

# Communication and Network Security

## exercises

Henrik Kramselund  
xhek@kea.dk

April 15, 2024



Note: exercises marked with **▲** are considered important. These contain subjects that are essential for the course and curriculum. Even if you don't work through the exercise, you are expected to know the subjects covered by these.

Exercises marked with **❗** are considered optional. These contain subjects that are related to the course and curriculum. You may want to browse these and if interested work through them. They may require more time than we have available during the course.

# Contents

1	 Download Kali Linux Revealed (KLR) Book 10 min	3
2	 Check your Kali VM, run Kali Linux 30 min	4
3	 Download Debian Administrators Handbook (DEB) Book 10 min	5
4	 Check your Debian VM 10 min	6
5	 Investigate /etc 10 min	7
6	 Enable firewall - 15min	8
7	 Git tutorials - 15min	10
8	 Use Ansible to install Elastic Stack	11
9	 Configure Elasticsearch Security	13
10	 Wireshark and Tcpcap 15 min	15
11	 Capturing TCP Session packets 10 min	17
12	 Whois databases 15 min	18
13	 IP address research 30 min	19
14	 Using ping and traceroute 10 min	20
15	 DNS and Name Lookups 10 min	21
16	 Nping check ports 10 min	22
17	 Try pcap-diff 15 min	24
18	 Discover active systems ping sweep 10 min	26
19	 Execute nmap TCP and UDP port scan 20 min	27
20	 Perform nmap OS detection 10 min	28
21	 EtherApe 10 min	29
22	 ARP spoofing and ettercap 20min	30
23	 TCP SYN flooding 30min	31
24	 TCP other flooding 15min	33
25	 UDP flooding NTP, etc. 15min	34
26	 ICMP flooding 15min	35

27	📌 Misc - stranger attacks 15min	37
28	⚠️ Perform nmap service scan 10 min	39
29	⚠️ SSH scanners - 15min	40
30	📌 Configure SSH keys for more secure access	42
31	⚠️ Nmap full scan - strategy 15 min	44
32	📌 Reporting HTML 15 min	45
33	⚠️ SSL/TLS scanners 15 min	47
34	⚠️ Internet scanners 15 min	48
35	📌 Nginx as a Transport Layer Security (TLS) endpoint 40 min	49
36	⚠️ Nginx logging 20 min	50
37	⚠️ Nginx filtering 40 min	51

## Preface

This material is prepared for use in Communication and Network Security workshop and was prepared by Henrik Kramselund , <http://www.zecurity.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](mailto: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.

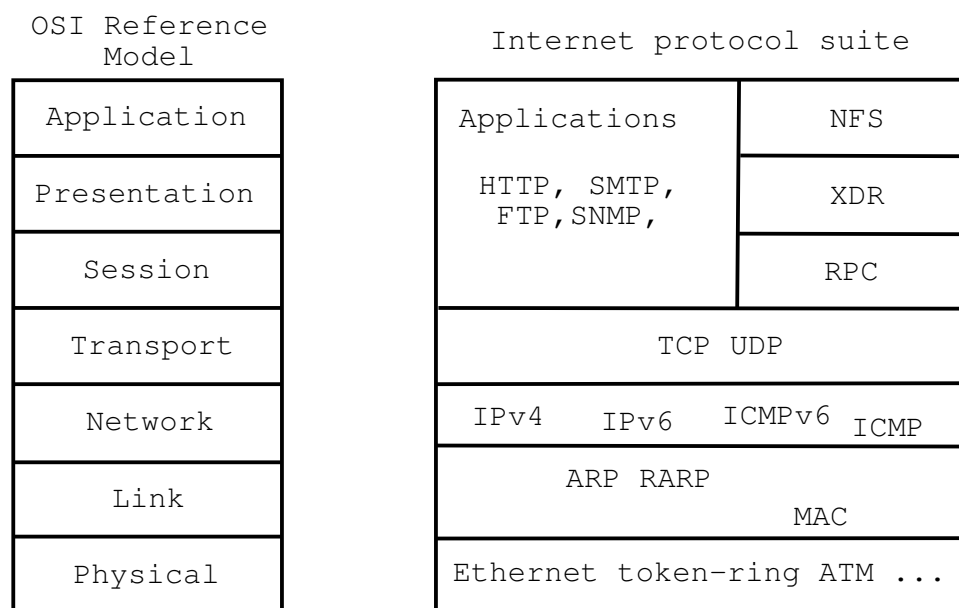


Figure 1: OSI og Internet Protocol suite

## Exercise content

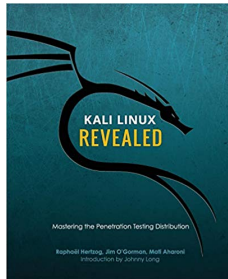
Most exercises follow the same procedure and has the following content:

- **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. Download PDF from: <https://kali.training/>

Direct link to PDF: <https://kali.training/downloads/Kali-Linux-Revealed-1st-edition.pdf>

**Solution:**

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

**Discussion:**

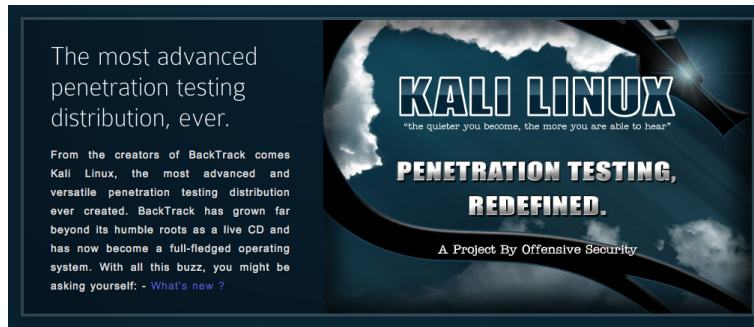
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

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

### **i** Download Debian Administrators Handbook (DEB) Book 10 min



#### **Objective:**

We need a Linux for running some tools during the course. I have chosen Debian Linux as this is open source, and the developers have released a whole book about running it.

