What is a Secure Network PROSA exercises

Henrik Kramselund
hlk@zencurity.com

September 20, 2025

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

Exercises marked with ℹ are considered optional. These contain subjects that are related to the course, but less important. 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

© license CC BY 3.0. 2025 Henrik Kramselund

© license CC BY 3.0. 2025 Henrik Kramselund

Preface

This material was originally prepared for use in a workshop What is a Secure Network PROSA workshopnd was prepared by Henrik Kramselund, Zencurity. 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 prosa-secure-network-2025-exercises in the repository 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://codeberg.org/kramse/kramse-labs

Dont be scared away by the many exercises, you can pick a few and have fun with those and ignore the rest. If you one day need to research network security the others may come in handy ☺

The exercise list includes my recommended tools I use myself and for teaching.

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.

Have fun and learn

© license CC BY 3.0. 2025 Henrik Kramselund

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.

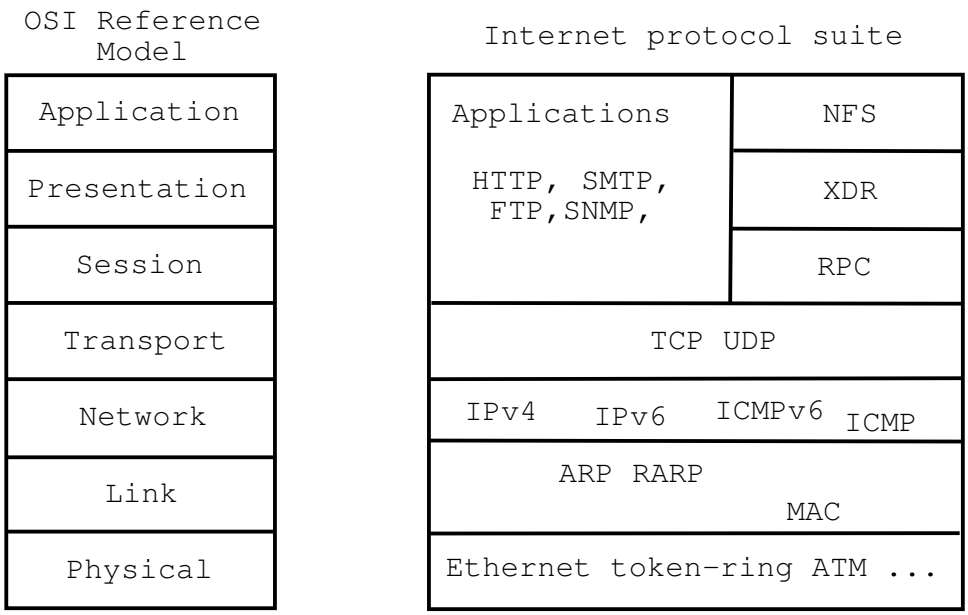


Figure 1: OSI og Internet Protocol suite

© license CC BY 3.0. 2025 Henrik Kramselund

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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 1

❶ 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:
Debian Linux is a mature Unix with great documentation. Kali Linux is based on Debian, so by learning Debian you can make infrastructure using Debian Linux, and test security using Kali Linux – and the administration will be the same commands.

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.

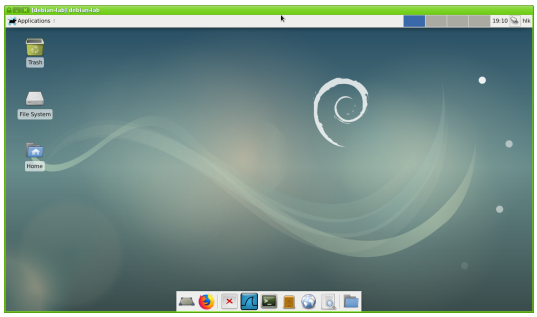
Debian Linux is a free operating system platform.

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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 2

❶ Check your Debian VM 10 min



Objective:
I use Debian as the base for exercises – they are tested on Debian AMD64 systems!

