Computer Systems Security

exercises

Henrik Kramselund
hlk@zencurity.com

March 4, 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

Preface

This material is prepared for use in courses and was prepared by Henrik Kramselund, http://www.zencurity.com . It describes the networking setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github
Look for system-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

Exercise content

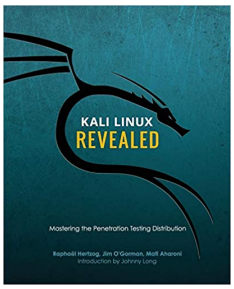
Most exercises follow the same procedure and has the following content:

- **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 10min



Kali Linux Revealed Mastering the Penetration Testing Distribution

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

This is named Kali Linux Revealed (KLR)

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

Suggested method:
Create folders for educational materials. Go to https://www.kali.org/

Read and follow the instructions for downloading the book.

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

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

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

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

Exercise 2

⚠ Download Debian Administrators Handbook (DEB) Book 10min

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 3

⚠ 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 4

⚠ Check your Debian VM 10min

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

We need a Debian Linux for running a few extra 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.

Hints:

Solution:
When you have a updated Debian Linux, then we are good.

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

Exercise 5

⚠ Investigate /etc 10min

Objective:
We will investigate the /etc directory on Linux

We need a Kali Linux and a Debian Linux VM, to compare

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.

```
hlk@debian:~$ cd /etc
hlk@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
hlk@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-9-lab:/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.

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 6

⚠ Enable UFW firewall - 10min

Source: Picture is Eilean Donan castle entrance
`2048px-Eilan_Donan_Castle_Entrance.jpg` from `https://en.wikipedia.org/wiki/Eilean_Donan`

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 could 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`

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

Exercise 7

❶ 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.

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

```
sudo 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 Git, ansible and my playbooks: (as root run, or with sudo as shown)

```
sudo apt -y install ansible git
```

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.
```

If you want to install the Docker system, you can run the Ansible playbook from the directory named docker-install.

Then run it with:

```
cd ~/kramse-labs/docker-install
ansible-playbook -v 1-dependencies
```

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 acccount 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 for more details.

Exercise 8

ℹ Use Ansible to install Elastic Stack

Objective:
See how a real deployment of a large application could happen with Ansible, installing Elasticsearch

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

Suggested method:
We will not run Elasticsearch, this is an example how to use Git and automated system configuration only!

If you wanted to run Elasticsearch you could either be 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. The latter would take less than 30min, the copy paste method could take hours.

Ansible used below is a configuration management tool `https://www.ansible.com/`

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
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

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

Solution:
~~When you have a updated VM and Ansible running, then we are good.~~ This an example exercise, not something we will run in this course!

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

⚠ Mitre ATT&CK Framework 25 min

MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

ATT&CK™

Source: Great resource for attack categorization

Objective:
See examples of attack methods used by real actors.

Purpose:
When analyzing incidents we often need to understand how they gained acccess, moved inside the network, what they did to escalate privileges and finally exfiltrate data.

Suggested method:
Go to the web site `https://attack.mitre.org/`, browse the matrix and read a bit here and there.

Browse the ATT&CK 101 Blog Post
`https://medium.com/mitre-attack/att-ck-101-17074d3bc62`

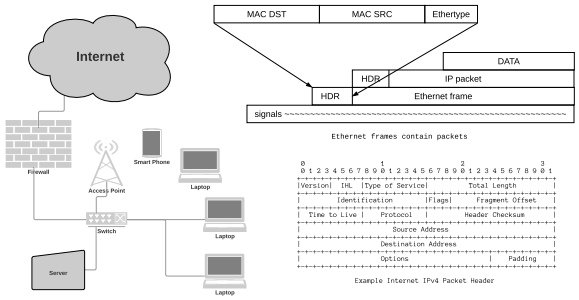
Hints:
The columns can be thought of as a progression. An attacker might perform recon first, then gain initial access etc. all the way to the right most columns.

Solution:
When you have researched a few details in the model you are done.

Discussion:
This is a large model which evolved over many years. You are not expected to remember it all, or understand it all.

Exercise 10

⚠ Create Lab network 15min



Sample IP lab network, not the one you are configuring.

Objective:
Make sure your two (or more) virtual machines can see each other.

Purpose:
We will be doing port scanning and would like the option to do exerccises using only the systems under your control, your virtual machines.

Suggested method:
We will first recommend the method listed in the MLSH book, page 13, Configuring a network for VirtualBox virtual machines.

Follow instruction for creating a connection using a USB Ethernet adapter and bridge your virtual machines.

This will give them the shortest path to the network - Ethernet and no NAT.

Hints:
Unfortunately this method mostly work with Ethernet, and NOT with Wi-Fi adapters!

If you only have a Wi-Fi you can instead create a new network using virtual box, remember to enable DHCP and then connect your VMs to this new network. Then they should be able to see each other with ping, and you can port scan etc.

If you use another technology than VirtualBox you can create similar settings, but you may have to Read The Fine Manual (RTFM).

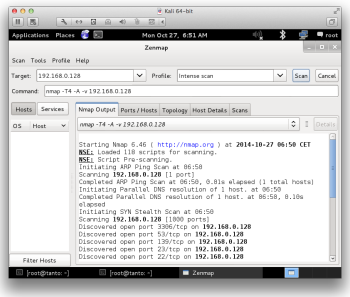
Solution:
When you have configured networks and can see the VMs you are done.

Discussion:
Why does this work better with Ethernet? Mostly it has got to do with drivers. The Ethernet drivers are more simple, with easier implementation of another virtual network card.

The Wi-Fi adapters typically need security settings etc. so adding this virtual network card for the bridging becomes more complicated.

Exercise 11

⚠ Discover active systems ping and port sweep 15min



Objective:
Use nmap to discover active systems and ports

Purpose:
Know how to use nmap to scan networks for active systems. These ports receive traffic from the internet and can be used for DDoS attacks.

Tip: Yes, filtering traffic further out removes it from processing in routers, firewalls, load balancers, etc. So making a stateless filter on the edge may be recommended.

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 ICMP and port probes: `nmap -sP 10.0.45.*`
- Port sweeps 80/tcp and 443/tcp: `nmap -p 80 10.0.45.*`
- Port sweeps UDP scans can be done: `nmap -sU -p 161 10.0.45.*`

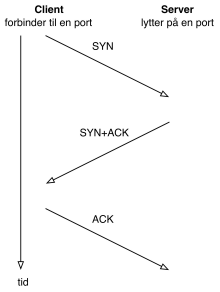
Discussion:
Quick scans quickly reveal interesting hosts, ports and services

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

We will now assume port 80/443 are open, as well as a few UDP services - maybe we can use them in amplification attacks later.

Exercise 12

⚠ Execute nmap TCP and UDP port scan 20 min



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

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

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

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

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

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

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

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

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

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

Exercise 13

⚠ Perform nmap OS detection 15min

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

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

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

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

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

Hints:

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

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

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

Exercise 14

⚠ Perform nmap service scan 15min

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

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

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

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

Solution:
Run nmap and get results.

Discussion:

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

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

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

Exercise 15

❶ Nmap full scan - 15min

Objective:
Documenting the security level of a network often requires extensive testing. Below are some examples of the scanning methodology needed.

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 multiple nmap and get results. At least TCP and UDP top-ports 10.

Discussion:
Recommendation it is highly recommended to always use:

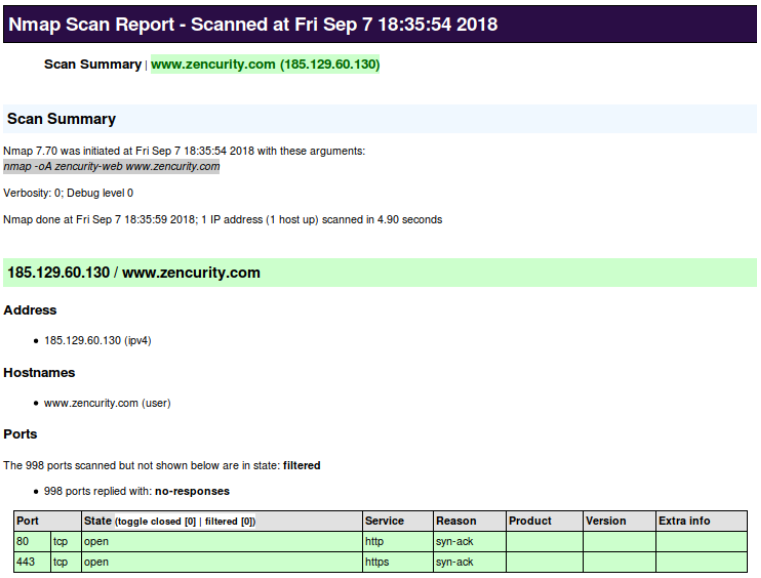
```
-iL <inputfilename>: Input from list of hosts/networks
-oA outputbasename: output in all formats, see later
```

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 10.1.2.3 192.0.2.123
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 16

❶ Reporting Nmap HTML 10min



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

Scan Summary | www.zencurity.com (185.129.60.130)

Scan Summary

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

Verbosity: 0; Debug level 0

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

185.129.60.130 / www.zencurity.com

Address

- 185.129.60.130 (ipv4)

Hostnames

- www.zencurity.com (user)

Ports

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

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

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

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

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

Suggested method:

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

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

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.
+Nmap 7.70 scan initiated Fri Sep 07 18:46:01 2018 as: nmap -oA zencurity-web-2 www.zencurit

 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 17

❶ Nping check ports 10min

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

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

Suggested method:

```
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 <ms
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 <ms
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 <ms

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:

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 18

ℹ Nmap Scripting Engine NSE scripts 20min

Objective:
Show the use of NSE scripts, copy/modify a script written in Lua.