This book is named The Debian Administrators Handbook, - shortened DEB

#### **Purpose:**

We need to install Debian Linux in a few moments, so better have the instructions ready.

#### **Suggested method:**

Create folders for educational materials. Go to download from the link <https://debian-handbook.info/> Read and follow the instructions for downloading the book.

#### **Solution:**

When you have a directory structure for download for this course, and the book DEB 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.

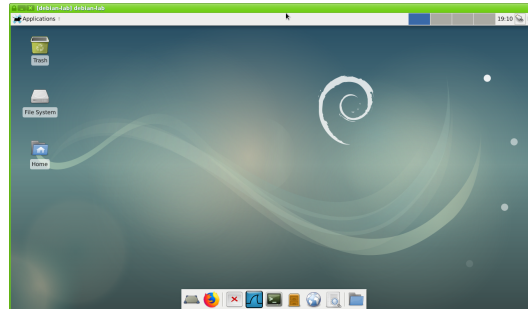
Debian Linux is a free operating system platform.

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

Not curriculum but explains how to use Debian Linux

## Exercise 4

### **i** Check your Debian VM 10 min



#### **Objective:**

Make sure your virtual machine is in working order.

We need a Debian 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 Debian VM.

**This is a bonus exercise - only one Debian is needed per team.**

#### **Hints:**

If you allocate enough memory and disk you won't have problems.

**I suggest 50G disk, 2CPU cores and 6Gb memory for this course, if you have this.**

#### **Solution:**

When you have a updated virtualisation software and a running VM, 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.

Debian Linux allows us to run Ansible and provision a whole SIEM in very few minutes.

## Exercise 5

### **i** Investigate /etc 10 min

#### **Objective:**

We will investigate the /etc directory on Linux. We need a Debian Linux

#### **Purpose:**

Start seeing example configuration files, including:

- User database /etc/passwd and /etc/group
- The password database /etc/shadow

#### **Suggested method:**

Boot your Linux VMs, log in

Investigate permissions for the user database files `passwd` and `shadow`

#### **Hints:**

Linux has many tools for viewing files, the most efficient would be `less`.

```
user@debian:~$ cd /etc
user@debian:/etc$ ls -l shadow passwd
-rw-r--r-- 1 root root   2203 Mar 26 17:27 passwd
-rw-r----- 1 root shadow 1250 Mar 26 17:27 shadow
user@debian:/etc$ ls
... all files and directories shown, investigate more if you like
```

Showing a single file: `less /etc/passwd` and press `q` to quit

Showing multiple files: `less /etc/*` then `:n` for next and `q` for quit

Trying reading the shadow file as your regular user:

```
user@debian:/etc$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Why is that? Try switching to root, using `su` or `sudo`, and redo the command.

#### **Solution:**

When you have seen the most basic files you are done.

Also note the difference between running as root and normal user. Usually books and instructions will use a prompt of hash mark `#` when the root user is assumed and dollar sign `$` when a normal user prompt.

#### **Discussion:**

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

Sudo is a tool often used for allowing users to perform certain tasks as the super user. The tool is named from superuser do! <https://en.wikipedia.org/wiki/Sudo>

## Exercise 6

### ⚠ Enable firewall - 15min

#### Objective:

Turn on a firewall and configure a few simple rules.

#### Purpose:

See how easy it is to restrict incoming connections to a server.

#### Suggested method:

Install a utility for firewall configuration.

You should also perform Nmap port scan with the firewall enabled and disabled.

#### Hints:

Using the ufw package it is very easy to configure the firewall on Linux.

Install and configuration can be done using these commands.

```
root@debian01:~# apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ufw
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 164 kB of archives.
After this operation, 848 kB of additional disk space will be used.
Get:1 http://mirrors.dotsrc.org/debian stretch/main amd64 ufw all 0.35-4 [164 kB]
Fetched 164 kB in 2s (60.2 kB/s)
...
root@debian01:~# ufw allow 22/tcp
Rules updated
Rules updated (v6)
root@debian01:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@debian01:~# ufw status numbered
Status: active
```

	To	Action	From
	--	-----	----
[ 1]	22/tcp	ALLOW IN	Anywhere
[ 2]	22/tcp (v6)	ALLOW IN	Anywhere (v6)

Also allow port 80/tcp and port 443/tcp - and install a web server. Recommend Nginx `apt-get install nginx`

#### Solution:

When firewall is enabled and you can still connect to Secure Shell (SSH) and web service, you are done.

**Discussion:**

Further configuration would often require adding source prefixes which are allowed to connect to specific services. If this was a database server the database service should probably not be reachable from all of the Internet.

Web interfaces also exist, but are more suited for a centralized firewall.

Configuration of this firewall can be done using ansible, see the documentation and examples at [https://docs.ansible.com/ansible/latest/modules/ufw\\_module.html](https://docs.ansible.com/ansible/latest/modules/ufw_module.html)

Should you have both a centralized firewall in front of servers, and local firewall on each server? Discuss within your team.

## Exercise 7

### ⚠ Git tutorials - 15min



#### Objective:

Try the program Git locally on your workstation

#### Purpose:

Running Git will allow you to clone repositories from others easily. This is a great way to get new software packages, and share your own.

Git is the name of the tool, and Github is a popular site for hosting git repositories.

#### Suggested method:

Run the program from your Linux VM. You can also clone from your Windows or Mac OS X computer. Multiple graphical front-end programs exist too.

Most important are Git clone and pull:

```
user@Projects:tt$ git clone https://github.com/kramse/kramse-labs.git
Cloning into 'kramse-labs'...
remote: Enumerating objects: 283, done.
remote: Total 283 (delta 0), reused 0 (delta 0), pack-reused 283
Receiving objects: 100% (283/283), 215.04 KiB | 898.00 KiB/s, done.
Resolving deltas: 100% (145/145), done.

user@Projects:tt$ cd kramse-labs/

user@Projects:kramse-labs$ ls
LICENSE README.md core-net-lab lab-network suricatazeek work-station
user@Projects:kramse-labs$ git pull
Already up to date.
```

#### Hints:

Browse the Git tutorials on <https://git-scm.com/docs/gittutorial> and <https://guides.github.com/activities/hello-world/>

We will not do the whole tutorials within 15 minutes, but get an idea of the command line, and see examples. Refer back to these tutorials when needed or do them at home.

Note: you don't need an account on Github to download/clone repositories, but having an account allows you to save repositories yourself and is recommended.

#### Solution:

When you have tried the tool and seen the tutorials you are done.

#### Discussion:

Before Git there has been a range of version control systems, see [https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control) for more details.

## Exercise 8

### **i** Use Ansible to install Elastic Stack

#### **Objective:**

Install and run Elasticsearch with additional programs – the elastic stack. Previously named ELK stack after Elasticsearch, Logstash and Kibana.

#### **Warning: This may become a bit long and we have other options.**

If you are not prepared to do a lot of work. On the other hand, this could be used as the basis for a production setup. YMMV.

#### **Purpose:**

See an example tool used for many projects, Elasticsearch from the Elastic Stack

#### **Suggested method:**

We will run Elasticsearch, either using the method from:

<https://www.elastic.co/guide/en/elastic-stack-get-started/current/get-started-elastic-stack.html>

or by the method described below using Ansible - your choice.

Ansible used below is a configuration management tool <https://www.ansible.com/> and you can adjust them for production use!

I try to test my playbooks using both Ubuntu and Debian Linux, but Debian is the main target for this training.

First make sure your system is updated, as root run:

```
apt-get update && apt-get -y upgrade && apt-get -y dist-upgrade
```

You should reboot if the kernel is upgraded :-)

Second make sure your system has ansible and my playbooks: (as root run)

```
apt -y install ansible git python
git clone https://github.com/kramse/kramse-labs
```

We will run the playbooks locally, while a normal Ansible setup would use SSH to connect to the remote node.

Then it should be easy to run Ansible playbooks, like this: (again as root, most packet sniffing things will need root too later)

```
cd kramse-labs/suricatazeek
ansible-playbook -v 1-dependencies.yml 2-suricatazeek.yml 3-elasticstack.yml 4-configuration.yml
```

Note: I keep these playbooks flat and simple, but you should investigate Ansible roles for real deployments.

If I update these, it might be necessary to update your copy of the playbooks. Run this while you are in the cloned repository:

```
git pull
```

Note: usually I would recommend running git clone as your personal user, and then use sudo command to run some commands as root. In a training environment it is OK if you want to run everything as root. Just beware.

Note: these instructions are originally from the course

Go to <https://github.com/kramse/kramse-labs/tree/master/suricatazeek>

### Hints:

Ansible is great for automating stuff, so by running the playbooks we can get a whole lot of programs installed, files modified - avoiding the Vi editor ☺

Example playbook content, installing software using APT:

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

### Solution:

When you have a updated VM and Ansible running, 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 9

### Configure Elasticsearch Security

#### Objective:

Configure the security settings for Elastic stack – mainly Kibana and Elasticsearch communication.

#### Purpose:

Getting access to dashboards and logging data should only be allowed for security administrators.

#### Suggested method:

When starting Kibana there is a small popup which guide you to the minimal setup. There are multiple steps and require some Linux/unix knowledge to perform them all. There is no requirement to perform this, if you are new to Unix and Elasticsearch, but make a mental note that it is available.

Currently:

<https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html>

#### Two settings are needed for elasticsearch

On every node in your cluster, add the `xpack.security.enabled` setting to the `/etc/elasticsearch/elasticsearch.yml` file and set the value to true, change from false to true:

```
xpack.security.enabled: true
```

and at the bottom add:

```
discovery.type: single-node
```

Then restart the elasticsearch service using `service elasticsearch restart`

Afterwards generate random passwords using:

```
/usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
```

This will print a number of passwords, which should be saved in a text file!

#### Two actions are needed for kibana

Add the `elasticsearch.username` setting to the file `/etc/kibana/kibana.yml` and set the value to the `kibana_system` user:

```
elasticsearch.username: "kibana_system"
```

Add the password for the `kibana_system` user to the Kibana keystore:

```
# /usr/share/kibana/bin/kibana-keystore create
# /usr/share/kibana/bin/kibana-keystore add elasticsearch.password
```

Then restart the kibana service using `service kibana restart`

Log in to Kibana as the elastic user.

Congratulate yourself if this works, good job.

You enabled password protection for your local cluster to prevent unauthorized access. You can log in to Kibana securely as the elastic user and create additional users and roles.

**Hints:**

You should be able to change a few lines of configuration using an editor, but for production use it is recommended to use Ansible or some other configuration management.

**Solution:**

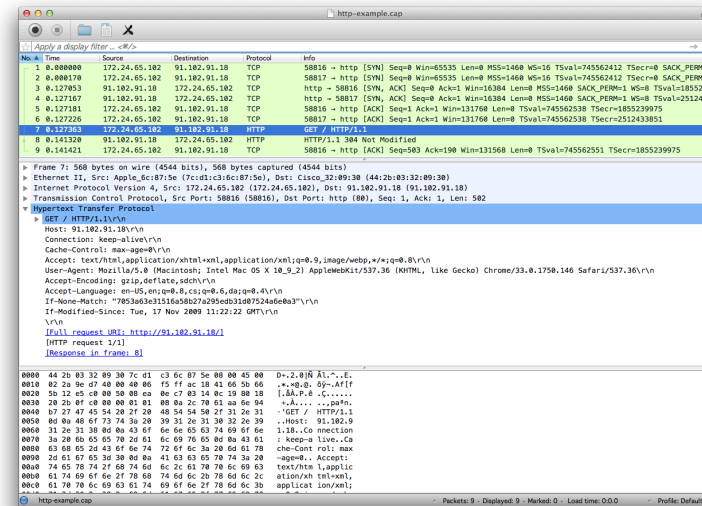
When you have read the instructions, and understood that Elastic stack includes some security settings, available and highly recommended for production use, you are done.

**Discussion:**

We could have added these settings and commands to Ansible playbooks, but since it is something that must be done differently in production clusters I considered it a good exercise.

## Exercise 10

### 🚨 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.

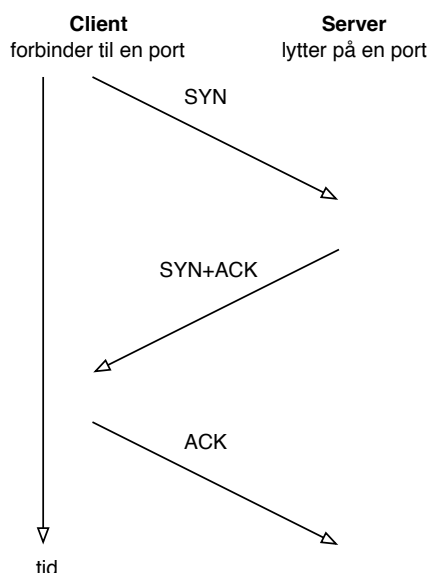
**Discussion:**

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://old.zeeq.org/community/traces.html>

## Exercise 11

### ⚠ 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 12

### 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 13

### ⚠ IP address research 30 min

**Objective:**

Work with IP addresses

**Purpose:**

What is an IP address?

Investigate the following IP addresses

- 192.168.1.1
- 192.0.2.0/24
- 172.25.0.1
- 182.129.62.63
- 185.129.62.63

Write down everything you can about them!

**Suggested method:**

Search for the addresses, look for web sites that may help.

**Hints:**

Download the fun guide from Julia Evans (b0rk) <https://jvns.ca/networking-zine.pdf>

Pay attention to Notation Time page

Lookup **ripe.net** they may have a service called stats or stat – something like that.

What is the Torproject? good, bad, neutral?

**Solution:**

When you have found some information about each of the above, say 2-3 facts about each you are done.

**Discussion:**

IP addresses are much more than an integer used for addressing system interfaces and routing packets.

We will later talk more about IP reputation

## Exercise 14

### ⚠ 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 on 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
```

Don't 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.



What's the difference between:

- **traceroute** and **traceroute -I**
- NB: traceroute -I is found on UNIX - traceroute using ICMP packets
- 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 15

### ⚠ 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 16

### ⚠️ 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 or failure. Allows very specific packets to be sent. It is part of the Nmap package.

#### 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
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
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 17

### **i** 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:**

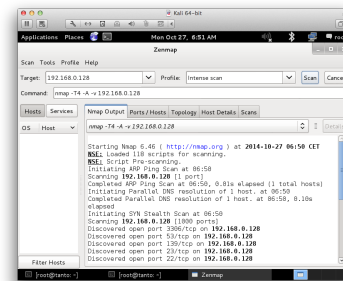
When you or your team mate has a running pcap-diff then you are done

**Discussion:**

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

## Exercise 18

### ⚠ 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:

Due to python version 2 being deprecated there are some missing files for running the tool Zenmap. This can be fixed by using the Kali Kaboxer `apt install zenmap-kbx`

Process is described in various posts around the internet. <https://www.kali.org/blog/introducing-kaboxer/>

The command to run Zenmap becomes: `zenmap-kbx`

As always try typing zen and press TAB twice, will show all commands starting with these letters.

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 19

### ⚠ 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 20

### ⚠ 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`

**Hints:**

The nmap tool 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 it can detect the family and in some cases it can identify the exact patch level of the system.

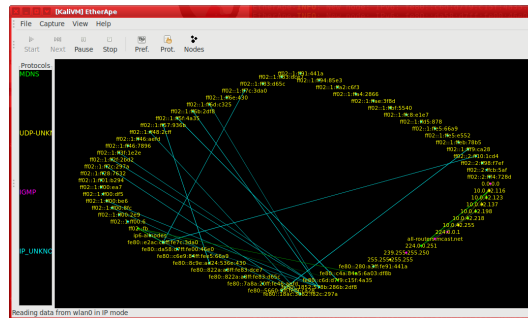
Better to use `-A` all the time, includes even more scripts and advanced stuff You can also save prefixes to scan in a text file, I usually name it `targets`

I also recommend adding `-oA` for writing output files. So a regular Nmap command might be:  
`nmap -p 1-65535 -A -oA full-tcp-scan -iL targets`



## Exercise 21

### **i** EtherApe 10 min



EtherApe is a graphical network monitor for Unix modeled after ethernan. 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 Wi-Fi network card may not be the best for sniffing in promiscuous and monitor mode.