If you want to use Debian, make sure your virtual machine is in working order.

You dont need both a Debian Linux and Kali for running tools in this booklet, but some are better suited for either, so you can choose to install both.

Purpose:
If your VM is not installed and updated you might run into trouble later.

Suggested method:
Go to https://codeberg.org/kramse/kramse-labs/

Read the instructions for the setup of a Debian VM.

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

I suggest 50G disk, 2CPU cores and 4Gb 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 3

ℹ Configure Sudo

Objective:
Learn how to configure the tool Sudo to allow administrative commands.

Purpose:
Sudo is the most common method for switching from a normal user to root which is the administrative user for Unix. This command allows you to use your own password and execute highly privileged commands easily.

Suggested method:
Not in sudoers file, cannot run sudo command

This can be fixed quite easily.

If you use the su command first, to switch user to root and run the visudo command:

```
hlk@debian01:~$ su -
// enter password
# visudo
```

You will get an editor, where you enter below the root line, your username and a similar line:

```
# User privilege specification
root ALL = (ALL:ALL) ALL
hlk ALL = (ALL:ALL) NOPASSWD: ALL
```

In the example my user is `hlk`

Then use ctrl-x if using Nano, and exit the editor - saving this configuration file.

Hints:
Most books about Unix has a convention to use the dollar sign if you are logged in as a regular user and the hash tag sign when logged in as root.

```
hlk@debian01:~$ echo "Hi I am just a regular user"

root@debian01:~# echo "this is a command being run by root"
```

Solution:
When you can switch between your user and root you are done.

Discussion:
Note sudo has had many security vulnerabilities, so you should keep your system up to date using apt regularly, read about updates in the DEB book.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 4

❶ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

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`

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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 5

ℹ 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:
Put Git on your list of technologies to learn.

If you feel like it, try 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://codeberg.org/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.codeberg.org/activities/hello-world/

We will not do the tutorials, 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 acccount allows you to save repositories yourself and is recommended.

Solution:
When you have understood the importance of this 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 for more details.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 6

ℹ 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.

ℹ

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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 7

❶ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 8

❶ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 9

❶ 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.

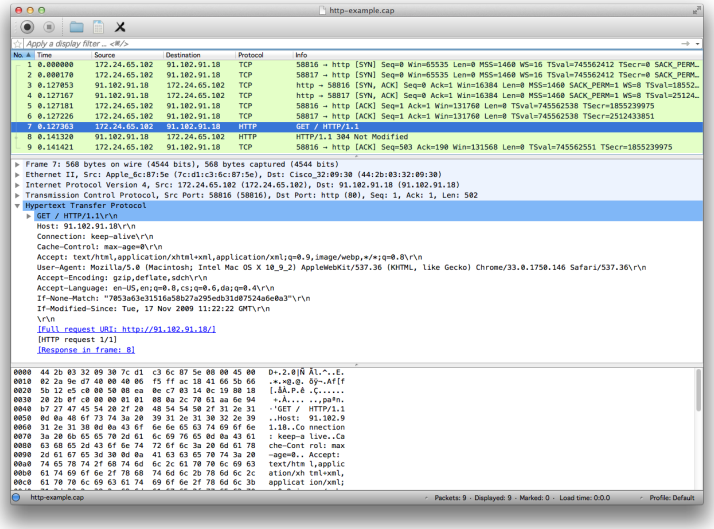
Discussion:
IP addresses are much more than an integer used for addressing system interfaces and routing packets.

We will later talk more about IP reputation

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 10

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

Suggested method:
Install Wireshark and/or Tcpdump from either their home page or using the Debian packages:
`sudo apt install wireshark tcpdump`

Run Wireshark or tcpdump to capture some packages.

The PPA book in 3rd edition page 44 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.