Purpose:
Investigate the scripts from Nmap, copy one, learn how to run specific script using options

Suggested method:

```
# cd /usr/share/nmap/scripts
# nmap --script http-default-accounts.nse www.zencurity.com
# cp http-default-accounts.nse http-default-accounts2.nse
# nmap --script http-default-accounts2.nse www.zencurity.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-07 19:45 CEST
...
```

This will allow you to make changes to existing scripts.

Hints:
We will do this quick and dirty - later when doing this at home, I recommend putting your scripts in your home directory or a common file hierarchy.

Solution:
Other examples

```
nmap --script http-enum 10.0.45.0/24
nmap -p 445 --script smb-os-discovery 10.0.45.0/24
```

Discussion:
There are often new scripts when new vulnerabilities are published. It is important to learn how to incorporate them into your scanning. When heartbleed roamed I was able to scan about 20.000 IPs for Heartbleed in less than 10 minutes, which enabled us to update our network quickly for this vulnerability.

It is also possible to run categories of scripts:

```
nmap --script "http-*"

     nmap --script "default or safe"
      This is functionally equivalent to nmap --script "default,safe". It loads all scripts th

     nmap --script "default and safe"
      Loads those scripts that are in both the default and safe categories.
```

or get help for a script:

```
# nmap -script-help http-vuln-cve2013-0156.nse
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-07 19:00 CEST

http-vuln-cve2013-0156
Categories: exploit vuln
https://nmap.org/nsedoc/scripts/http-vuln-cve2013-0156.html
  Detects Ruby on Rails servers vulnerable to object injection, remote command
  executions and denial of service attacks. (CVE-2013-0156)

  All Ruby on Rails versions before 2.3.15, 3.0.x before 3.0.19, 3.1.x before
  3.1.10, and 3.2.x before 3.2.11 are vulnerable. This script sends 3 harmless
  YAML payloads to detect vulnerable installations. If the malformed object
  receives a status 500 response, the server is processing YAML objects and
  therefore is likely vulnerable.

  References:
  * https://community.rapid7.com/community/metasploit/blog/2013/01/10/exploiting-ruby-
on-rails-with-metasploit-cve-2013-0156',
  * https://groups.google.com/forum/?fromgroups=#!msg/rubyonrails-security/61bkgvnSGTQ/nehwjA8
  * http://cvedetails.com/cve/2013-0156/
```

Some scripts also require, or allow arguments into them:

```
  nmap -sC --script-args 'user=foo,pass=",=bar",paths=/admin,/cgi-bin,xmpp-info.server_name=lo
```

Exercise 19

⚠ Configure a Database - 20 min

Objective:
Try creating a database and add a few users. We will use MariaDB as an example. You can read more about MariaDB at https://mariadb.org/

Purpose:
Show that creating a database is very easy, and adding a new user cost literally nothing.

So why aren't more developers concerned with security, least privilege, creating read-only users etc.

Suggested method:
Get an overview of the LibreNMS install instructions located at:
https://docs.librenms.org/Installation/Install-LibreNMS/

We are running Debian, and since we only need to play with the database, only install that part:

```
apt install mariadb-client mariadb-server
```

and follow only the instructions in Configure MariaDB – the ones with Debian!

Repeated here for completeness, but `vi` changed into edit.

Configure MariaDB – Debian

Edit `/etc/mysql/mariadb.conf.d/50-server.cnf` using vi, nano or another preferred editor

Within the [mysqld] section add:

```
innodb_file_per_table=1
lower_case_table_names=0
```

Start the database

```
systemctl enable mariadb
systemctl restart mariadb
```

As root run the database client program `mysql`

```
mysql -u root
```

NOTE: Change the 'password' below to something secure.

Add a user for the system:

```
CREATE DATABASE librenms CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'librenms'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
FLUSH PRIVILEGES;
exit
```

Add another user, with read only:

```
CREATE USER 'my_readonly'@'localhost' IDENTIFIED BY 'secret_password';
GRANT SELECT ON librenms.* TO 'my_readonly'@'localhost';
FLUSH PRIVILEGES;
```

Such a user would be able to see into the database, fetch statistics etc.

Solution:
When you have a running database with two users, you are done.

Discussion:
Lots of systems need access to data, but if noone ask – what permissions, we often default to read AND write. We should default to read-only.

Exercise 20

⚠ RBAC Access permissions on GitHub 30-45min

Objective:
See actual real life example of permissions.

Note: This exercise requires a GitHub account, so make sure your group has one – it is recommended to do this with others in a group.

Purpose:
GitHub is a very popular code sharing site.

Suggested method:
Go to GitHub web page:
https://help.github.com/en/articles/access-permissions-on-github

Follow links to other pages, like:
https://help.github.com/en/articles/permission-levels-for-an-organization

Hints:
Some might already have an account on GitHub - maybe work through adding a repository and adding collaborators.

If you have an organisation, even better.

Solution:
When you have discussed GitHub permissions and played with a repository you are done.

Discussion:
The internet is decentralized, but recent years see more centralization - GitHub, DNS Google DNS, Cloudflare.

What are some problems in this?

Exercise 21

⚠ Password Cracking 15min

Objective:
Crack your own passwords using John the Ripper

Purpose:
See how fast hashes from bad algorithms can be cracked, and how new ones are slow to crack.

Suggested method:
John the Ripper is available from the web page, but also as a package:
`https://www.openwall.com/john/`

You should install from the package system using apt install, do apt search first to find the package name.

1. Install John, if not already there: `apt install whois john`
2. Copy the local password database, as root: `cp /etc/shadow /root/mypasswords`
3. Start cracking: `john --single /root/mypasswords`
4. Restart with incremental mode, bruteforce: `john --incremental /root/mypasswords`

You can make it easier if you add a few users with bad passwords first.

NOTE: newer Debian 12 systems use a hashing method which is NOT supported by John. You can instead create a bad hash using the tool `mkpasswd -5`

```
root@Projects:~# mkpasswd -5
Password:      // I entered henrik42 as password
$1$ENlQhB5/$trB73FyYwHl/53Ys9LrQf0
```

Then copy this to the mypasswords file using nano or vi editors.

```
root@Projects:~# cat mypasswords
root:!*:19562:0:99999:7:::
user:$1$4f1IYkFK$OQzP.MogNjOubDvtFO6GS1:19562:0:99999:7:::
tinyproxy:!:19562::::::
root@Projects:~# john --single mypasswords
Created directory: /root/.john
Loaded 1 password hash (md5crypt [MD5 32/64 X2])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 100% 0g/s 18387p/s 18387c/s 18387C/s user999991901..999991900
Session completed
```

Hints:
You can download other sample hashes from
`https://hashcat.net/wiki/doku.php?id=example_hashes`

Solution:
When you have cracked at least one password then you are done.

Discussion:
A better tool might be hashcat, found at:
http://hashcat.net/wiki/

This tool can be used with GPUs Graphical Processing Units / graphic cards for more speed.

Still I find John is often sufficient to crack bad passwords, and also for verification purposes it works great.

Exercise 22

❶ Configure SSH keys for more secure access 30min

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:

1. First generate a SSH key, using `ssh-keygen`
2. Copy the public key onto a system, using `ssh-copy-id`
3. Test using the ssh command

Generate key:

```
$ 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:l5esp66lQArFOlXqOoHnxpg8zRS6shK8nx9KGf+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` :

```
$ 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/3FYPY84cpTKmEW7XoQ4zDNf/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
hlk@10.0.42.147's password:

Number of key(s) added:        1

Now try logging into the machine, with:   "ssh -o 'IdentitiesOnly yes' 'hlk@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.

You can add keys to memory, allowing you to enter a passphrase at the beginning of the session, but no passphrases or passwords after:

```
$ ssh-add .ssh/kramse
// Passphrase here
Identity added: .ssh/kramse (.ssh/kramse)
$ ssh-add -l
4096 SHA256:YsR+HhK7u7045ZL8ZDZnlgEnrv+RQG4eJI5oBJAEacs .ssh/kramse (RSA)
```

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.

You should remove the possibility for root logins and logins using password, when keys work!

This is done editing the file `/etc/ssh/sshd_config` and the options:


```
#PermitRootLogin prohibit-password
PermitRootLogin no
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
```


Exercise 23

⚠ CIS Benchmarks teaser 30min

Objective:
Checkout CIS benchmarks

Purpose:
We have talked about operating system security, starting with user accounts on Linux. There are much more to security than users, and there are multiple tools available. One such framework is the Center for Internet Security (CIS) Benchmarks.

Their benchmarks are also implemented in multiple tools, which can help automate checking of the settings.

We will now take a quick look at the CIS benchmarks, and more benchmarking will be discussed later.

Suggested method:
Choose your operating system/solution and check out at least the general CIS benchmark description and a specific one.

Main site `https://www.cisecurity.org/` and some example benchmarks, suggestions:

- Microsoft Windows Server `https://www.cisecurity.org/benchmark/microsoft_windows_server`
- Microsoft Azure https://www.cisecurity.org/benchmark/azure
- Azure Linux `https://www.cisecurity.org/benchmark/azure_linux`
- Debian Linux `https://www.cisecurity.org/benchmark/debian_linux`
- Kubernetes `https://www.cisecurity.org/benchmark/kubernetes`

Hints:
Center for Internet Security (CIS) Benchmarks are also available for Windows, so checkout their offerings:

`https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark`

Solution:
When you have seen the CIS web site, read about a specific benchmark you are done.

Discussion:
We will discuss other tools in class.

