool for checking ports through a network

**Purpose:**
Nping can check if probes can reach through a network, reporting success of failure. Allows very specific packets to be sent.

**Suggested method:**
Run the command using a common port like Web HTTP:

```
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
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
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 1
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 1
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

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
```

**Hints:**
A lot of options are similar to Nmap

**Solution:**
When you have tried it towards an open port, a closed port and an IP/port that is filtered you are done.

**Discussion:**
A colleague of ours had problems sending specific IPsec packets through a provider. Using a tool like Nping it is possible to show what happens, or where things are blocked.

Things like changing the TTL may provoke ICMP messages, like this:

```
root@KaliVM:~# nping --tcp -p 80 --ttl 3 www.zencurity.com

Starting Nping 0.7.70 ( https://nmap.org/nping ) at 2018-09-07 19:08 CEST
SENT (0.0303s) TCP 10.137.0.24:37244 > 185.129.60.130:80 S ttl=3 id=60780 iplen=40  seq=1997801125 win=1480
RCVD (0.0331s) ICMP [10.50.43.225 > 10.137.0.24 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=28456 iplen=7
SENT (1.0314s) TCP 10.137.0.24:37244 > 185.129.60.130:80 S ttl=3 id=60780 iplen=40  seq=1997801125 win=1480
RCVD (1.0337s) ICMP [10.50.43.225 > 10.137.0.24 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=28550 iplen=7
SENT (2.0330s) TCP 10.137.0.24:37244 > 185.129.60.130:80 S ttl=3 id=60780 iplen=40  seq=1997801125 win=1480
RCVD (2.0364s) ICMP [10.50.43.225 > 10.137.0.24 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=28589 iplen=7
SENT (3.0346s) TCP 10.137.0.24:37244 > 185.129.60.130:80 S ttl=3 id=60780 iplen=40  seq=1997801125 win=1480
RCVD (3.0733s) ICMP [10.50.43.225 > 10.137.0.24 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=29403 iplen=7
SENT (4.0366s) TCP 10.137.0.24:37244 > 185.129.60.130:80 S ttl=3 id=60780 iplen=40  seq=1997801125 win=1480
RCVD (4.0558s) ICMP [10.50.43.225 > 10.137.0.24 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=30235 iplen=7

Max rtt: 38.574ms | Min rtt: 2.248ms | Avg rtt: 13.143ms
Raw packets sent: 5 (200B) | Rcvd: 5 (360B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.07 seconds
```

# Exercise 10

## Try pcap-diff 15 min

**Objective:**
Try both getting an utility tool from Github and running an actual useful tool for comparing packet captures.

**Purpose:**
Being able to get tools and scripts from Github makes you more effective.

The tool we need today is https://github.com/isginf/pcap-diff **Suggested method:**
Git clone the repository, follow instructions for running a packet diff.

Try saving a few packets in a packet capture, then using tcpdump read and write a subset - so you end up with two packet captures:

```
sudo tcpdump -w icmp-dump.cap
// run ping in another window, which probably creates ARP packets
// Check using tcpdump
sudo tcpdump -r icmp-dump.cap arp
reading from file icmp-dump.cap, link-type EN10MB (Ethernet)
10:06:18.077055 ARP, Request who-has 10.137.0.22 tell 10.137.0.6, length 28
10:06:18.077064 ARP, Reply 10.137.0.22 is-at 00:16:3e:5e:6c:00 (oui Unknown), length 28
10:06:24.776987 ARP, Request who-has 10.137.0.6 tell 10.137.0.22, length 28
10:06:24.777107 ARP, Reply 10.137.0.6 is-at fe:ff:ff:ff:ff:ff (oui Unknown), length 28
// Write the dump - but without the ARP packets:
sudo tcpdump -r icmp-dump.cap -w icmp-dump-no-arp.cap not arp
```

With these pcaps you should be able to do:

```
sudo pip install scapy
git clone https://github.com/isginf/pcap-diff.git
cd pcap-diff/

$ python pcap_diff.py -i ../icmp-dump.cap -i ../icmp-dump-no-arp.cap -o diff.cap
Reading file ../icmp-dump.cap:
Found 23 packets

Reading file ../icmp-dump-no-arp.cap:
Found 19 packets

Diffing packets:

Found 2 different packets

Writing diff.cap
// Try reading the output packet diff:

$ sudo tcpdump -r diff.cap
```

```
reading from file diff.cap, link-type EN10MB (Ethernet)
10:06:24.777107 ARP, Reply 10.137.0.6 is-at fe:ff:ff:ff:ff:ff (oui Unknown), length 28
10:06:24.776987 ARP, Request who-has 10.137.0.6 tell 10.137.0.22, length 28
```

Note: I ran these on a Debian, so I needed the sudo, if you run this on Kali there is no need to use sudo.