Use the command `ip address` to show interfaces, mine is enX0 – and then:

```
root@debian-lab:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
```

© license CC BY 3.0. 2025 Henrik Kramselund

```
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever{\bf
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000}
    link/ether 00:16:3e:5e:6c:00 brd ff:ff:ff:ff:ff:ff
    altname enx00163e5e6c00
    inet 10.137.0.51/24 brd 10.137.0.255 scope global noprefixroute enX0
       valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:fe5e:6c00/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
root@debian-lab:~# tcpdump -nei enX0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enX0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:55:10.091195 00:16:3e:5e:6c:00 > fe:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 98:
10.137.0.51 > 9.9.9.9: ICMP echo request, id 1, seq 1, length 64
13:55:10.105924 fe:ff:ff:ff:ff:ff > 00:16:3e:5e:6c:00, ethertype IPv4 (0x0800), length 98:
9.9.9.9 > 10.137.0.51: ICMP echo reply, id 1, seq 1, length 64
13:55:11.092277 00:16:3e:5e:6c:00 > fe:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 98:
10.137.0.51 > 9.9.9.9: ICMP echo request, id 1, seq 2, length 64
13:55:11.107360 fe:ff:ff:ff:ff:ff > 00:16:3e:5e:6c:00, ethertype IPv4 (0x0800), length 98:
9.9.9.9 > 10.137.0.51: ICMP echo reply, id 1, seq 2, length 64
13:55:12.094214 00:16:3e:5e:6c:00 > fe:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 98:
10.137.0.51 > 9.9.9.9: ICMP echo request, id 1, seq 3, length 64
13:55:12.110098 fe:ff:ff:ff:ff:ff > 00:16:3e:5e:6c:00, ethertype IPv4 (0x0800), length 98:
9.9.9.9 > 10.137.0.51: ICMP echo reply, id 1, seq 3, length 64
13:55:13.095628 00:16:3e:5e:6c:00 > fe:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 98:
10.137.0.51 > 9.9.9.9: ICMP echo request, id 1, seq 4, length 64
13:55:13.109071 fe:ff:ff:ff:ff:ff > 00:16:3e:5e:6c:00, ethertype IPv4 (0x0800), length 98:
9.9.9.9 > 10.137.0.51: ICMP echo reply, id 1, seq 4, length 64
```

I ran `ping 9.9.9.9` in another terminal window.

If you want to save packets use option `-w` like: `tcpdump -i enX0 -w my-capture.cap icmp`

Solution:
When Wireshark or Tcpdump is installed and you have captured some packets, you are done. 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.

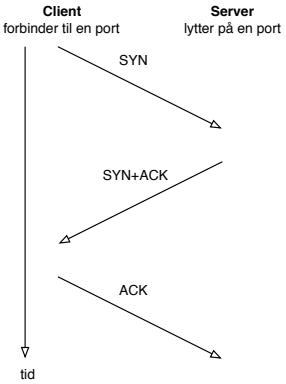
Discussion:
Wireshark is just an example other packet analyzers exist, some commercial and some open source like Wireshark

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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 11

Capturing network packets – 15min



Objective:
Sniff 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

```
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 packets 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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 12

Install Zeek on Your Debian – 15min

Objective:
See the installation of the Zeek tool.

Purpose:
We will run Zeek on the Web with small pcaps. Running Zeek locally will allow us to run larger packet captures, and/or use live traffic.

Suggested method:
Install Git and Ansible, `sudo apt install git ansible`

If you haven't already use Git to clone the kramse-labs repository:

```
user@Projects:tt$ git clone https://codeberg.org/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.
```

If you already cloned it, go into the folder and do a `git pull` to update your copy.