Exercise 24

❶ Example Policies up to 45min

Objective:
See real world high level policies and discuss where to start

Purpose:
When writing your first policy it may be hard to know what to include. Starting from an example is often easier.

Suggested method:
Find your AUP for the ISPs we use, you use, your company uses.

Hints:
Policies for different environments are often very different in scope and goals.

SANS, for example, publishes a list of template policies that you can edit for your own needs. At the time of writing, its list of topics are:

- Acceptable Encryption Policy
- Acceptable Use Policy
- Clean Desk Policy
- Disaster Recovery Plan Policy
- Digital Signature Acceptance Policy
- Email Policy
- Ethics Policy
- Pandemic Response Planning Policy
- Password Construction Guidelines
- Password Protection Policy
- Security Response Plan Policy
- End User Encryption Key Protection Policy
- Acquisition Assessment Policy
- Bluetooth Baseline Requirements Policy
- Remote Access Policy
- Remote Access Tools Policy
- Router and Switch Security Policy
- Wireless Communication Policy
- Wireless Communication Standard
- Database Credentials Policy
- Technology Equipment Disposal Policy
- Information Logging Standard
- Lab Security Policy
- Server Security Policy
- Software Installation Policy
- Workstation Security (For HIPAA) Policy
- Web Application Security Policy

Source: `https://www.sans.org/information-security-policy/`

Example, how do you handle BYOD Bring your own devices, educational institutions you expect students to bring them, in a secure enterprise only company devices may be allowed.

Solution:
When you have seen at least two different policies you are done.

Discussion:
How do you both write AND create awareness about a policy?

Exercise 25

⚠ Example Password policies on Linux up to 30min

Objective:
See an example of how to implement password policy on Linux **Purpose:**

Suggested method:
Follow the book example starting at page 63, Installing and configuring pwquality

You don't have to install this software, but read the example.

Hints:
This is not the only way, and other operating systems might have this built-in.

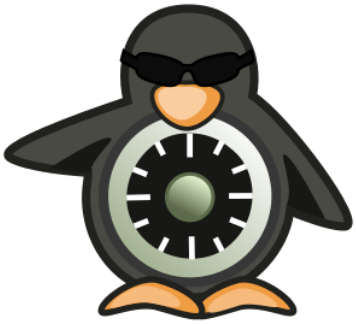
Solution:
When you have seen this example, you are done. I recommend reading the next examples to about expiration and brute-force too.

Discussion:
Are there cases where password and account expiration still makes sense?

Sure, consultants hired for a shorter period of time should definitely have accounts that expire.

Exercise 26

ℹ SELinux Introduction up to 60min



Objective:
Check out the SELinux system
https://www.debian.org/doc/manuals/debian-handbook/sect.selinux.en.html

and the setup instructions at:
https://wiki.debian.org/SELinux/Setup

This is a very complex system, and for that reasone seldom used in production. Not working right now - Create a secret file, that you can read, but root cant.

It is recommended to create a snapshot of your VM before messing with SELinux!

Purpose:
Everybody reads about Discretionary Access Control (DAC) and Mandatory Access Control (MAC) but few realize that Linux implements it. DAC is the most common case, where you decide the security settings on files – permissions with the Read, Write, Execute bit. MAC can be implemented using SELinux – for certain cases.

Suggested method:
Try enabling and disabling the policies in your Debian VM.

First install prerequisites - approx 75MB download on my system:
`apt-get install selinux-basics selinux-policy-default auditd`

Then run activation of SELinux:
`selinux-activate`

```
root@debian-lab:~# selinux-activate
Activating SE Linux
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.9.0-9-amd64
Found initrd image: /boot/initrd.img-4.9.0-9-amd64
Found linux image: /boot/vmlinuz-4.9.0-8-amd64
Found initrd image: /boot/initrd.img-4.9.0-8-amd64
done
SE Linux is activated.  You may need to reboot now.
root@debian-9-lab:~#
```

Perform the reboot, `shutdown -r now` then check again.

Not enabled will show this, try again:

```
root@debian-lab:~# sestatus
SELinux status:                 disabled
```

Enabled, but not the current mode and mode from config file discrepancy:

```
root@debian:~# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   enforcing
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      30
```

While playing I had changed the mode temporarily to enforcing! Next reboot would make SELinux run in the more permissive mode

26.0.1 Part 2 - do this when SELinux is enabled

Create a directory and a test file:

```
root@debian:~# setenforce 0   // set mode permissive!
root@debian:~# cd
root@debian:~# mkdir /etc/private
root@debian:~# echo "hey" > /etc/private/README
root@debian:~# cat /etc/private/README
hey
root@debian:~#
```

Root can read the file, yay!

Copy example files:

```
cp -r /usr/share/doc/selinux-policy-dev/examples .
cd examples/
```


Create a file myprivate.te with this content:

```
policy_module(myprivate, 1.0)

########################################
#
# Declarations
#
type etc_private_t;
fs_associate(etc_private_t)

type sysadm_t;
type sysadm_exec_t;

userdom_admin_user_template(sysadm_t)

allow sysadm_t etc_private_t:{dir file} relabelto;
```

Note last line is missing a sysadm domain, does not work.

Then compile using this: `make myprivate.pp`

```
root@debian:~/examples# make myprivate.pp
Compiling default myprivate module
/usr/bin/checkmodule:  loading policy configuration from tmp/myprivate.tmp
/usr/bin/checkmodule:  policy configuration loaded
/usr/bin/checkmodule:  writing binary representation (version 17) to tmp/myprivate.mod
Creating default myprivate.pp policy package
rm tmp/myprivate.mod.fc tmp/myprivate.mod
root@debian:~/examples#
```

then it should have been possible to enable/disable enforcing mode, and see the file becoming unreadable - even by root.

Something is wrong, when enabling enforcing mode, the chcon command fails:

```
root@debian:~/examples# setenforce 1
root@debian:~/examples# chcon -R -t etc_private_t /etc/private/README
chcon: failed to change context of '/etc/private/README' to system_u:object_r:etc_private_t:s0
root@debian:~/examples# chcon -R -t etc_private_t /etc/private
chcon: failed to change context of 'README' to system_u:object_r:etc_private_t:s0: Invalid arg
chcon: failed to change context of '/etc/private' to system_u:object_r:etc_private_t:s0: Inval

root@debian:~/examples# setenforce 0
root@debian:~/examples# chcon -R -t etc_private_t /etc/private/README
root@debian:~/examples#
// When Linux returns to the command prompt without messages no errors were observed
```

So SELinux IS preventing us from doing it :-D

this example is in parts based on this blog post:
http://blog.siphos.be/2015/07/restricting-even-root-access-to-a-folder/

Hints:
Keeping SELinux enabled may NOT be a good idea, since some tools may not work correctly, until policies are downloaded, written or installed.

Temporarily disable SELinux:
`echo 0 > /sys/fs/selinux/enforce`

Temporarily enable SELinux:
`echo 1 > /sys/fs/selinux/enforce`

or use the command `setenforce 0` or `setenforce 1`

The main config for setting permissive or enforcing mode is `/etc/selinux/config`:

```
root@debian-lab:~# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# default - equivalent to the old strict and targeted policies
# mls     - Multi-Level Security (for military and educational use)
# src     - Custom policy built from source
SELINUXTYPE=default

# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Solution:
When you have enabled and seen the commands used, you are done.

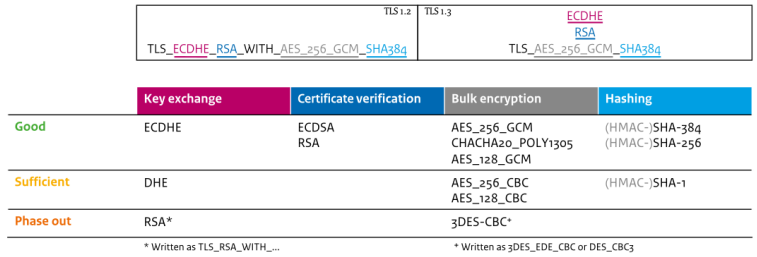
It is easy to have multiple hours disappear when working with SELinux.

Discussion:
Yes, the root user can disable the SELinux protection :-D

I had Firefox crash at least once during this exercise, so beware - fancy and bigger applications may crash when using this!

Exercise 27

⚠ SSL/TLS scanners 15min



TLS 1.2	TLS 1.3
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE RSA TLS_AES_256_GCM_SHA384

	Key exchange	Certificate verification	Bulk encryption	Hashing
Good	ECDHE	ECDSA RSA	AES_256_GCM CHACHA20_POLY1305 AES_128_GCM	(HMAC-)SHA-384 (HMAC-)SHA-256
Sufficient	DHE		AES_256_CBC AES_128_CBC	(HMAC-)SHA-1
Phase out	RSA*		3DES-CBC*	
	* Written as TLS_RSA_WITH_...		* Written as 3DES_EDE_CBC or DES_CBC3	

Figure 2 – Cipher suite notation in TLS 1.2 and TLS 1.3. The table summarizes algorithm selections and their security level.
Not included in the (old) cipher suite notation are: versions; hash functions for certificate verification; hash functions for key exchange; key sizes & choice of groups; and options. These can be found in their respective sections. For ordering, refer to the section Prefer faster and safer algorithms.

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 www.kramse.org
Version: 1.10.5-static
OpenSSL 1.0.2e-dev xx XXX xxxx

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

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

Also run it against SMTPTLS if possible. Choose a mail server, `mail.kramse.org` can work like this:

```
sslscan   --starttls-smtp mail.kramse.org:25
```

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

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

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

Exercise 28

❶ Nmap Ikescan IPsec 15min

unfinished, will be updated later

Objective:
Try Nmap and Ikescan

Purpose:
Check settings on Internet Key Exchange protocol, which is a part of IPsec IP security framework - which is used for Virtual Private Network (VPN) tunnels.

Suggested method:
Ike-scan is available in the Kali package system, so install using apt.

It seems the code is now on Github:
`https://github.com/royhills/ike-scan`

Where you can read more about running the tool.

Hints:
This tools sends a lot of proposals to a firewall/VPN gateway and recognizes the responses.

You should look for 3DES, DES and older versions of MAC algorithms like MD5 and SHA1.

Note: you can also try the ike-version script in Nmap, which can give a little extra information:

```
-- @usage
-- nmap -sU -sV -p 500 <target>
-- nmap -sU -p 500 --script ike-version <target>
--
-- @output
-- PORT    STATE SERVICE REASON       VERSION
-- 500/udp open  isakmp  udp-response Fortinet FortiGate v5
-- | ike-version:
-- |   vendor_id: Fortinet FortiGate v5
-- |   attributes:
-- |     Dead Peer Detection v1.0
-- |_    XAUTH
-- Service Info: OS: Fortigate v5; Device: Network Security Appliance; CPE: cpe:/h:fortinet:fo
```

Note: port 500/udp and 4500/udp are the common ones used for IKE.

Solution:
When you have tried the tool against at least one VPN gateway you are done. Perhaps try it against your company VPN, this is NOT an attack - more like a probe sent.

Discussion:
You should review and update settings for encryption at least once a year, or when news of another attack on algorithms are found.

The current recommendation for VPN connections with IKE are listed below.

Use the following guidelines when configuring Internet Key Exchange (IKE) in VPN technologies:
* Avoid IKE Groups 1, 2, and 5.
* Use IKE Group 15 or 16 and employ 3072-bit and 4096-bit DH, respectively.
* When possible, use IKE Group 19 or 20. They are the 256-bit and 384-bit ECDH groups, respectively.
* Use AES for encryption.

Paper:
https://www.cisco.com/c/en/us/about/security-center/next-generation-cryptography.html

Exercise 29

ℹ SSH scanners 15min

Objective:
Try ssh scanners, similar to sslscan 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

⚠ 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?

Exercise 31

⚠ Perform privilege escalation using files 30min

The brave new world of IPv6

Objective:
Perform a simple privilege escalation attack

Purpose:
Try and test a back door script.

Suggested method:

1. Create a shell copy with SUID bit set as privileged user
2. Run the command as non-privileged user to see if it works

Hints:
In this exercise first try out the malicious commands for creating a back door shell program. Login in as root, then:

```
root@debian:~# rm /tmp/.xxsh
root@debian:~# apt install zsh
...
root@debian:~# cp /bin/zsh /tmp/.xxsh
root@debian:~# chmod +sw /tmp/.xxsh
```

Then test using a normal user, another window:

```
hlk@debian:~$ /tmp/.xxsh
# id
```



```
uid=1000(hlk) gid=1000(hlk) euid=0(root) egid=0(root) groups=0(root),24(cdrom),25(floppy),
29(audio),30(dip),44(video),46(plugdev),108(netdev),112(lpadmin),117(scanner),1000(hlk)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
#
```

The effective user id should be 0 which is root. It might not work as intended, due to enhanced security in the shell programs! Namely it wont work with Bourne Again Shell (bash) but maybe with dash. If dash is not installed, try installing it.

When this manual process work. Then an attacker would automate it, make it into a small script. Imagine if the root user was running automated scripts, and you could add yours to a directory used in the PATH for these automated ones.

This happens in a lot of devices and hosts today.

The main takeaway is that root scripts should ALWAYS have a PATH defined, and ALL directories used by root script should only be writable by root!

Solution:
When you have created the shell copy you are done. Both seeing it work and not work has value. If it does not work with bash, it is a good thing!

Further advanced steps would be to add this into some PATH writable by you, and letting a cron job escalate. Then do a cron job that uses this command - a cron job running every 5 minutes using the `ls` command and introduce your malicious script by putting it before the real command in the PATH.

Discussion:
Why is the file named with a dot as the first character?

Does the /tmp folder need to be a place to run scripts? No, but many applications unfortunately require exactly this.

What is defense in depth? Does it apply here?

Finish off the exercise by running, and looking at the output from:
`find / -perm -4000 -o -perm -6000`

Exercise 32

ℹ Anti-virus and "endpoint security" 30min

Objective:
Discuss when to use Anti-virus and "endpoint security"

Purpose:
Anti-virus programs have been shown to catch some viruses, useful.

Anti-virus programs have been shown to be insecure programs that also slows down systems, counter-productive and increases target surface and exposure.

Suggested method:
Sit in groups 3-5 – discuss among yourselves. Write down plus and minus for using anti-virus – especially which use-cases should use AV, and which shouldn't.

Hints:
In some cases people have installed AV products for check-mark security, the check-list said to have AV, so we installed a mail scanner on this web server – bad security.

Where do we install anti-malware defenses?

- Mailscanners
- Proxy systems – web proxies, incoming, outgoing, problems with TLS?
- On firewalls – what is a firewall, can we do inspection, really?!
- File systems – network shares servers, storage systems
- What about virtualized systems
- ...

Solution:
When we have done a collected talk and discussion we are done.

Discussion:
I dont use anti-virus products at all. I do use a lot of backup though.

Which is more trust-worthy - a restored system or a system cleaned by random anti-virus program?

Exercise 33

ℹ Buffer Overflow 101 - 30-40min

Objective:
Run a demo program with invalid input - too long.

Purpose:
See how easy it is to cause an exception.

Suggested method:

- Small demo program `demo.c`
- Has built-in shell code, function `the_shell`
- Compile: `gcc -o demo demo.c`
- Run program `./demo test`
- Goal: Break and insert return address

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char **argv)
{       char buf[10];
         strcpy(buf, argv[1]);
         printf("%s\n",buf);
}
int the_shell()
{  system("/bin/dash");  }
```

NOTE: this demo is using the dash shell, not bash - since bash drops privileges and won't work.

Use GDB to repeat the demo by the instructor.

Hints:
First make sure it compiles:

```
$ gcc -o demo demo.c
$ ./demo hejsa
hejsa
```

Make sure you have tools installed:

```
apt-get install gdb
```

Then run with debugger:

```
$ gdb demo
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
```

```
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from demo...(no debugging symbols found)...done.
(gdb)
(gdb) run `perl -e "print 'A'x22; print 'B'; print 'C'"`
Starting program: /home/user/demo/demo `perl -e "print 'A'x22; print 'B'; print 'C'"`
AAAAAAAAAAAAAAAAAAAAAABC

Program received signal SIGSEGV, Segmentation fault.
0x0000434241414141 in ?? ()
(gdb)
// OR
(gdb)
(gdb) run $(perl -e "print 'A'x22; print 'B'; print 'C'")
Starting program: /home/user/demo/demo `perl -e "print 'A'x22; print 'B'; print 'C'"`
AAAAAAAAAAAAAAAAAAAAAABC

Program received signal SIGSEGV, Segmentation fault.
0x0000434241414141 in ?? ()
(gdb)
```

Note how we can see the program trying to jump to address with our data. Next step would be to make sure the correct values end up on the stack.

Solution:
When you can run the program with debugger as shown, you are done.

Discussion:

the layout of the program - and the address of the `the_shell` function can be seen using the command `nm`:

```
$ nm demo
0000000000201040 B __bss_start
0000000000201040 b completed.6972
                 w __cxa_finalize@@GLIBC_2.2.5
0000000000201030 D __data_start
0000000000201030 W data_start
0000000000000640 t deregister_tm_clones
00000000000006d0 t __do_global_dtors_aux
0000000000200de0 t __do_global_dtors_aux_fini_array_entry
0000000000201038 D __dso_handle
0000000000200df0 d _DYNAMIC
```



```
0000000000201040 D _edata
0000000000201048 B _end
0000000000000804 T _fini
0000000000000710 t frame_dummy
0000000000200dd8 t __frame_dummy_init_array_entry
0000000000000988 r __FRAME_END__
0000000000201000 d _GLOBAL_OFFSET_TABLE_
                 w __gmon_start__
000000000000081c r __GNU_EH_FRAME_HDR
00000000000005a0 T _init
0000000000200de0 t __init_array_end
0000000000200dd8 t __init_array_start
0000000000000810 R _IO_stdin_used
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
0000000000200de8 d __JCR_END__
0000000000200de8 d __JCR_LIST__
                 w _Jv_RegisterClasses
0000000000000800 T __libc_csu_fini
0000000000000790 T __libc_csu_init
                 U __libc_start_main@@GLIBC_2.2.5
0000000000000740 T main
                 U puts@@GLIBC_2.2.5
0000000000000680 t register_tm_clones
0000000000000610 T _start
                 U strcpy@@GLIBC_2.2.5
                 U system@@GLIBC_2.2.5
000000000000077c T the_shell
0000000000201040 D __TMC_END__
```

The bad news is that this function is at an address `000000000000077c` which is hard to input using our buffer overflow, please try ☺We cannot write zeroes, since strcpy stop when reaching a null byte.

