Systems Security Exercises

Intro to IT-security 2024

Henrik Kramselund
xhek@kea.dk

April 15, 2024

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

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

Contents

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 intro-to-it-security-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

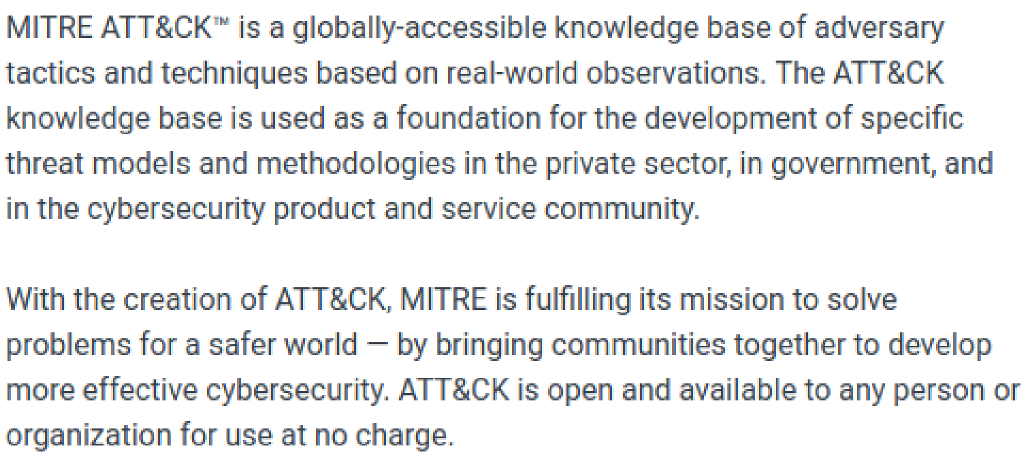
Most exercises follow the same procedure and has the following content:

- **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

Mitre ATT&CK Framework 15min



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 2

ℹ 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.

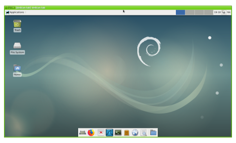
Debian Linux is a free operating system platform.

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 Debian VM 10min



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

We need a Debian 11 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 4

⚠ 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 5

⚠ 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 6

ℹ Git tutorials - 15min

Objective:
Try the program Git locally on your workstation

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

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

Suggested method:
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.

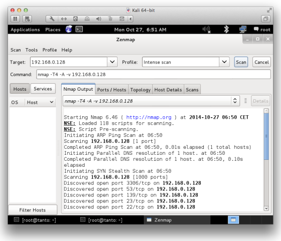
Note: you don't need an account on Github to download/clone repositories, but having an acccount allows you to save repositories yourself and is recommended.

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 7

⚠ Discover active systems ping and port sweep 15min



Target: 192.168.0.128 Profile: Intense scan

Command: nmap -T4 -A -v 192.168.0.128

```
nmap -T4 -A -v 192.168.0.128

Starting Nmap 6.46 ( http://nmap.org ) at 2014-10-27 06:50 CET
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 06:50
Scanning 192.168.0.128 [1 port]
Completed ARP Ping Scan at 06:50, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:50
Completed Parallel DNS resolution of 1 host. at 06:50, 0.10s
elapsed
Initiating SYN Stealth Scan at 06:50
Scanning 192.168.0.128 [1000 ports]
Discovered open port 3306/tcp on 192.168.0.128
Discovered open port 53/tcp on 192.168.0.128
Discovered open port 139/tcp on 192.168.0.128
Discovered open port 23/tcp on 192.168.0.128
Discovered open port 22/tcp on 192.168.0.128
```

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:
Install Nmap on your Debian VM, `apt install nmap`. Try different scans:

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

Use the prefixes and IP addresses handed out by the instructor! They may be different from the ones below!

Hints:
Try nmap in sweep mode

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.*`

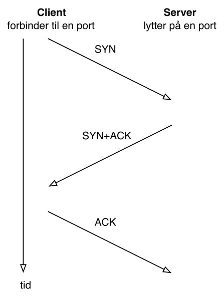
Discussion:
Quick scans quickly reveal interesting hosts, ports and services

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 8

⚠ Execute nmap TCP and UDP port scan 15min



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 9

⚠ 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. Use more advanced features in Nmap to discover services.

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.

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.

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

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

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.