```
hlk@debian-lab:~$ cd kramse-labs/suricatazeek/
hlk@debian-lab:~/kramse-labs/suricatazeek$ git pull
Already up to date.
```

Then use Ansible to install Zeek with a few modifications:

- Add the OpenSuse Build Service repository, with key
- Install Zeek and some other tools
- Make a few modifications to the configuration: JSON output

```
hlk@debian-lab:~/kramse-labs/suricatazeek$ ansible-playbook 1-dependencies.yml 2-suricatazeek.yml

PLAY [run the playbook tasks on the localhost] *********************************************

TASK [Gathering Facts] *********************************************
ok: [127.0.0.1]

TASK [Exit if running Kali] *********************************************
skipping: [127.0.0.1]
```

```
TASK [lineinfile] ***************************************************
ok: [127.0.0.1]

TASK [Install a list of dependencies] *******************************************************
ok: [127.0.0.1]

PLAY RECAP *********************************************
127.0.0.1                  : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Hints:
Zeek installed in this way is the recommended way. This will allow you to update packages using regular Debian upgrade process: `apt update;apt upgrade`

Solution:
When you have read the exercise and understand the basics of the installation, you are done.

Don't worry if it didn't go well.

Discussion:
If using Ansible easier than copy pasting a lot of commands?

What method would you prefer?

Should the instructor have prepared a complete Virtual Machine image? a Container image?

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 13

❶ 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. It is part of the Nmap package.

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

```
root@debian:~# 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.

© license CC BY 3.0. 2025 Henrik Kramselund

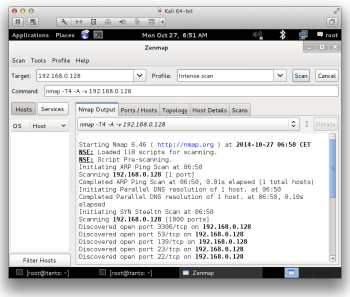
Things like changing the TTL may provoke ICMP messages, like this:

```
root@debian:~# 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=
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=
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=
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=
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=

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
```

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 14

ℹ 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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 15

❶ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 16

ℹ 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` – or `-A` which also add even more good stuff.

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 in 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`

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 17

❶ 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 professsional 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 hosts 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.

© license CC BY 3.0. 2025 Henrik Kramselund

```
-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://codeberg.org/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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 18

❶ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 19

❶ 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 professsional DDoS testing.

This time we will select UDP mode:

```
-2 --udp
       UDP mode, by default hping3 will send udp to target hosts 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 dont́ 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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 20

❶ 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 professsional 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:

© license CC BY 3.0. 2025 Henrik Kramselund

```
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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 21

❶ 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 professsional DDoS testing.

Apart from TCP,UDP and ICMP this tool can also produce packets for dynamic routing testting, 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 Lamberti 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://codeberg.org/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.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 22

ℹ 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 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 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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 23

❶ 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 checck 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` 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/`

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 responsebility?

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 24

ℹ Wireguard - 60 min

Objective:
Research the new wireguard between two Linux servers

Purpose:
Know there is alternative to connect servers securely.

Wireguard is very easy to setup, as it requires very little configuration.

Suggested method:
Find devices and operating systems that support Wireguard

Linux and OpenBSD can do it out-of-the-box https://www.wireguard.com/quickstart/

Hints:
They have a Demo Server which allows you to try out Wireguard

Solution:
When you have a working connection between them, you are done.

Discussion:
We wont do this in class, but I have a lot of friends that use Wireguard in production for complex setups.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 25

❶ Zeek on the web 10min

Objective:
Try Zeek Network Security Monitor - without installing it.

Purpose:
Show a couple of examples of Zeek scripting, the built-in language found in Zeek Network Security Monitor

Suggested method:
Go to `http://try.zeek.org/#/?example=hello` and try a few of the examples.

Hints:
The exercise The Summary Statistics Framework can be run with a specifc PCAP.

```
192.168.1.201 did 402 total and 2 unique DNS requests in the last 6 hours.
```

Solution:
You should read the example Raising a Notice. Getting output for certain events may be interesting to you.

Discussion:
Zeek Network Security Monitor is an old/mature tool, but can still be hard to get started using. I would suggest that you always start out using the packages available in your Ubuntu/Debian package repositories. They work, and will give a first impression of Zeek. If you later want specific features not configured into the binary packages, then install from source.