We can compile our program as 32-bit using this, and disable things like ASLR, stack protection also:

```
sudo apt-get install gcc-multilib
sudo bash -c 'echo 0 > /proc/sys/kernel/randomize_va_space'
gcc -m32 -o demo demo.c -fno-stack-protector -z execstack -no-pie
```

Then you can produce 32-bit executables:

```
// Before:
user@debian-9-lab:~/demo$ file demo
demo: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-
linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=82d83384370554f0e3bf4ce5030f6e3a7a5ab5ba, not stripped
// After - 32-bit
user@debian-9-lab:~/demo$ gcc -m32 -o demo demo.c
user@debian-9-lab:~/demo$ file demo
demo: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-
linux.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=5fe7ef8d6fd820593bbf37f0eff14c30c0cbf174, not stripped
```

And layout:

```
0804a024 B __bss_start
0804a024 b completed.6587
0804a01c D __data_start
0804a01c W data_start
...
080484c0 T the_shell
0804a024 D __TMC_END__
080484eb T __x86.get_pc_thunk.ax
080483a0 T __x86.get_pc_thunk.bx
```

Successful execution would look like this - from a Raspberry Pi:

```
$ gcc -o demo demo.c
$ nm demo | grep the_shell
000104ec T the_shell
$

...
(gdb) run `perl -e " print 'A'x16; print chr(0xec).chr(04).chr(0x01);" `
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/pi/demo/demo `perl -e " print 'A'x16; print chr(0xec) . chr(04)  . chr (0x01);" `
AAAAAAAAAAAAAAAA
$
```

Started a new shell.

you can now run the "exploit" - which is the shell function AND the misdirection of the instruction flow by overflow:

```
pi@raspberrypi:~/demo $ gcc -o demo demo.c
pi@raspberrypi:~/demo $ sudo chown root.root demo
pi@raspberrypi:~/demo $ sudo chmod +s demo
pi@raspberrypi:~/demo $ id
uid=1000(pi) gid=1000(pi) grupper=1000(pi),4(adm),20(dialout),24(cdrom),27(sudo),29(audio),44(video),46(plugdev),60
pi@raspberrypi:~/demo $ ./demo `perl -e " print 'A'x16; print chr(0xec).chr(04).chr(0x01);" `
AAAAAAAAAAAAAAAA
# id
uid=1000(pi) gid=1000(pi) euid=0(root) egid=0(root) grupper=0(root),4(adm),20(dialout),24(cdrom),27(sudo),29(audio)
#
```

Exercise 34

❶ Small programs with data types 15min

Objective:
Try out small programs similar to:

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv)
{       (void) argc;  (void) argv;
                short int i1 = 32767;
                printf("First debug int is %d\n", i1);
                i1++;
                printf("Second debug int is now %d \n", i1);
}
```

```
user@Projects:programs$ gcc -o int1 int1.c && ./int1
First debug int is 32767
Second debug int is now -32768
```

Purpose:
See actual overflows when going above the maximum for the selected types.

Suggested method:
Compile program as is. Run it. See the problem. Then try changing the int type, try without short, and with signed and unsigned. Note differences

Hints:
Use a calculator to find the maximum, like 2^{16}, 2^{32} etc.

Solution:
When you have tried adding one to a value and seeing it going negative, you are done.

Discussion:
Computers are not always correct when doing calculations. Above was shown with integers, and it is even worse for floating point.

> The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point arithmetic established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). The standard addressed many problems found in the diverse floating-point implementations that made them difficult to use reliably and portably. Many hardware floating-point units use the IEEE 754 standard.

Source: `https://en.wikipedia.org/wiki/IEEE_754`

Exercise 35

⚠ Real Vulnerabilities up to 30min

Objective:
Look at real vulnerabilities. Choose a few real vulnerabilities, prioritize them.

Purpose:
See that the error types described in the books - are still causing problems.

Suggested method:
We will use the 2019 Exim errors as starting examples. Download the descriptions from:

- Exim RCE CVE-2019-10149 June
  `https://www.qualys.com/2019/06/05/cve-2019-10149/return-wizard-rce-exim.txt`
- Exim RCE CVE-2019-15846 September
  `https://exim.org/static/doc/security/CVE-2019-15846.txt`

When done with these think about your own dependencies. What software do you depend on? How many vulnerabilities and CVEs are for that? Each year has huge new vulnerabilities, like the 2020 and 2021 shown above.

- CVE-2020 Netlogon Elevation of Privilege
  https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-1472
- Log4J RCE (CVE-2021-44228) - and follow up like CVE-2021-45046, also look at scanners like:
  https://github.com/fullhunt/log4j-scan

What is CVSS – Common Vulnerability Scoring System?
`https://nvd.nist.gov/vuln-metrics/cvss`

I depend on the OpenBSD operating system, and it has flaws too:
`https://www.openbsd.org/errata65.html`

You may depend on OpenSSH from the OpenBSD project, which has had a few problems too:
`https://www.openssh.com/security.html`

Hints:
Remote Code Execution can be caused by various things, but most often some kind of input validation failure.

Solution:
When you have identified the specific error type, is it buffer overflows? Then you are done.

Discussion:
How do you feel about running internet services. Lets discuss how we can handle running insecure code. What other methods can we use to restrict problems caused by similar vulnerabilities. A new product will often use a generic small computer and framework with security problems.

Exercise 36

⚠ Email Security – up to 45min

Objective:
Talk and plan roll-out of security mechanisms based on DNS records. Domain Name System. Check you personal domain and domain used for work. If you are new to this I suggest experimenting with your own personal domain, or create one.

Purpose:
Make sure everyone attending know about methods to restrict sending of false emails, how to secure this using DNSSEC, SPF, DMARC - DNS based updates to your email domain security

Email security - Goals

- SPF Sender Policy Framework
  `https://en.wikipedia.org/wiki/Sender_Policy_Framework`
- DKIM DomainKeys Identified Mail
  `https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail`
- DMARC Domain-based Message Authentication, Reporting and Conformance
  `https://en.wikipedia.org/wiki/DMARC`
- DANE DNS-based Authentication of Named Entities
  `https://en.wikipedia.org/wiki/DNS-based_Authentication_of_Named_Entities`
- Brug allesammen, check efter ændringer!

DNS-based Authentication of Named Entities (dane)

> "TLSA records store hashes of remote server TLS/SSL certificates. The authenticity of a TLS/SSL certificate for a domain name is verified by DANE protocol (RFC 6698). DNSSEC and TLSA validation results are displayer by using several icons."

Suggested method:
Use services on the internet, such as https://internet.nl/ and https://dmarcian.com/ to see current status for your domains.

Hints:
I suggest the following strategy when you implement these methods, if you dare do it right now. If you make a plan.

Basic mail security

1. Implement DNSSEC - turn it on, most likely easy
2. Configure Sender Policy Framework, perhaps only `~all` tilde means soft fail
3. Configure DomainKeys Identified Mail
4. Configure receiving email address for DMARC

5. Configure Domain-based Message Authentication - reject none

Spend some time trying different tools for DMARC reporting. A month or a week, depending on the domain and your users. Github alone has 100s of projects concerned with parsing, reporting and working with DMARC.

Then after some time has passed, and you have reviewed reporting from DMARC, turn it on for real:

1. Configure SPF to disallow with hard fail use `-all` minus
2. Configure DMARC with reject - reject emails not following policy

Advanced mail security

1. Create real certificates for DANE
2. Publish them ☺

Helpful hints from https://blog.apnic.net/2017/01/06/lets-encrypt-dane/

Solution:
When the internet is ridden of falsified spam you are done ☺ - its up to you.

Take domain(s) of your choice and make a table:

Domain ✉	DNS NS 2+	DNSSEC	SPF	DKIM	DMARC	DANE
zencurity.com	✔	✔	✔		✔	

Discussion:
You need to research before making changes to important domains. If you have domains that never send email then add the following SPF and DMARC to avoid misuse.

My own DNS template for parked domains:

```
gdns.template v=spf1 -all 43200
_dmarc.gdns.template v=DMARC1; p=reject; 43200
```

Exercise 37

⚠ Research Virtual Machine Escapes 20min

Objective:
Research how exploits can escape from Guest Virtual Machine to Host operating system. Multiple examples exist for both client virtualisation and datacenter virtualisation.

Purpose:
Research VM escapes - understand that isolation and separation does not always work. Think about how to design systems with this in mind. Perhaps virtualisation should be built using two clusters, one for external services and one for internal?

Suggested method:
Find list of CVE or do internet search. Perform searches using the virtualisation technology used in your networks. Note: even though Virtual box is used as example below other technologies like Microsoft HyperV, VMware, Xen etc. have similar problems!

examples:

- https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=virtualbox list multiple vulnerabilities.
- `https://github.com/MorteNoir1/virtualbox_e1000_0day` VirtualBox E1000 Guest-to-Host Escape The E1000 has a vulnerability allowing an attacker with root/administrator privileges in a guest to escape to a host ring3. Then the attacker can use existing techniques to escalate privileges to ring 0 via /dev/vboxdrv.
- `https://en.wikipedia.org/wiki/Virtual_machine_escape`

Hints:
Providing virtualisation is today done using hardware features in the CPU of the system. Along with the hardware features are drivers and features provided by the virtualisation system, which has errors.

Having drivers and kernel modules with errors can sometimes result in flaws exploitable by guest virtual machines.

Solution:
There is often no solution other than to patch systems, when new vulnerabilities are found - update your virtualisation NOW if you are missing updates.

Never open virtual machines from untrusted sources on your laptop with confidential data. Don't trust that the security provided is enough for researching live malware on virtual systems.

Discussion:
Is it possible to create multiple virtualisation cluster? - yes, some organisations already have multiple clusters for various reasons. Some might have development, staging and production as different clusters.

Also be aware that a lot of malware has checks trying to find out if it is running in a virtual machine, or isolated in a lab.

Exercise 38

❶ Try running a Docker container 20min

Objective:
Research how Docker containers work.

Purpose:
Docker containers are used all around the world, often together with private cloud systems. They run in the host OS kernel, and thus requires fewer resources, and provides less isolation.

> The underlying technology
> Docker is written in the Go programming language and takes advantage of several features of the Linux kernel to deliver its functionality. Docker uses a technology called namespaces to provide the isolated workspace called the container. When you run a container, Docker creates a set of namespaces for that container.
>
> These namespaces provide a layer of isolation. Each aspect of a container runs in a separate namespace and its access is limited to that namespace.

Source: `https://docs.docker.com/get-started/overview/`

Suggested method:
Find the main web page of Docker, `https://www.docker.com`

Look at the architecture of Docker, they currently have an article:
`https://www.docker.com/resources/what-container`

What is a Container?
A standardized unit of software

Browse the architecture for a bit, and then run docker on your Debian.

The instruction are on the page:
`https://docs.docker.com/engine/install/debian/`

Note: I have added a directory in my kramse-labs with playbooks for installing on Debian – use them!