**Hints:**
Git is one of the most popular software development tools, and Github is a very popular site for sharing open source tools.

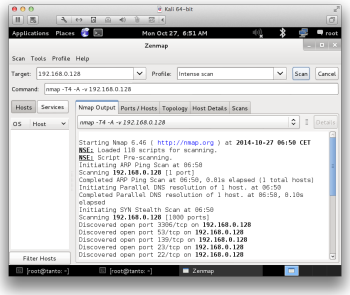
**Solution:**
When you or your team mate has a running pcap-diff then you are done

**Discussion:**
I often find that 90% of my tasks can be done using existing open source tools.

# Exercise 11

## Discover active systems ping sweep 10 min



**Objective:**
Use nmap to discover active systems

**Purpose:**
Know how to use nmap to scan networks for active systems.

**Suggested method:**
Try different scans,

- Ping sweep to find active systems
- Port sweeps to find active systems with specific ports

**Hints:**
Try nmap in sweep mode - and you may run this from Zenmap

**Solution:**
Use the command below as examples:

- Ping sweep `nmap -sP 10.0.45.*`
- Port sweeps `nmap -p 80 10.0.45.*`

**Discussion:**
Quick scans quickly reveal interesting hosts, ports and services

Also now make sure you understand difference between single host scan 10.0.45.123/32, a whole subnet /24  250 hosts 10.0.45.0/24 and other more advanced targeteting like 10.0.45.0/25 and 10.0.45.1-10

# Exercise 12

## Execute nmap TCP and UDP port scan 20 min

**Objective:**
Use nmap to discover important open ports on active systems

**Purpose:**
Finding open ports will allow you to find vulnerabilities on these ports.

**Suggested method:**
Use `nmap -p 1-1024 server` to scan the first 1024 TCP ports and use Nmap without ports. What is scanned then?

Try to use `nmap -sU` to scan using UDP ports, not really possible if a firewall is in place.

If a firewall blocks ICMP you might need to add `-Pn` to make nmap scan even if there are no Ping responses

**Hints:**
Sample command: `nmap -Pn -sU -p1-1024 server` UDP port scanning 1024 ports without doing a Ping first

**Solution:**
Discover some active systems and most interesting ports, which are 1-1024 and the built-in list of popular ports.

**Discussion:**
There is a lot of documentation about the nmap portscanner, even a book by the author of nmap. Make sure to visit http://www.nmap.org

TCP and UDP is very different when scanning. TCP is connection/flow oriented and requires a handshake which is very easy to identify. UDP does not have a handshake and most applications will not respond to probes from nmap. If there is no firewall the operating system will respond to UDP probes on closed ports - and the ones that do not respond must be open.

When doing UDP scan on the internet you will almost never get a response, so you cannot tell open (not responding services) from blocked ports (firewall drop packets). Instead try using specific service programs for the services, sample program could be `nsping` which sends DNS packets, and will often get a response from a DNS server running on UDP port 53.

# Exercise 13

## Perform nmap OS detection 10 min

**Objective:**
Use nmap OS detection and see if you can guess the brand of devices on the network

**Purpose:**
Getting the operating system of a system will allow you to focus your next attacks.

**Suggested method:**
Look at the list of active systems, or do a ping sweep.

Then add the OS detection using the option `-O`

Better to use -A all the time, includes even more scripts and advanced stuff See the next exercise.

**Hints:**
The nmap can send a lot of packets that will get different responses, depending on the operating system. TCP/IP is implemented using various constants chosen by the implementors, they have chosen different standard packet TTL etc.

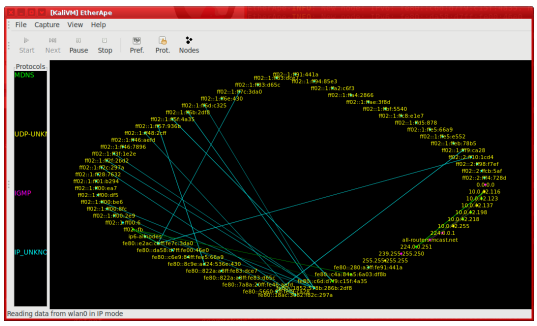
**Solution:**
Use a command like `nmap -O -p1-100 10.0.45.45` or `nmap -A -p1-100 10.0.45.45`

**Discussion:**
nmap OS detection is not a full proof way of knowing the actual operating system, but in most cases in can detect the family and in some cases it can identify the exact patch level of the system.