The tool was renamed in 2018 from Bro to Zeek. Some commands and files still reference the old names.

Also Zeek uses a zeekctl program to start/stop the tool, and a few config files which we should look at. From a Debian system they can be found in `/opt/zeek/etc` :

This is from the Debian package from the Zeek project, which is why it is using the path `/opt`.

`/opt/zeek/etc`:

```
root@debian-lab:/opt/zeek/etc# ls -la
total 24
drwxrwsr-x  3 root zeek 4096 Apr 16 20:03 .
drwxr-xr-x 10 root root 4096 Apr 16 20:03 ..
-rw-rw-r--  1 root zeek  262 Jan 28  2015 networks.cfg
-rw-rw-r--  1 root zeek  651 Jan 28  2015 node.cfg
-rw-rw-r--  1 root zeek 3052 Jan 28  2015 zeekctl.cfg
drwxr-xr-x  2 root zeek 4096 Apr 16 20:03 zkg
```

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 26

⚠ Zeek DNS capturing domain names – 15min

Objective:
We will now start using Zeek on our systems.

Purpose:
Try Zeek with example traffic, and see what happens.

Suggested method packet capture file:
Use Nitroba.pcap can be found in various places around the internet

```
hlk@debian-lab:~$ cd
hlk@debian-lab:~$ wget http://downloads.digitalcorpora.org/corpora/scenarios/2008-nitroba/nitroba.pcap
hlk@debian-lab:~$ mkdir $HOME/zeek;cd $HOME/zeek; zeek -r ../nitroba.pcap
... zeek reads the packets
hlk@debian-lab:~/zeek$ ls
conn.log  dns.log  dpd.log  files.log  http.log  packet_filter.log
sip.log  ssl.log  weird.log  x509.log
hlk@debian-lab:~/zeek$ less *
```

Use :n to jump to the next file in less, go through all of them.

Suggested method Live traffic:
Make sure Zeek is configured as a standalone probe and configured for the right interface. Linux used to use eth0 as the first ethernet interface, but now can use others, like ens192 or enx00249b1b2991.

```
root@debian:/opt/zeek/etc# cat node.cfg
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=eth0
...
```

Hints:
There are multiple commands for showing the interfaces and IP addresses on Linux. The old way is using `ifconfig -a` newer systems would use `ip a`

Note: if your system has a dedicated interface for capturing, you need to turn it on, make it available. This can be done manually using `ifconfig eth0 up`

Solution:
When you either run Zeek using a packet capture or using live traffic

Running with a capture can be done using a command line such as: `zeek -r traffic.pcap`

Using zeekctl to start it would be like this:

```
// Use the deploy command to initialize and start zeek first
debian:~ root# zeekctl

Welcome to ZeekControl 1.5
Type "help" for help.

[ZeekControl] > install
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
...
debian:etc root# grep eth0 node.cfg
interface=eth0
```

Afterwards you can stop and start as you wish:

```
[ZeekControl] > start
... starting zeek
// Exit using ctrl-d and then look at logs
debian:zeek root# cd /opt/zeek/logs/current
debian:zeek root# pwd
/opt/zeek/logs/current
debian:current root# tail -f dns.log
```

You should be able to spot entries like this:

```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto
trans_id        rtt query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name
AA      TC      RD      RA      Z       answers TTLs    rejected
1538982372.416180 CD12Dc1SpQm42QW4G3 10.xxx.0.145 57476 10.x.y.141 53        udp
20383       0.045021        www.dr.dk       1       C_INTERNET      1 A 0 NOERROR
F F T T 0 www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93 60.000000,20409.000000,20.000000 F
```

Note: this show ALL the fields captured and dissected by Zeek, there is a nice utility program zeek-cut which can select specific fields:

```
root@debian:/opt/zeek/logs/current# cat dns.log | zeek-cut -d ts query answers | grep dr.dk
2018-10-08T09:06:12+0200 www.dr.dk www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93
```

If your file is already in JSON format, you cannot use zeek-cut, but you can use other tools like jQuery `jq`.