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 10

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:

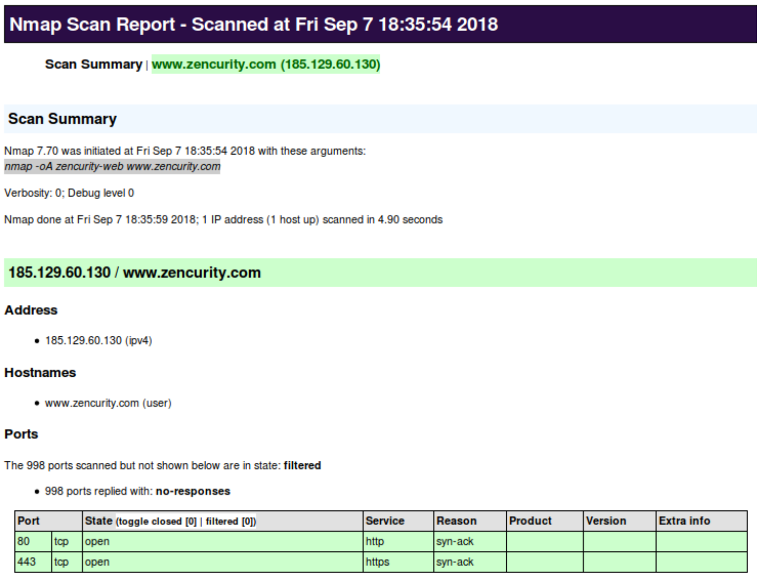
```
-iL <inputfilename>: Input from list of hosts/networks
-oA outputbasename: output in all formats, see later
```

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 11

ℹ 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 12

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 13

⚠ 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 the parts metioned here:

- Install Required Packages – only Mariadb
- Add librenms user
- Download LibreNMS
- Set permissions

Install Required Packages:

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

and follow only the instructions in Configure MariaDB – the ones with Debian 11!

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

Configure MariaDB – Debian 11

Edit `/etc/mysql/mariadb.conf.d/50-server.cnf`

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 14

⚠ RBAC Access permissions on GitHub up to 30min

Objective:
See actual real life example of permissions.

Note: This exercise requires a GitHub account, so make sure your group has one. Maybe do groups of 3-4 for more discussion.

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 15

⚠ 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
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.

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 16

🛈 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:l5esp66lQArF0lXqOoHnxpg8zRS6shK8nx9KGf+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 17

⚠ 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.

Update 2024: it seems VMware just released some patches for important updates to their ESXi products. See if you can locate an article either on the internet or at VMware.com describing these problems.

Exercise 18

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 19

⚠ CIS Benchmarks 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:

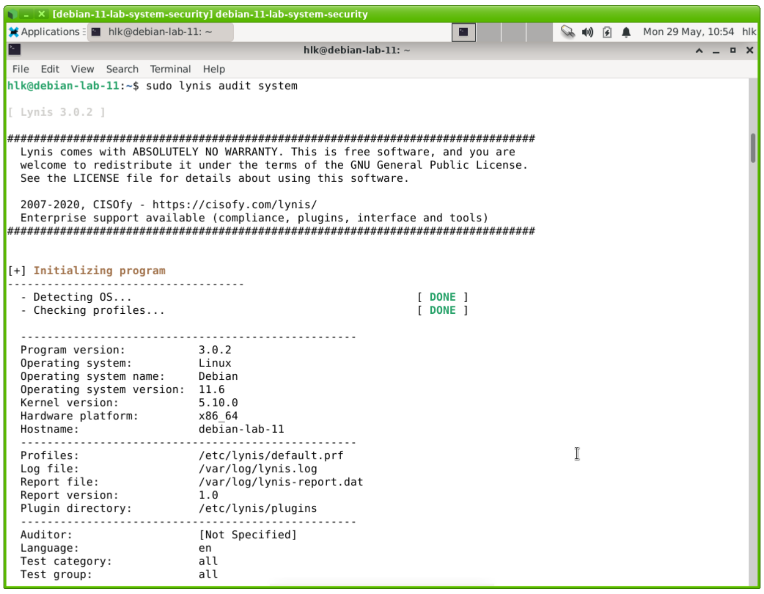
`https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark`

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 20

⚠ Lynis Auditing, System hardening, and Compliance testing 30min


[debian-11-lab-system-security] debian-11-lab-system-security
Applications
hlk@debian-lab-11: ~
Mon 29 May, 10:54 hlk
hlk@debian-lab-11: ~
File Edit View Search Terminal Help

```
hlk@debian-lab-11:~$ sudo lynis audit system

[ Lynis 3.0.2 ]

################################################################################
  Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
  welcome to redistribute it under the terms of the GNU General Public License.
  See the LICENSE file for details about using this software.

  2007-2020, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)
################################################################################


[+] Initializing program
------------------------------------
  - Detecting OS...                                           [ DONE ]
  - Checking profiles...                                      [ DONE ]

  ---------------------------------------------------
  Program version:           3.0.2
  Operating system:          Linux
  Operating system name:     Debian
  Operating system version:  11.6
  Kernel version:            5.10.0
  Hardware platform:         x86_64
  Hostname:                  debian-lab-11
  ---------------------------------------------------
  Profiles:                  /etc/lynis/default.prf
  Log file:                  /var/log/lynis.log
  Report file:               /var/log/lynis-report.dat
  Report version:            1.0
  Plugin directory:          /etc/lynis/plugins
  ---------------------------------------------------
  Auditor:                   [Not Specified]
  Language:                  en
  Test category:             all
  Test group:                all
```

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 21

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 22

ℹ Example Policies up to 25min

Objective:
See real world high level policies and discuss where to start. There is no need to re-invent the wheel over and over. Re-use things already made!

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 23

⚠ Centralized syslog 30min

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.