Put your network card for the virtual system into bridging mode, and produce some data using Nmap ping scanning. Something like this for your local network `nmap -sP 192.168.0.0/24`

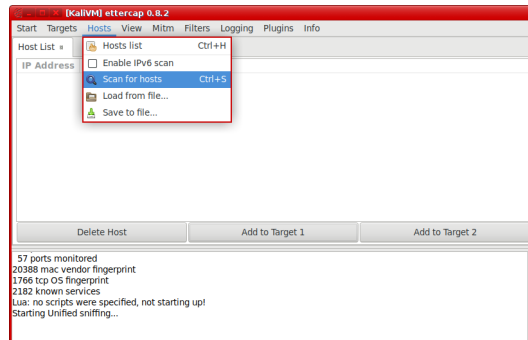
#### **Solution:**

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

#### **Discussion:**

## Exercise 22

### **i** ARP spoofing and ettercap 20min



#### **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.

You might ruin the network temporarily for others when playing with this, so be cautious.

#### **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 23

### **i** TCP SYN flooding 30min

#### **Objective:**

Start a webserver attack using SYN flooding tool hping3.

#### **Purpose:**

See how easy it is to produce packets on a network using hacker programs.

The tool we will use is very flexible and can produce ICMP, UDP and TCP using very few options. This tool is my primary one for doing professional DDoS testing.

```
-1 --icmp
    ICMP mode, by default hping3 will send ICMP echo-request, you can set other ICMP
    type/code using --icmptype --icmpcode options.

-2 --udp
    UDP mode, by default hping3 will send udp to target host's port 0.  UDP header  tunable
    options are the following: --baseport, --destport, --keep.
```

TCP mode is default, so no option needed.

#### **Suggested method:**

Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

Try doing the most common attacks TCP SYN flood using hping3:

```
hping3 --flood -p 80 -S 10.0.45.12
```

You should see something like this:

```
HPING 10.0.45.12: NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.45.12 hping statistic ---
352339 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

You can try different ports with TCP flooding, try port 22/tcp or HTTP(S) port 80/tcp and 443/tcp

#### **Hints:**

The tool we use can do a lot of different things, and you can control the speed. You can measure at the server being attacked or what you are sending, commonly using ifpps or such programs can help.

By changing the speed we can find out how much traffic is needed to bring down a service. This measurement can then be re-checked later and see if improvements really worked.

This allows you to use the tool to test devices and find the breaking point, which is more interesting than if you can overload, because you always can.

```
-i --interval
    Wait the specified number of seconds or micro seconds between sending each packet.
    --interval X set wait to X seconds, --interval uX set wait to X micro seconds. The de
    fault is to wait one second between each packet. Using hping3 to transfer files tune
    this option is really important in order to increase transfer rate. Even using hping3
    to perform idle/spoofing scanning you should tune this option, see HPING3-HOWTO for
    more information.

--fast Alias for -i u10000. Hping will send 10 packets for second.

--faster
    Alias for -i u1. Faster then --fast ;) (but not as fast as your computer can send pack
    ets due to the signal-driven design).

--flood
    Sent packets as fast as possible, without taking care to show incoming replies. This
    is ways faster than to specify the -i u0 option.
```

### Solution:

When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

### Discussion:

Gigabit Ethernet can send up to 1.4 million packets per second, pps.

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>

Receiving systems, and those en route to the service, should be checked for resources like CPU load, bandwidth, logging. Logging can also overload the logging infrastructure, so take care when configuring this in your own networks.

## Exercise 24

### **i** TCP other flooding 15min

**Objective:**

Start a webserver attack using TCP flooding tool hping3.

**Purpose:**

Run various other common attacks

TCP mode is default, so no option needed.

**Suggested method:**

Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

```
hping3 --flood -p 80 -R 10.0.45.12
```

You should see something like this:

```
HPING 10.0.45.12: NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.45.12 hping statistic ---
352339 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

**Hints:**

Common attacks use the SYN, as shown in previous exercise, but other popular TCP attacks are RST, PUSH, URG, FIN, ACK attacks - setting one or more flags in the packets.

-L	--setack	set TCP ack
-F	--fin	set FIN flag
-S	--syn	set SYN flag
-R	--rst	set RST flag
-P	--push	set PUSH flag
-A	--ack	set ACK flag
-U	--urg	set URG flag
-X	--xmas	set X unused flag (0x40)
-Y	--ymas	set Y unused flag (0x80)

**Solution:**

When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

**Discussion:**

If an attacker varies the packets they can be harder to filter out, and the attacks succeed.

## Exercise 25

### **i** UDP flooding NTP, etc. 15min

#### **Objective:**

Start a webserver attack using UDP flooding tool hping3.

#### **Purpose:**

See how easy it is to produce packets on a network using hacker programs.

The tool we will use is very flexible and can produce ICMP, UDP and TCP using very few options. This tool is my primary one for doing professional DDoS testing.

This time we will select UDP mode:

```
-2 --udp
    UDP mode, by default hping3 will send udp to target host's port 0.  UDP header  tunable
    options are the following: --baseport, --destport, --keep.
```

#### **Suggested method:**

Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

```
hping3 --flood -2 -p 53 10.0.45.12
```

#### **Hints:**

Try doing the most common attacks:

- UDP flooding, try port 53/udp DNS, 123/udp NTP and port 161/udp SNMP

#### **Solution:**

When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

#### **Discussion:**

Many networks don't send and receive a lot of UDP traffic. If you measure a baseline of the protocols needed on a daily basis you might be able to configure a profile for normal usage, and filter out bad traffic in case of attacks.

A starting point might be to allow full bandwidth for TCP, 10% UDP and 1% ICMP. This will ensure that even if an attacker is sending more than 1% ICMP only a fraction reaches your network and systems.