Discussion:
Why is DNS interesting?

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 27

⚠ Zeek TLS capturing certificates – 15min

Objective:
Run more traffic through Zeek, see the various files.

Purpose:
See that even though HTTPS and TLS traffic is encrypted it often show names and other values from the certificates and servers.

Suggested method:
Run Zeek capturing live traffic, start https towards some sites. A lot of common sites today has shifted to HTTPS/TLS.

Hints:
use zeekctl start and watch the output directory

```
root@debian:/opt/zeek/logs/current# ls *.log
communication.log  dhcp.log files.log known_services.log packet_filter.log  stats.log
stdout.log x509.log conn.log dns.log known_hosts.log loaded_scripts.log  ssl.log
stderr.log weird.log
```

We already looked at `dns.log`, now check `ssl.log` and `x509.log`

Solution:
When you have multiple log files with data from Zeek, and have looked into some of them. You are welcome to ask questions and look into more files.

Discussion:
How can you hide that you are going to HTTPS sites?

Hint: VPN

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 28

⚠ Suricata from files – 15min


```
{
  "timestamp": "2008-07-22T03:51:08.754839+0200",
  "flow_id": 1271686344663190,
  "pcap_cnt": 84,
  "event_type": "dns",
  "src_ip": "192.168.1.64",
  "src_port": 2132,
  "dest_ip": "192.168.1.254",
  "dest_port": 53,
  "proto": "UDP",
  "pkt_src": "wire/pcap",
  "dns": {
    "version": 2,
    "type": "query",
    "id": 49569,
    "rrname": "www.blogger.com",
    "rrtype": "A",
    "tx_id": 0,
    "opcode": 0
  }
}
```


Objective:
Run traffic through Suricata like we did with Zeek, see the various output files.

Purpose:
See that the Suricata tools basically function similarly to Zeek. Main difference is that Zeek writes everything it decodes, while Suricata is often configured with a ruleset that allows for alerting about known bad things.

Suggested method:
Run Suricata like we did with Zeek on the Nitroba packet capture.

```
hlk@debian-lab:~$ cd
hlk@debian-lab:~$ wget http://downloads.digitalcorpora.org/corpora/scenarios/2008-nitroba/nitroba.pcap
... skip this command if you already downloaded previously
hlk@debian-lab:~$ mkdir $HOME/suri;cd $HOME/suri;suricata -l . -r ../nitroba.pcap -c /etc/suricata/suricata.yaml
i: suricata: This is Suricata version 7.0.10 RELEASE running in USER mode
W: detect: No rule files match the pattern /var/lib/suricata/rules/suricata.rules
W: detect: 1 rule files specified, but no rules were loaded!
W: unix-manager: Unix socket: UNIX socket bind(/var/run/suricata-command.socket) error: Address already in use
W: unix-manager: Unable to create unix command socket
i: threads: Threads created -> RX: 1 W: 2 FM: 1 FR: 1   Engine started.
i: suricata: Signal Received.  Stopping engine.
i: pcap: read 1 file, 94410 packets, 54670237 bytes
hlk@debian-lab:~/suri$ ls
eve.json  fast.log  stats.log  suricata.log
```

Read the eve.json using `less eve.json` or perhaps `cat eve.json | jq | less`

Hints:

© license CC BY 3.0. 2025 Henrik Kramselund

Solution:
When you have multiple log files with data from Suricata, and have looked into some of them. You are welcome to ask questions and look into more files.

Discussion:

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 29

ℹ Test a DNS server 30min

Objective:
Try communicating with DNS commands.

Purpose:
Learn how basic the setup of Unbound DNS server is.