```
user@Projects:~$ cd projects/github/kramse-labs/
user@Projects:kramse-labs$ ls
core-net-lab/  lab-network/ work-station/  README.md
docker-install/  suricatazeek/ LICENSE
user@Projects:kramse-labs$ cd docker-install/
user@Projects:docker-install$ pwd
/home/user/projects/github/kramse-labs/docker-install
user@Projects:docker-install$ ls
1-dependencies.yml  README.md
user@Projects:docker-install$ ansible-playbook 1-dependencies.yml
... about 10 minutes at most
```

Hints:
I recommend using the repositories, as future updates will become available there. This make future apt update/upgrade more simple.

Solution:
When you understand the basic architecture of Docker containers, you are done.

or

When you can run a small docker container, you are done.

```
docker run hello-world
```

Discussion:
Many software packages can be used via Docker containers. This allows the programmer to prepare a small image, and the user to run this directly.

Are there any security issues, many, so be careful.

See for example:
`https://docs.docker.com/engine/security/`

Exercise 39

⚠ Research Cisco ACI security assessment 45min

Objective:
Research how the product from Cisco named Application Centric Infrastructure was assessed by german security company ERNW.

Purpose:
Research ACI vulnerabilities – understand what when isolation and separation is not implemented. Think about how to design systems with this in mind. Perhaps security infrastructure should be built using modern methods. Say not using the root user for EVERYTHING, like we found out in the 1990s.

Suggested method:
Download the white paper 68 from ERNW:
`https://ernw.de/en/whitepapers/issue-68.html`

- Browse table of contents
- Look into one or more of the vulns, read about it
- Perhaps look it up in the CVE databases, and find CVSS
- Search for Cisco ACI and see if further vulns have been found over the years since 2019 (there have)

Hints:
There are these vulnerabilities to select from:

- Remote Code Execution on Leaf Switches over IPv6 via Local SSH Server (CVE-2019-1836, CVE2019-1803, and CVE-2019-1804)
- Cisco Nexus 9000 Series Fabric Switches ACI Mode Fabric Infrastructure VLAN Unauthorized Access Vulnerability (CVE-2019-1890)
- Cisco Nexus 9000 Series Fabric Switches Application Centric Infrastructure Mode Link Layer Discovery Protocol Buffer Overflow Vulnerability (CVE-2019-1901)
  This specific daemon is running as root
- Cisco Application Policy Infrastructure Controller REST API Privilege Escalation Vulnerability (CVE-2019-1889)

Solution:
When you have read about one of the vulnerabilities found and understood it, you are done.

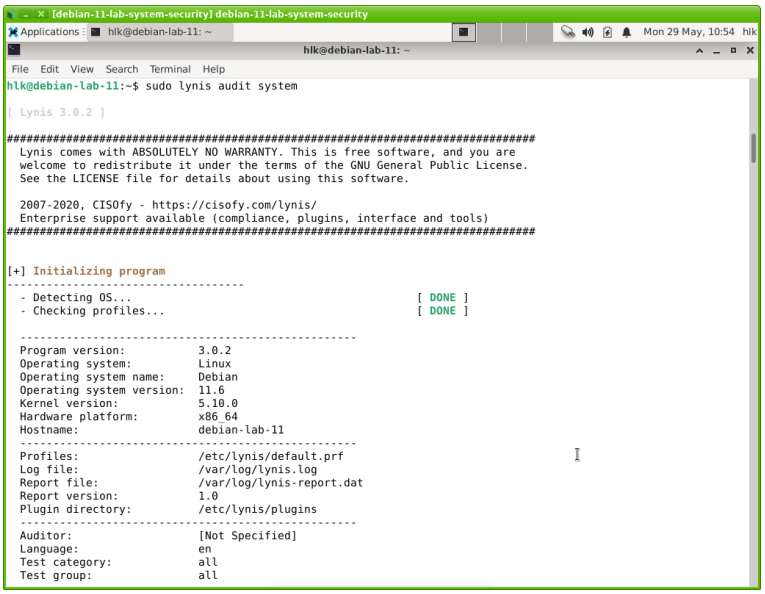
Discussion:
Shouldn't a product by Cisco for security purposes using Linux use the most basic of controls?

I found it horrible and inside the system there were many examples of insecure scripting, bad habits, no security designed or implemented – rather the opposite.

Running everything as root make every little mistake become a serious issue.

Exercise 40

⚠ Lynis Auditing, System hardening, and Compliance testing 20min



Objective:
Try out the Lynis tool for scanning your local Unix server.

Purpose:
Lynis is a quick tool to run, and gives concrete advise for things to change.

Suggested method:
Install using APT on your Debian and run the tool

Hints:
Use the web site for the tool Lynis `https://cisofy.com/lynis/` and audit your system

Solution:
When you have run the tool you are done. Better if you have browsed some of the output

Discussion:

Exercise 41

⚠ DNSSEC KeyTrap 20min

> A reminder of the value and relevance of DNS-OARC (https://dns-oarc.net/) in helping improve the security, and reliability of the Internet's Domain Name System:
>
> A critical denial-of-service vulnerability, known as KeyTrap (CVE-2023-50387 and the related CVE-2023-50868), would have allowed attackers to exhaust CPU resources on DNS resolvers across the Internet. Shortly after it was identified in late 2023, key personnel from all major DNS operators and vendors used OARC's facilities to coordinate the work to mitigate this vulnerability, with the last software patches being released just a couple of days ago. Amazing work from everyone, and just like Bill I'm proud of the community for getting this sorted before anything was leaked.

Source: Phil Regnauld
`https://www.linkedin.com/posts/philregnauld_lovedns-dns-dnsoarc-activity-7164303186424537088-PVCu`

Objective:
Research the DNSSEC related KeyTrap denial-of-service vulnerability – CVE-2023-50387

Purpose:
See how a real life vulnerability can affect systems, the implications and how it was coordinated.

Suggested method:
First look up the vulnerability using the CVE id. Then do some investigation into vulnerable products (most of the DNS resolver software) and discuss how this could affect a network.

Hints:
The DNSSEC vulnerability is affecting the design of the protocol, so many implementations would have similar code implemented. This would make it irresponsible for the researchers to just publish their findings. It ended up being DNS OARC that coordinated response and DNS vendors.
`https://www.dns-oarc.net/`

Solution:
When you have read about the vulnerability and discussed how to handle it a little, you are done.

Discussion:
Look up responsible disclosure. Why do we have a need for that.
Further links, posted by Bill Woodcock Executive Director at Packet Clearing House:

- `https://www.athene-center.de/en/news/press/key-trap`
- `https://nlnetlabs.nl/news/2024/Feb/13/unbound-1.19.1-released/`
- ISC has disclosed six vulnerabilities in BIND 9 (CVE-2023-4408, CVE-2023-5517, CVE-2023-5679, CVE-2023-6516, CVE-2023-50387, CVE-2023-50868)
  `https://seclists.org/oss-sec/2024/q1/125`
- `https://pi-hole.net/blog/2024/02/13/fixing-two-new-dnssec-vulnerabilities/#page-content`
- Certain DNSSEC aspects of the DNS protocol (in RFC 4033, 4034, 4035, 6840, and related RFCs) allow remote attackers to cause a denial of service
  `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-50387`
- The Closest Encloser Proof aspect of the DNS protocol (in RFC 5155 when RFC 9276 guidance is skipped) allows remote attackers to cause a denial of service
  `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-50868`

Exercise 42

ℹ SYN flooding 101 - 15min

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.

```
-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.

```
hping3 --flood -p 80 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
```

Try doing the most common attacks, RTFM hping3:

- ICMP flooding
- UDP flooding, try port 53 and port 123
- TCP flooding, try port 22 or port 80 on your debian perhaps

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.

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

Exercise 43

⚠ Centralized syslog 15min

Objective:
See how server syslog is configured on regular Unix/Linux.

Centralized syslogging and example system can demonstrate how easy it is to get started

Purpose:
The main idea of this exercise is to understand how easy network connected systems can send log data.

This should be the common case, sending logs off system - to avoid an attacker being able to hide tracks and logs from exploits performing intrusion and escalation.

Suggested method:
Log into your local Linux systems or network devices, see how syslog is configured.

Hints:
Look in the config file, may be in /etc/syslog or /etc/syslog-ng/syslog-ng.conf

Sample output from old-skool syslogd

```
*.err;kern.debug;auth.notice;authpriv.none;mail.crit    /dev/console
*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none /var/log/messages
kern.debug;user.info;syslog.info                        /var/log/messages
auth.info                                               /var/log/authlog
authpriv.debug                                          /var/log/secure
...
# Uncomment to log to a central host named "loghost".
#*.notice;auth,authpriv,cron,ftp,kern,lpr,mail,user.none        @loghost
#kern.debug,user.info,syslog.info                               @loghost
#auth.info,authpriv.debug,daemon.info                           @loghost
```

Solution:
When you understand how to configure syslog from a couple of devices and has looked up which protocol and port it uses. (default is 514/udp)

Discussion:
There are syslog senders for Windows too. Other systems define their own format for sending, example Beats - lightweight data shippers https://www.elastic.co/products/beats

I recommend using the elastic stack, previously the ELK stack, https://www.elastic.co/products/. The products can be used without license and can give a lot of experience with this kind of product. This will enable you to better describe your logging needs for evaluating other products.