# Exercise 14

## EtherApe 10 min



> 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. Node statistics can be exported.

**Objective:**
Use a tool to see more about network traffic, whats going on in a network.

**Purpose:**
Get to know the concept of a node by seeing nodes communicate in a graphical environment.

**Suggested method:**
Use the tool from Kali

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

**Hints:**
Your built-in network card may not be the best for sniffing. Borrow one from Henrik.

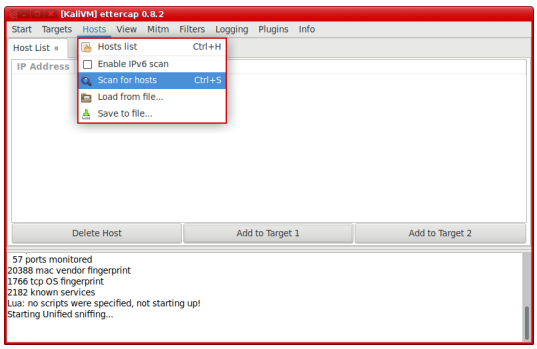
**Solution:**
When you have the tool running and showing data, you are done.

**Discussion:**

# Exercise 15

## ARP spoofing and ettercap 20 min




**Objective:**
Use a tool to see more about network traffic, whats going on in a network.

**Purpose:**
Start the tool, do a scan and start sniffing between your laptop and the router.

**Suggested method:**

1. Start the tool using `ettercap --gtk` to get the graphical version.
2. Select menu Info, Help - and read about unified and bridged sniffing.
3. Start Unified sniffing from Sniff, Unified sniffing - select your network card.
4. Select Hosts - Scan

You should be able to see some hosts. Then the next step would be to initiate attacks - which are menu-driven and easy to perform.

**Hints:**
We might be messing to much with the traffic, so attacks wont succeed. Some coordination is needed.

**Solution:**
When you can scan for hosts and realize how easy that was, you are done.

**Discussion:**
How many admins know about ARP spoofing, ARP poisoning?

# Exercise 16

## Perform nmap service scan 10 min

**Objective:**
Use more advanced features in Nmap to discover services.

**Purpose:**
Getting more intimate with the system will allow more precise discovery of the vulnerabilities and also allow you to select the next tools to run.

**Suggested method:**
Use `nmap -A` option for enabling service detection and scripts

**Hints:**
Look into the manual page of nmap or the web site book about nmap scanning

**Solution:**
Run nmap and get results.

**Discussion:**
Some services will show software versions allowing an attacker easy lookup at web sites to known vulnerabilities and often exploits that will have a high probability of success.

Make sure you know the difference between a vulnerability which is discovered, but not really there, a false positive, and a vulnerability not found due to limitations in the testing tool/method, a false negative.

A sample false positive might be reporting that a Windows server has a vulnerability that you know only to exist in Unix systems.

# Exercise 17

## Nmap full scan - strategy 15 min

**Objective:**
Write down your Nmap strategy, and if needed create your own Nmap profile in Zenmap.

**Purpose:**
Doing a port scan often requires you to run multiple Nmap scans.

**Suggested method:**
Use Zenmap to do:

1. A few quick scans, to get web servers and start web scanners/crawlers
2. Full scan of all TCP ports, -p 1-65535
3. Full or limited UDP scan, `nmap -sU --top-ports 100`
4. Specialized scans, like specific source ports

**Hints:**
Using a specific source ports using -g/–source-port <portnum>: Use given port number with ports like FTP 20, DNS 53 can sometimes get around router filters and other stateless Access Control Lists

**Solution:**
Run nmap and get results.

**Discussion:**
Recommendation it is highly recommended to always use:

```
-iL <inputfilename>: Input from list of hosts/networks
-oA outputbasename: output in all formats, see later
```

Some examples of real life Nmaps I have run recently:

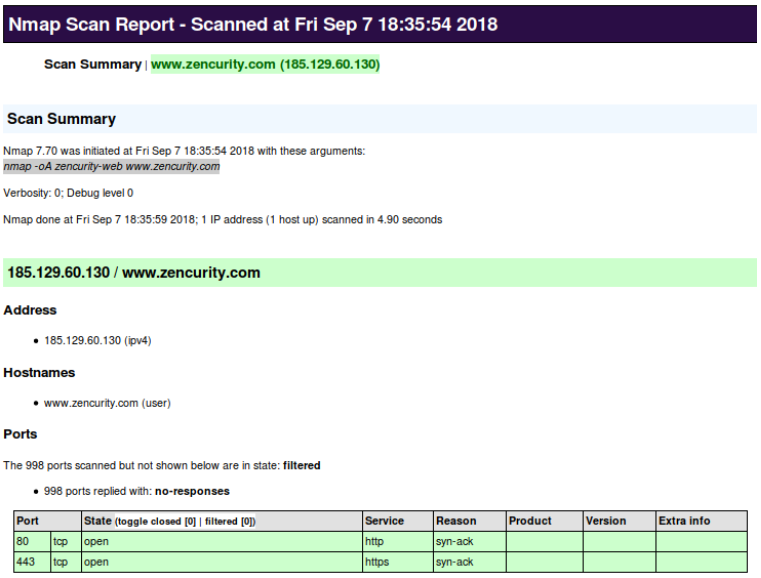
```
dns-scan: nmap -sU -p 53 --script=dns-recursion -iL targets -oA dns-recursive
bgpscan: nmap -A -p 179 -oA bgpscan -iL targets
dns-recursive: nmap -sU -p 53 --script=dns-recursion -iL targets -oA dns-recursive
php-scan: nmap -sV --script=http-php-version -p80,443 -oA php-scan -iL targets
scan-vtep-tcp: nmap -A -p 1-65535 -oA scan-vtep-tcp 185.129.60.77 185.129.60.78
```

```
snmp-10.x.y.0.gnmap: nmap -sV -A -p 161 -sU --script=snmp-info -oA snmp-10xy 10.x.y.0/19
snmpscan: nmap -sU -p 161 -oA snmpscan --script=snmp-interfaces -iL targets
sshscan: nmap -A -p 22 -oA sshscan -iL targets
vncscan: nmap -A -p 5900-5905 -oA vncscan -iL targets
```

# Exercise 18

## Reporting HTML 15 min



Nmap Scan Report - Scanned at Fri Sep 7 18:35:54 2018

Scan Summary | www.zencurity.com (185.129.60.130)

Scan Summary

Nmap 7.70 was initiated at Fri Sep 7 18:35:54 2018 with these arguments:
*nmap -oA zencurity-web www.zencurity.com*

Verbosity: 0; Debug level 0

Nmap done at Fri Sep 7 18:35:59 2018; 1 IP address (1 host up) scanned in 4.90 seconds

185.129.60.130 / www.zencurity.com

Address

- 185.129.60.130 (ipv4)

Hostnames

- www.zencurity.com (user)

Ports

The 998 ports scanned but not shown below are in state: **filtered**

- 998 ports replied with: **no-responses**

| Port | | State (toggle closed [0] \| filtered [0]) | Service | Reason | Product | Version | Extra info |
|---|---|---|---|---|---|---|---|
| 80 | tcp | open | http | syn-ack | | | |
| 443 | tcp | open | https | syn-ack | | | |

**Objective:**
Show the use of XML output and convert to HTML

**Purpose:**
Reporting data is very important. Using the oA option Nmap can export data in three formats easily, each have their use. They are normal, XML, and grepable formats at once.

**Suggested method:**
First run Nmap, with output, then process it

```
sudo nmap -oA zencurity-web www.zencurity.com
xsltproc zencurity-web.xml > zencurity-web.html
```

**Hints:**
Nmap includes the stylesheet in XML and makes it very easy to create HTML.

If the tool `xsltproc` is not installed, then install it: `apt-get install xsltproc`

**Solution:**
Run XML through xsltproc, command line XSLT processor, or another tool

**Discussion:**

Options you can use to change defaults:

```
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
```

Also check out the Ndiff tool

```
hlk@cornerstone03:~$ ndiff zencurity-web.xml zencurity-web-2.xml
-Nmap 7.70 scan initiated Fri Sep 07 18:35:54 2018 as: nmap -oA zencurity-web www.zencurity.com
+Nmap 7.70 scan initiated Fri Sep 07 18:46:01 2018 as: nmap -oA zencurity-web-2 www.zencurity.com

 www.zencurity.com (185.129.60.130):
 PORT    STATE SERVICE VERSION
+443/tcp open  https
```

(I ran a scan, removed a port from the first XML file and re-scanned)

# Exercise 19

## SSL/TLS scanners 15 min

**Objective:**
Try the Online Qualys SSLLabs scanner https://www.ssllabs.com/ Try the command line tool sslscan checking servers - can check both HTTPS and non-HTTPS protocols!