Suggested method:
Setup Unbound on Debian using the instructions from
https://nlnetlabs.nl/documentation/unbound/howto-setup/

Note: there is a binary unbound package which can be installed with `apt install unbound` so use that, and config is then in `/etc/unbound` .

Use command line tools to test your server, if not available try ping - which will do a lookup. Retry the commands from exercise: **ℹ DNS and Name Lookups 10 min** which is number **7**.

The server to use is then `127.0.0.1`, use it to ask for a domain like this:

```
$ host www.dr.dk 127.0.0.1
```

Note: this command is in the Debian package `bind9-host` install with `sudo apt install bind9-host`

Hints:
The documentation for Unbound includes a small example with the access-control directive, which is the most important one for recursive use.

```
# unbound.conf for a local subnet.
server:
    interface: 0.0.0.0
    interface: ::0
    access-control: 192.168.0.0/16 allow
    access-control: ::1 allow
    verbosity: 1
```

Solution:
When you have tried sending a few DNS requests you are done, asking for a remote name, from a DNS server running on localhost: `host www.dr.dk 127.0.0.1`

Discussion:
How do we enforce new versions of protocol - as old as DNS?!

If you like, purge the Unbound software including config, with
`apt-get purge unbound`

and try installing Pi-Hole https://pi-hole.net/

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 30

❶ Configure a Mirror Port 10min

Objective:
Mirror ports are a way to copy traffic to Suricata and other devices - for analyzing it. Below are the steps document on a Juniper switch to show how. This could be used for an Intrusion Detection System (IDS).

Most switches which are configurable have this possibility.

This is a reading exercise – showing the steps one would go through to configure network wide packet capture

Purpose:
We want to capture traffic for multiple systems, so we select an appropriate port and copy the traffic. In our setup, we select the uplink port to the internet/router.

It is also possible to buy passive taps, like a fiber splitter, which then takes part of the signal, and is only observable if you look for signal strength on the physical layer.

Suggested method:
Configuring a mirror port on a Juniper EX2200-C running Junos could look like this:

```
root@ex2200-c# show ethernet-switching-options | display set
set ethernet-switching-options analyzer mirror01 input ingress interface ge-0/1/1.0
set ethernet-switching-options analyzer mirror01 input egress interface ge-0/1/1.0
set ethernet-switching-options analyzer mirror01 output interface ge-0/1/0.0
set ethernet-switching-options storm-control interface all
```

If using the TP-Link T1500G-10PS then this link should describe the process:
`https://www.tp-link.com/en/configuration-guides/mirroring_traffic/?configurationId=18210`

Which describe:

1. Choose the menu MAINTENANCE > Mirroring
2. Select Edit for the Mirror Session 1
3. In the Destination Port Config section, specify a destination port for the mirroring session, and click Apply
4. In the Source Interfaces Config section, specify the source interfaces and click Apply

Using the command line would be similar to this:

```
Switch#configure
Switch(config)#monitor session 1 destination interface gigabitEthernet 1/0/7
Switch(config)#monitor session 1 source interface gigabitEthernet 1/0/1-4 both
Switch(config)#monitor session 1 source cpu 1 both
```

© license CC BY 3.0. 2025 Henrik Kramselund

Hints:
When checking your own devices this is often called SPAN ports, Mirror ports or similar.
`https://en.wikipedia.org/wiki/Port_mirroring`

Solution:
When we can see the traffic from the network, we have the port configured - and can run any tool we like. Note: specialized capture cards can often be configured to spread the load of incoming packets onto separate CPU cores for performance. Capturing 100G and more can also be done using switches like the example found on the Zeek web site using an Arista switch 7150.

Discussion:
When is it ethical to capture traffic?

Cisco has called this Switched Port Analyzer (SPAN) or Remote Switched Port Analyzer (RSPAN), so many will refer to them as SPAN-ports.