This is done using Logstash as the server - can also receive SNMP traps!

Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite stash. - often Elasticsearch https://www.elastic.co/products/logstash

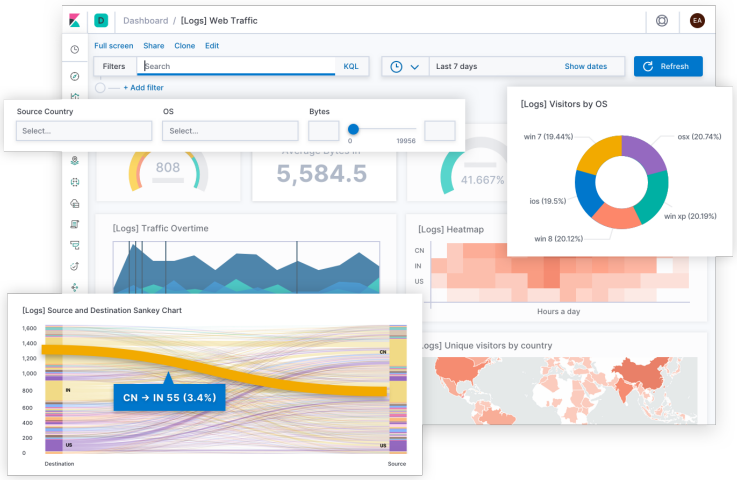
Other very popular systems are:

- Splunk https://www.splunk.com
- Graylog https://www.graylog.org/
- InfluxDB https://www.influxdata.com/
- Grafana The open platform for analytics and monitoring https://grafana.com/ also includes Grafana Loki now https://grafana.com/oss/loki/
- Prometheus Monitoring system & time series database https://prometheus.io/

Remember doing logging og performance metrics can also become a security charateristics. Availability is a critical metric for most commercial systems.

Exercise 44

ℹ Getting started with the Elastic Stack 15min



Screenshot from `https://www.elastic.co/kibana`

Objective:
Get ready to start using Elasticsearch, read - but dont install.

Purpose:
We need some tools to demonstrate integration. Elasticsearch is a search engine and ocument store used in a lot of different systems, allowing cross application integration.

Suggested method:
Visit the web page for Getting started with the Elastic Stack :
`https://www.elastic.co/guide/en/elastic-stack-get-started/current/get-started-elastic-stack.html`

Read about the tools, and the steps needed for manual installation.

You dont need to install the tools currently, I recommend using Debian and Ansible for bringing up Elasticsearch. You are of course welcome to install, or try the Docker method.

Hints:
Elasticsearch is the name of the search engine and document store. Today Elastic Stack contains lots of different parts.

We will focus on these parts:

- Elasticsearch - the core engine
- Logstash - a tool for parsing logs and other data.
  `https://www.elastic.co/logstash`

  > "Logstash dynamically ingests, transforms, and ships your data regardless of format or complexity. Derive structure from unstructured data with grok, decipher geo coordinates from IP addresses, anonymize or exclude sensitive fields, and ease overall processing."

- Kibana - a web application for accessing and working with data in Elasticsearch
  `https://www.elastic.co/kibana`

Solution:
When you have browsed the page you are done.

Discussion:
You can read more about Elasticsearch at the wikipedia page:
`https://en.wikipedia.org/wiki/Elasticsearch`

Exercise 45

❶ Run Elasticsearch in Containers 30min

Objective:
Run an Elasticsearch stack with Suricata IDS in containers

Purpose:
See how easy it is to test various solutions, even if they are complicated

Suggested method:
Use the kickstart-2-selks.pdf document in Fronter.

The steps are also outlined in their repository `https://github.com/StamusNetworks/SELKS.git`

I would like for you to install Docker and try out SELKS `https://www.stamus-networks.com/selks`

If you want to use the same as me with Debian VM, which installs in less than 30minutes:

- ☐ Install a basic Debian 12 Bookworm with Sudo configured
- ☐ Install git and Ansible, see our exercise:
  `sudo apt install git ansible`
- ☐ Clone the Github repo: https://github.com/kramse/kramse-labs
  `git clone https://github.com/kramse/kramse-labs`
- ☐ Go into this repository and install Docker, there is a small README.md too:
  `cd kramse-labs/docker-install` and then `ansible-playbook 1-dependencies.yml`
- ☐ Enable Docker: `systemctl enable docker` and reboot the VM
- ☐ Check docker, `docker run hello-world`
- ☐ Clone the SELKS repository:
  `git clone https://github.com/StamusNetworks/SELKS.git`
- ☐ Go into this and run docker-compose as described in the instructions:
  `https://github.com/StamusNetworks/SELKS/wiki/Docker`
  make sure to select the right network interface, so Suricata can sniff packets I did NOT install Portainer
- ☐ Use a browser to access the platform on `https://127.0.0.1` – and enjoy!

This will provide a basic Elasticsearch version 7, with Kibana and Suricata

Hints:
Make sure your Debian has enough memory, 8Gb will run, less will be tough, more will be better.

Solution:
When you have seen the system on the instructors PC your are done.

Discussion:
There are lots of projects that allow you to test solution from containers. This is often much easier and faster then following the instructions for installing the project. On the other hand, you may want to follow install instructions if it for a production setup.

Exercise 46

➊ Create Kibana Dashboard 15min

Objective:
See Kibana and understand how it is configured.

Purpose:
Kibana is a very popular system for creating dashboards from data in elasticsearch.

Learning how to create and import dashboards is a good exercise.

Suggested method:
Instructor will provide a method for running Elasticsearch and Kibana for this exercise. See the previous exercise ☺

Note: usually Kibana should be available on port 5601 on localhost (127.0.0.1) only! It is recommended to keep this configuration and then add a web server like Nginx or Apache in front. This will further allow authentication and other features.

Using Firefox visit Kibana on the link provided by the instructor.

If this is the first time you need to select `logstash-*` as a default index. Note: Kibana is an advanced and powerful tool in itself.

Hints:
Logstash and Elastic stack are a great way to get started with dashboarding.

However, running a big installation is harder than it looks. Make sure to have multiple servers and good monitoring.

Solution: When you have browsed Kibana, seen how you can add graphs and combine them into dashboards - using the GUI you are done. Previously creating dashboards was harder and often required programming knowledge.

Discussion:
Making dashboard are an art form. We will NOT start creating beautiful dashboards.

If you want, there is a SELKS LiveCD dedicated to suricata which also includes more tools for administration of rules and getting alerts:
https://www.stamus-networks.com/open-source/

Dashboards can be exported as JSON into files, and can be loaded using shell commands, example the ones from: https://github.com/StamusNetworks/KTS7

The commands are similar to

```
git clone https://github.com/StamusNetworks/KTS7.git
cd KTS7
bash load.sh
```

Note: KTS version needs to match Elasticsearch version! So for ES version 7, use KTS7

Exercise 47

⚠ File System Forensics 30min

Objective:
Open a file system dump

Purpose:
Learn a bit of computer forensics using a free tool.

Suggested method:
We will use a free toolkit, and an older version - easier to install.

The Sleuth Kitő is a collection of command line tools and a C library that allows you to analyze disk images and recover files from them. It is used behind the scenes in Autopsy and many other open source and commercial forensics tools.

Autopsyő is an easy to use, GUI-based program that allows you to efficiently analyze hard drives and smart phones. It has a plug-in architecture that allows you to find add-on modules or develop custom modules in Java or Python.

http://www.sleuthkit.org/

1. Install tools
2. Acquire test images - download file system images
3. Open test images using tools

Installing the tools is described on the web page, but using apt on Kali Linux should be OK. Note: this is not the newest version!

Test images can be found at:
http://dftt.sourceforge.net/

Example, the EXT3FS file system:
http://dftt.sourceforge.net/test4/index.html

For this do the following - tested on Kali Linux:

1. Install tools
   ```
   apt-get install autopsy sleuthkit testdisk
   ```
2. Acquire test images - download and unzip
   ```
   cd ~; mkdir forensic;cd ~/forensic;unzip
   ```


3. Start autopsy from command line
4. Open test images using tools, use a browser http://localhost:9999/autopsy
5. Add a new case, fill out wizards case: "My case", investigator: "hlk"
6. Add host, fill out wizard, name: "host1", time zone: "CEST"
7. Add image file - location full path to the file containing a file system, choose type: "partition" with symlink is fine
8. Then use the analyze button to start analyzing this file system
9. Click and get a feel for the tool

```
user@KaliVM:~$
user@KaliVM:~$ mkdir forensic
user@KaliVM:~$ cd forensic/
user@KaliVM:~/forensic$ unzip ../Downloads/4-kwsrch-ext3.zip
Archive:  ../Downloads/4-kwsrch-ext3.zip
  inflating: 4-kwsrch-ext3/COPYING-GNU.txt
  inflating: 4-kwsrch-ext3/README.txt
  inflating: 4-kwsrch-ext3/ext3-img-kw-1.dd
  inflating: 4-kwsrch-ext3/index.html
user@KaliVM:~/forensic$ pwd
/home/user/forensic
user@KaliVM:~/forensic$
```