This is especially effective for protocols like ICMP which is not used for large data transfers.

## Exercise 26

### **i** ICMP flooding 15min

#### **Objective:**

Start a webserver attack using ICMP flooding tool hping3.

#### **Purpose:**

See how easy it is to produce packets on a network using hacker programs.

The tool we will use is very flexible and can produce ICMP, UDP and TCP using very few options. This tool is my primary one for doing professional DDoS testing.

This time we will select UDP mode:

```
-1 --icmp
    ICMP mode, by default hping3 will send ICMP echo-request, you can set other ICMP
    type/code using --icmptype --icmpcode options.
```

#### **Suggested method:**

Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

Try doing the most common attack:

- ICMP flooding with echo

```
hping3 --flood -1 10.0.45.12
```

#### **Hints:**

Common attacks use ICMP ECHO, but other types can be sent in the packets.

```
ICMP
-C --icmptype icmp type (default echo request)
-K --icmpcode icmp code (default 0)
--force-icmp send all icmp types (default send only supported types)
--icmp-gw set gateway address for ICMP redirect (default 0.0.0.0)
--icmp-ts Alias for --icmp --icmptype 13 (ICMP timestamp)
--icmp-addr Alias for --icmp --icmptype 17 (ICMP address subnet mask)
--icmp-help display help for others icmp options
```

#### **Solution:**

When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

#### **Discussion:**

If you have a 10G network connection, do you REALLY need 10Gbps of ICMP traffic?

Probably not, and routers can often filter this in wirespeed.

Routers have extensive Class-of-Service (CoS) tools today and a starting point might be as shown in Juniper Junos policer config:

```
term limit-icmp {  
    from {  
        protocol icmp;  
    }  
    then {  
        policer ICMP-100M;  
        accept;  
    }  
}  
term limit-udp {  
    from {  
        protocol udp;  
    }  
    then {  
        policer UDP-1000M;  
        accept;  
    }  
}
```

This effectively limit the damage an attacker can do. Your firewall and IDS devices will be free to spend more processing on the remaining protocols.



## Exercise 27

### Misc - stranger attacks 15min

Various other attacks are possible, sending illegal combinations of flags etc.

#### Objective:

Start a webserver attack using the packet generator and flooding tool t50.

#### Purpose:

See how easy it is to produce packets on a network using hacker programs.

The tool we will use is very flexible and can produce ICMP, UDP and TCP using very few options. This tool is another primary one for doing professional DDoS testing.

Apart from TCP,UDP and ICMP this tool can also produce packets for dynamic routing testing, OSPF, EIGRP and other esoteric RSVP, IPSEC, RIP and GRE.

```
$ t50 -help
T50 Experimental Mixed Packet Injector Tool v5.8.3
Originally created by Nelson Brito <nbrito@sekure.org>
Previously maintained by Fernando Mercês <fernando@mentebinaria.com.br>
Maintained by Frederico Lambert Pissarra <fredericopissarra@gmail.com>

Usage: t50 <host[/cidr]> [options]
Common Options:
  --threshold NUM      Threshold of packets to send      (default 1000)
  --flood              This option supersedes the 'threshold'
  --encapsulated       Encapsulated protocol (GRE)         (default OFF)
  -B,--bogus-csum      Bogus checksum                     (default OFF)
  --shuffle            Shuffling for T50 protocol          (default OFF)
  -q,--quiet           Disable INFOs
  --turbo              Extend the performance              (default OFF)
  -l,--list-protocols  List all available protocols
  -v,--version         Print version and exit
  -h,--help            Display this help and exit
...
Some considerations while running this program:
1. There is no limitation of using as many options as possible.
2. Report t50 bugs at https://gitlab.com/fredericopissarra/t50.git.
3. Some header fields with default values MUST be set to '0' for RANDOM.
4. Mandatory arguments to long options are mandatory for short options too.
5. Be nice when using t50, the author DENIES its use for DoS/DDoS purposes.
6. Running t50 with '--protocol T50' option sends ALL protocols sequentially.
```

#### Suggested method:

Connect to the LAB network using Ethernet! Borrow a USB network card if you dont have one.

Start your Kali VM in bridged mode, try a basic TCP flooding attack against the server provided by the instructor, or your own Debian server.

Run the help page, and browse options.

```
t50 -h
```

#### Hints:

The tools we use can do a lot of different things and using the command line options can produce high speed packet attacks without having to program in C ourselves.

Try doing a special attack:

- t50 with '-protocol T50' option sends ALL protocols, so try:  
`t50 --protocol T50 10.0.45.12`

**Solution:**

When your team has sent +1 million packets per second into the network, from one or two laptops - you are done.

**Discussion:**

Gigabit Ethernet can send up to 1.4 million packets per second, pps.

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>

Receiving systems, and those en route to the service, should be checked for resources like CPU load, bandwidth, logging. Logging can also overload the logging infrastructure, so take care when configuring this in your own networks.

## Exercise 28

### 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.

Notice how the output changes if you enable/disable the firewall on the system under test. Nmap tries to report open, filtered and closed in a sensible way. So if most ports are filtered and scanning 100 ports it might say 98 filtered, 1 open and 1 close for instance.

**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 29

### ⚠ SSH scanners - 15min

#### Objective:

Try ssh scanners, similar to ssllscan and Nmap sshscan

#### Purpose:

We often need to find older systems with old settings.