> Remote SPAN (RSPAN): An extension of SPAN called remote SPAN or RSPAN. RSPAN allows you to monitor traffic from source ports distributed over multiple switches, which means that you can centralize your network capture devices. RSPAN works by mirroring the traffic from the source ports of an RSPAN session onto a VLAN that is dedicated for the RSPAN session. This VLAN is then trunked to other switches, allowing the RSPAN session traffic to be transported across multiple switches. On the switch that contains the destination port for the session, traffic from the RSPAN session VLAN is simply mirrored out the destination port.

Source: https://community.cisco.com/t5/networking-documents/understanding-span-rspan-and-erspan/ta-p/3144951

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 31

⚠ IP address research 15 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

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 32

⚠ Data types: IP reputation – 15min

Objective:
Find out what IP reputation lists are with some examples.

Purpose:
Identifying bad things can be hard.

We have a concept named, Indicators of Compromise (IoC).

> Indicators of Compromise (IOC) any piece of information that can be used to objectively describe a network intrusion, expressed in a platform-independent manner. Say, if a server connects to THIS specific IP, which is KNOWN to be the control and command server for a malware strain, we conclude the device is infected.

For this we can often download files for identification purposes with some reputation.

Suggested method:
We will start with the example of AlienVaults IP Reputation database

Note: links change, and the links books often does not work!

First visit:
https://rdrr.io/github/hrbrmstr/netintel/man/Alien.Vault.Reputation.html

Then download this, maybe need to use wget or other command line tool:
http://reputation.alienvault.com/reputation.data

It contains bad IP addresses so your Anti-Virus programs may warn you about the file!

Hints:
Passive DNS systems exist, which allow you to lookup older records, things that have moved.

Maltrail software contains a lot of lists
https://github.com/stamparm/maltrail

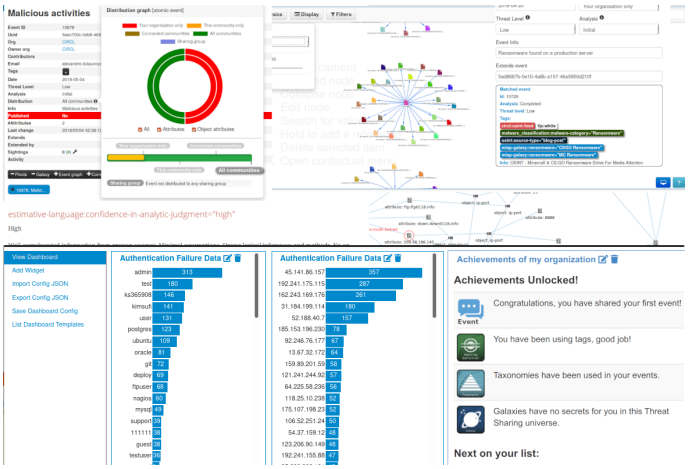
Solution:
When you have understood that data from others can help your identification efforts, you are done.

Discussion:
We also have used reputation a lot in fighting spam and scam emails.

© license CC BY 3.0. 2025 Henrik Kramselund

Exercise 33

❶ Research MISP Project 30min



Objective:
Research the MISP Project, if you like run it locally on your workstation

Evaluate if this is something you would like to have permanently or during an incident.

Purpose:
Running MISP Project is will allow you to fetch reputation lists easily and analyse logs better

Suggested method:
Go to the web site and look at installation path:
`https://www.misp-project.org/download/`

You can try to run the application, or we can see the demo on my VM. If we decide to install it, it will take longer.

It may be possible to use a VM image from `https://vm.misp-project.org/`

Credentials are:

```
For the MISP web interface -> admin@admin.test:admin
For the system -> misp:Password1234
```

Hints:
A VM image is probably fastest, and there may also be Docker images available YMMV.

Solution:
When you have seen the installation instructions and considered installing it you are done. If you can manage to get it running with the allotted time, great!

Discussion:
Downloading VM images can be fine for testing, but can be harder to run later. May not be based on the operating system your organisation prefer, can monitor etc.

© license CC BY 3.0. 2025 Henrik Kramselund