Note: I run as user hlk, so note down the full path for the imagefile, in my case `/home/user/forensic/4-kwsrch-ext3/ext3-img-kw-1.dd`

```
root@KaliVM:~# autopsy

================================================================================

                     Autopsy Forensic Browser
                http://www.sleuthkit.org/autopsy/
                           ver 2.24

================================================================================
Evidence Locker: /var/lib/autopsy
Start Time: Wed Jun  5 16:16:12 2019
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

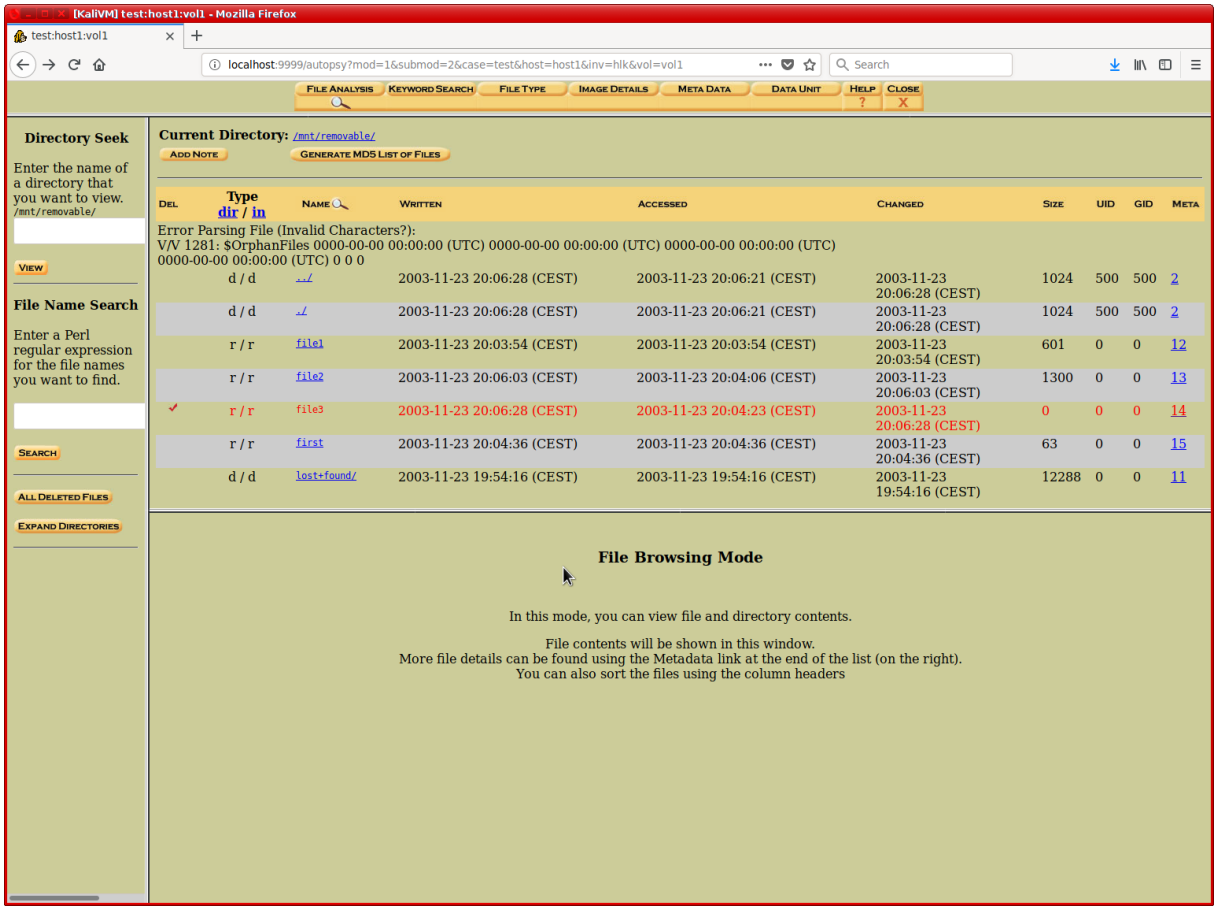
Hints:
Generating a time line of timestamps with date created, modification etc. can sometimes highlight the

interesting times. A hacker breaking in and replacing a file would often end up having modified time stamps.

If you want to automate or use the command line for other reasons there are some documentation available, example `http://wiki.sleuthkit.org/index.php?title=FS_Analysis`

Solution:
When you team has opened at least one file system from an image file, you are done.

Hopefully you should be able reach something like this:



Discussion:
These tools are quite old, but still very usable. The older tools often came from research, legal and government agencies.

We use them to see the filesystem data directly, without modification – which is essential for any forensics tool. There are more userfriendly tools these days.

Some are targetted at recovering data, like lost photos on USB devices, others are targetted at enterprise use.

Exercise 48

❶ Clean or rebuild a server 20min

The brave new world of IPv6

Objective:
Think about a hacked system, how can you clean such a system?

Purpose:
Realize that you can never be completely sure the system really is secure.

Suggested method:
Consider the system from exercise 31

We created a back door in this system: (We created it in /tmp so it may have been deleted, but lets say it was created in /sbin instead)

Commands executed:

```
root@debian:~# rm /tmp/.xxsh
root@debian:~# cp /bin/dash /tmp/.xxsh
root@debian:~# chmod +sw /tmp/.xxsh
```

Is this the only file left by the attacker?

Did they change other files, configurations, added users, changed user passwords?

Hints:
A forensics investigation might perform a complete dump of the file systems and use TASK/Autopsy. Then by generating a timeline it might be possible to find the back door files. Perhaps.

Solution:
Remove the back door, and associated hacked accounts.

In real life you would: Rebuild your Debian server. Automate the setup of critical systems. Have good backup of critical data.

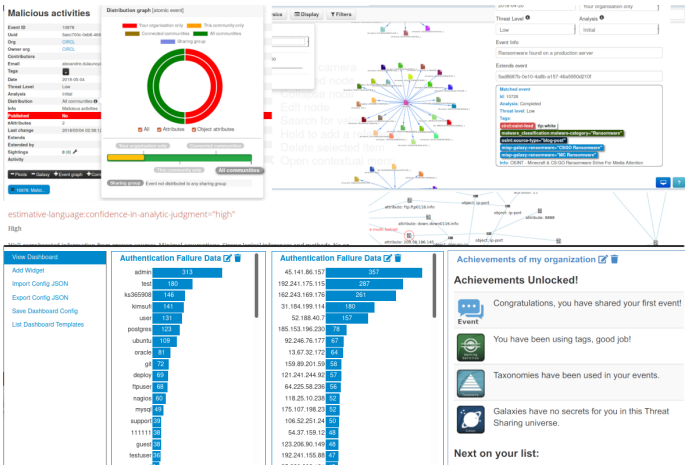
Discussion:
Cleaning systems and whole environments is very hard.

An attacker may have spent only 30 minutes, but the investigation might take 100 hours. This is a huge difference in resources spent.

No such thing as Was just browsing the system

Exercise 49

❶ Install MISP Project 45min



Objective:
Try installing the application MISP Project 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 analyse

Suggested method:
Run the program from a Linux VM

OR 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
```

Either way go to the web site and decide an installation path:
`https://www.misp-project.org/download/`

Hints:
A VM images 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.

Exercise 50

ℹ Cloud environments influence on incident response 20min

Objective:
Talk about the difference in computer forensics in cloud environments.

Cloud environments, or mixed environments between cloud and traditional environments present new challenges.

Purpose:
Discuss what sources of information is available.

Traditional computer forensics often use these sources:

- Network forensics
- Applications logs
- Operating system logs
- Disk imaging

Cloud environments can often use these sources:

- Logging from authentication
- Limited network forensics
- Applications logs

This relies more on the capabilities of the cloud vendor and often cloud environments are also much more dynamic. Some services are also provided by the cloud vendor, separating the management away from the customer configured environment - with good or bad consequences for computer forensics.

Suggested method:
Discuss in your group, how would you investigate an incident in your solutions.

Has any in our group performed incident handling in cloud environments.

Hints:
NIST has a few papers about this subject.

Example: Identifying Evidence for Implementing a Cloud Forensic Analysis Framework https://www.nist.gov/publications/identifying-evidence-implementing-cloud-forensic-analysis-framework

Solution:
Download the linked paper and browse it. It contains an example cloud and the conclusion scratches the surface of what a cloud maybe should provide.

Discussion:
Cloud computer forensics seem immature, but must be researched.

If you organization relies on cloud computing it is critcal to update incident handling procedures for these new challenges.

Exercise 51

ℹ Evaluate Scope Towards PCI

Objective:
Evaluate your infrastructure, quick gap analysis for becoming CIS/PCI compliant

Purpose:
Consider the term scope and influence on securing an environment.

Suggested method:
Consider a small network like the one used for exercises. Consider the requirements for running exercises, why did we isolate this part from the rest of the network.

What are the parts used:

- Router(s), switches, network equipment
- Firewall(s)
- Design DMZ, LAN, WLAN
- "Servers" - your laptops with Debian
- Larger systems and evironments used web servers, email services

Write down the current status, What are the network parts, organization and applications.

Consider the impact of adding wireless network to this setup. Is this a requirement, could we do without it.

Consider if firewalls would be best practice or an actual requirement, how to implement this requirement for your organization.

Which email security standards do you implement now, which ones should you consider.

Hints:
Running a host firewall seldom hurts performance today. So turn on the firewall as described in exercise ⚠ Enable UFW firewall - 10min

Solution:
When you have browsed the CIS web site and documents, and documented the scope in 3-5 sentences or a small table. You are done.

Discussion:

What is more important:

- Personal data, non-biometric
- Financial data, like credit card data
- Biometric data iris, DNA etc.

- Password data

What if they are leaked, how to recover?