#### Suggested method:

Use Nmap with built-in scripts for getting the authentication settings from SSH servers

#### Hints:

Nmap includes lots of scripts, look into the directory on Kali:

```
$ ls /usr/share/nmap/scripts/*ssh*
/usr/share/nmap/scripts/ssh2-enum-algos.nse  /usr/share/nmap/scripts/ssh-publickey-acceptance.nse
/usr/share/nmap/scripts/ssh-auth-methods.nse /usr/share/nmap/scripts/ssh-run.nse
/usr/share/nmap/scripts/ssh-brute.nse       /usr/share/nmap/scripts/sshv1.nse
/usr/share/nmap/scripts/ssh-hostkey.nse
```

```
$ sudo nmap -A -p 22 --script "ssh2-enum-algos,ssh-auth-methods" 10.0.45.2
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-20 08:46 CET
Nmap scan report for 10.0.42.6
Host is up (0.0038s latency).
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco/3com IPSSHd 6.6.0 (protocol 2.0)
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
| ssh2-enum-algos:
|   kex_algorithms: (1)
|     diffie-hellman-group1-sha1
|   server_host_key_algorithms: (1)
|     ssh-dss
|   encryption_algorithms: (6)
|     aes128-cbc
|     aes192-cbc
|     aes256-cbc
|     blowfish-cbc
|_    cast128-cbc
|_    3des-cbc
|   mac_algorithms: (4)
|     hmac-sha1
|     hmac-sha1-96
|     hmac-md5
|     hmac-md5-96
|   compression_algorithms: (1)
|_    none
```

#### Solution:

When you have tried running against one or two SSH servers, you are done.

#### Discussion:

I recommend disabling password login on systems connected to the internet.

Having only public key authentication reduces or even removes the possibility for brute force attacks succeeding.

I also move the service to a random high port, which then requires an attacker must perform port scan to find it - more work.

Thus a better and more modern OpenSSH would look like this:

```
PORT      STATE SERVICE VERSION
4xxxx/tcp open  ssh      OpenSSH 7.9 (protocol 2.0)
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
| ssh2-enum-algos:
|   kex_algorithms: (4)
|     curve25519-sha256@libssh.org
|     diffie-hellman-group16-sha512
|     diffie-hellman-group18-sha512
|     diffie-hellman-group14-sha256
|   server_host_key_algorithms: (4)
|     rsa-sha2-512
|     rsa-sha2-256
|     ssh-rsa
|     ssh-ed25519
|   encryption_algorithms: (6)
|     chacha20-poly1305@openssh.com
|     aes128-ctr
|     aes192-ctr
|     aes256-ctr
|     aes128-gcm@openssh.com
|     aes256-gcm@openssh.com
|   mac_algorithms: (3)
|     umac-128-etm@openssh.com
|     hmac-sha2-256-etm@openssh.com
|     hmac-sha2-512-etm@openssh.com
|   compression_algorithms: (2)
|     none
|_    zlib@openssh.com
```

## Exercise 30

### **i** Configure SSH keys for more secure access

#### **Objective:**

See how SSH keys can be used.

#### **Purpose:**

Secure Shell is a very powerful administration tool. Administrators use this for managing systems. If an attacker gains access they can perform the same tasks.

Using SSH keys for access and disabling password based logins effectively prevents brute-force login attacks from succeeding.

#### **Suggested method:**

First generate a SSH key, use:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hlk/.ssh/id_rsa.
Your public key has been saved in /home/hlk/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:l5esp66lQArF0lXq0oHnxpg8zRS6shK8nx9KGf+oSp4 root@debian01
The key's randomart image is:
+---[RSA 2048]-----+
|      .              |
|    . o              |
|    . =             |
|  .. =.      o .    |
| o.*o. . S o +      |
| oB==+o    . o      |
| +*B=.o.    o .      |
| =++.o +. o o        |
| oEo=oo .ooo         |
+-----[SHA256]-----+
```

Then use the utility tool `ssh-copy-id` for copying the public key to the server. Install tool if not available using `apt` :

```
hlk@kunoichi:hlk$ ssh-copy-id -i /home/hlk/.ssh/id_rsa hlk@10.0.42.147
/usr/local/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/kramse.pub"
The authenticity of host '10.0.42.147 (10.0.42.147)' can't be established.
ECDSA key fingerprint is SHA256:DP6jqadDWEJW/3FY84cpTKmEW7XoQ4zDNf/RdTu6M.
Are you sure you want to continue connecting (yes/no)? yes
/usr/local/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
```

```
/usr/local/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you  
are prompted now it is to install the new keys  
h1k@10.0.42.147's password:
```

```
Number of key(s) added:      1
```

Now try logging into the machine, with: `"ssh -o 'IdentitiesOnly yes' 'h1k@10.0.42.147'"`  
and check to make sure that only the key(s) you wanted were added.

This is the best tool for the job!

The public must exist in the `authorized_keys` file, in the right directory, with the correct permissions  
... use `ssh-copy-id`

**Hints:**

You can publish the public part of your SSH keys in places such as Github and Ubuntu installation  
can fetch this during install, making the use of SSH keys extremely easy.

**Solution:**

When you can login using key you are done.

**Discussion:**

We have not discussed using passphrase on the key, neither how to turn off password based logins by  
reconfiguring the SSH daemon. This is left as an exercise for the reader.

## Exercise 31

### ⚠ 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 32

### **i** Reporting HTML 15 min

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

Scan Summary | [www.zencurify.com](http://www.zencurify.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 zencurify-web www.zencurify.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.zencurify.com](http://www.zencurify.com)

### Address

- 185.129.60.130 (ipv4)

### Hostnames

- [www.zencurify.com](http://www.zencurify.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 greppable formats at once.

#### Suggested method:

First run Nmap, with output, then process it

```
sudo nmap -oA zencurify-web www.zencurify.com
xsltproc zencurify-web.xml > zencurify-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 33

### ⚠ SSL/TLS scanners 15 min

**Objective:**

Try the Online Qualys SSL Labs scanner <https://www.ssllabs.com/> Try the command line tool `sslsan` 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 www.kramse.dk
Version: 1.10.5-static
OpenSSL 1.0.2e-dev xx XXX xxxx

Testing SSL server www.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 SMTP TLS if possible.