**Purpose:**
Learn how to efficiently check TLS settings on remote services.

**Suggested method:**
Run the tool against a couple of sites of your choice.

```
root@kali:~# sslscan --ssl2 web.kramse.dk
Version: 1.10.5-static
OpenSSL 1.0.2e-dev xx XXX xxxx

Testing SSL server web.kramse.dk on port 443
...
  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.kramse.dk
Altnames: DNS:*.kramse.dk, DNS:kramse.dk
Issuer:   AlphaSSL CA - SHA256 - G2
```

Also run it without `--ssl2` and against SMTPTLS if possible.

**Hints:**
Originally sslscan is from http://www.titania.co.uk but use the version on Kali, install with apt if not installed.

**Solution:**
When you can run and understand what the tool does, you are done.

**Discussion:**
SSLscan can check your own sites, while Qualys SSLLabs only can test from hostname

# Exercise 20

## sslstrip 15 min

**Objective:**
sslstrip https://moxie.org/software/sslstrip/

**Purpose:**
Read about the tool, and lets try to run it - on a single VM.

> This tool provides a demonstration of the HTTPS stripping attacks that I presented at Black Hat DC 2009. It will transparently hijack HTTP traffic on a network, watch for HTTPS links and redirects, then map those links into either look-alike HTTP links or homograph-similar HTTPS links. It also supports modes for supplying a favicon which looks like a lock icon, selective logging, and session denial. For more information on the attack, see the video from the presentation below.

**Suggested method:**
Make sure tool is installed, Then run it and intercept your own traffic from the same system.

You may run this on the wireless and try intercepting others.

**Hints:**
IF you are using wireless - most likely, then make sure to run on the same channel/AP/frequency. Either switch everything to 2.4GHz and have only one AP or just do the mitm on a single host - run browser and mitmproxy on the same VM.

**Solution:**
When you have intercepted some traffic you are done, we will spend at least 30-45 minutes doing various mitm related stuff.

**Discussion:**

# Exercise 21

## mitmproxy 30 min

**Objective:**
mitmproxy https://mitmproxy.org/

mitmproxy is a free and open source interactive HTTPS proxy

**Purpose:**
Try running a mitm attack on your phone or another laptop.

**Suggested method:**
Make sure tool is installed

Then use the command line interface to verify it is working, then switch to the Web interface for playing with the tool.

**Hints:**
IF you are using wireless - most likely, then make sure to run on the same channel/AP/frequency. Either switch everything to 2.4GHz and have only one AP or just do the mitm on a single host - run browser and mitmproxy on the same VM.

**Solution:**
When you have intercepted some traffic you are done, we will spend at least 30-45 minutes doing various mitm related stuff.

**Discussion:**

# Exercise 22

## sslsplit 10 min

**Objective:**
Read about sslsplit https://www.roe.ch/SSLsplit - transparent SSL/TLS interception system

**Purpose:**
This tool has a lot of features described on the home page.

> Overview
> SSLsplit is a tool for man-in-the-middle attacks against SSL/TLS encrypted network connections. It is intended to be useful for network forensics, application security analysis and penetration testing.
>
> SSLsplit is designed to transparently terminate connections that are redirected to it using a network address translation engine. SSLsplit then terminates SSL/TLS and initiates a new SSL/TLS connection to the original destination address, while logging all data transmitted. Besides NAT based operation, SSLsplit also supports static destinations and using the server name indicated by SNI as upstream destination. SSLsplit is purely a transparent proxy and cannot act as a HTTP or SOCKS proxy configured in a browser.

**Suggested method:**
If we were to use this tool, we would redirect traffic using ”firewalls”/routers

- SSLsplit currently supports the following operating systems and NAT engines:
- FreeBSD: pf rdr and divert-to, ipfw fwd, ipfilter rdr
- OpenBSD: pf rdr-to and divert-to
- Linux: netfilter REDIRECT and TPROXY
- Mac OS X: ipfw fwd and pf rdr

**We wont run this tool, but beware such tools exist**

**Hints:**
Specifically read the section *SSLsplit implements a number of defences against mechanisms* - and think about the consequences to a regular user.

**Solution:**
When you feel you have a idea about what this tool can do then you are done.

**Discussion:**
Should tools like this even exist?