**Hints:**

Originally `sslsan` 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 SSL Labs only can test from hostname

## Exercise 34

### Internet scanners 15 min

**Objective:**

Try the Online scanners <https://internet.nl/> and a few more.

**Purpose:**

Learn how to efficiently check settings on remote services.

**Suggested method:**

There are multiple portals and testing services which allow you to check a domain, mail settings or web site.

Run tools against a couple of sites of your choice.

- <https://internet.nl/> Generic checker
- <https://www.hardenize.com/> Generic checker
- [https://www.wormly.com/test\\_ssl](https://www.wormly.com/test_ssl) Test TLS
- <https://observatory.mozilla.org/> Web site headers check
- <https://dnsviz.net/> DNS zone check
- <https://rpki.cloudflare.com/> Check RPKI - route validator enter IP address  
More information about this: [https://labs.ripe.net/author/nathalie\\_nathalie/rpki-test/](https://labs.ripe.net/author/nathalie_nathalie/rpki-test/)

Others exist, feel free to suggest some.

**Hints:****Solution:**

When you can run and understand what at least one tool does, you are done.

**Discussion:**

Which settings are most important, which settings are your responsibility?

## Exercise 35

### 📌 Nginx as a Transport Layer Security (TLS) endpoint 40 min

#### Objective:

Read how to configure Nginx with TLS locally on your workstation This is a longer exercise. Feel free to complete this exercise at home.

#### Purpose:

Web services with TLS is a requirement in many circumstances. Unfortunately having TLS enabled requires both certificates, settings and large software packages like OpenSSL. A lot of vulnerabilities have been found in these and updating them may prove hard.

Having a centralized entry where TLS is served to the internet may help you.

#### Suggested method:

Run the programs from your Debian Linux VM, use `apt install nginx` if not already installed.

Follow a guide like the one from Nginx:

[http://nginx.org/en/docs/http/configuring\\_https\\_servers.html](http://nginx.org/en/docs/http/configuring_https_servers.html)

Check using `ssllscan` if your site is working, and configured according to best current practice.

#### Hints:

Note: above link does NOT show how to generate certificates and keys, so you need to find this yourself. A good place would be at certificate providers, search for Nginx CSR Certificate Signing Request – just dont order certificates.

A full blown tutorial from Digital Ocean:

<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-on-debian-10>

My kramse-labs also includes some example configs, check with `git pull`

#### Solution:

When you have configured an instance of Nginx you are done.

#### Discussion:

A great document about Transport Layer Security (TLS) is available from the web site of NCSC in the Netherlands:

<https://english.ncsc.nl/publications/publications/2021/january/19/it-security-guidelines-for-transport-layer-security-2.1>

Dont forget to add the recommended HTTP Strict Transport Security header to your configuration, if your site is in production.

[https://cheatsheetseries.owasp.org/cheatsheets/HTTP\\_Strict\\_Transport\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html)

A regular production site could also benefit from Lets Encrypt certificates updated automatically using one of the many clients available. Try searching for Lets Encrypt and Nginx.

## Exercise 36

### ⚠️ Nginx logging 20 min

**Objective:**

See the common log format used by web servers.

[https://en.wikipedia.org/wiki/Common\\_Log\\_Format](https://en.wikipedia.org/wiki/Common_Log_Format)

**Purpose:**

Knowing that a common format exist, allow you to choose between multiple log processors.

**Suggested method:**

Run Nginx on your Debian Linux VM and then check the logs.

```
# cd /var/log/nginx
# ls
# less access.log
# less error.log
```

Produce some bad logs using Nikto or using a browser, and check `error.log`

**Hints:**

A lot of scanning activities would result in error logs, so if you observe a rise in 404 not found or similar, then maybe you are being targetted.

**Solution:**

When you have tried the tool and seen some data you are done.

**Discussion:**

I commonly recommend tools like Packetbeat and other tools from Elastic to process logs, see <https://www.elastic.co/beats/packetbeat>

Another popular one is Matomo formerly known as Piwik  
<https://matomo.org/>.

## Exercise 37

### ⚠ Nginx filtering 40 min

#### Objective:

See how Nginx can filter a request easily.

#### Purpose:

Running Nginx with a filtering configuration can protect some resource, or part of a web site from attacks.

Example configuration:

```
server {
    listen      80;
    server_name service.dev;
    access_log  /var/log/nginx/access.log;
    error_log   /var/log/nginx/error.log debug;
    # Proxy settings
    proxy_set_header    Host $http_host;
    proxy_redirect      off;
    location / {
        # Catch all
        proxy_pass       http://127.0.0.1:81/;
    }
    location /admin/ {
        # /admin/ only
        allow 192.168.5.0/24;
        deny  all;
        proxy_pass       http://127.0.0.1:81/;
    }
}
```

Note; this does a proxy pass to another service locally, you may need to change it. Perhaps you can use another Nginx example running on port 3000.

If you use the JuiceShop running in Docker, and as a directory to disallow, perhaps the `/ftp/` one.

#### Suggested method:

Run the configuration from your Debian Linux VM

#### Hints:

My kramse-labs also includes some example configs, check with `git pull`

Having a negative list is bad, better to have a positive list of allowed requests.

#### Solution:

When you have tried above and seen Nginx block your request, you are done.

#### Discussion:

Multiple modules exist for Nginx, Apache, PHP etc. for blocking bad requests. Which one is right for your setup, you must research for yourselves.