PROJECT: VULNER PENETRATION TESTING

Phua Tong Huat (S32)

Class Code: CFC020823

Trainer: Kar Wei

Table of Contents

Introduction

In this report, we will be looking at how we can automate the process of finding vulnerabilities in local networks to stay up to date and protect our assets. Automation will save time and effort as it reduces the need for work hours to conduct repeated processes manually and importantly, it can also reduce the probability of human errors when carrying out repetitive tasks.

The main objectives of the automation process are:

1. Use Nmap port scan to scan the network for ports and services
   - Doing a port scan allows us to identify vulnerabilities, access points and any other potentially sensitive information related to the open ports.
2. Map vulnerabilities using Nmap-vulners scan
   - Nmap-vulners queries the Vulners exploit database when we use the NSE script. We are able to discover CVEs related to the open ports and services
   - Nmap-vulners is also preferred over Nmap-vuln script as it is much shorter and more realistic to run over large networks
3. Check for weak login credentials using Medusa to brute force the various services
   - Testing for weak login credentials is a critical security measure as they can be used to exploit the system and may lead to potential data breaches or unauthorized access to sensitive information
   - Therefore it is very important to identify weak login credentials and rectify the vulnerabilities.

Methodologies

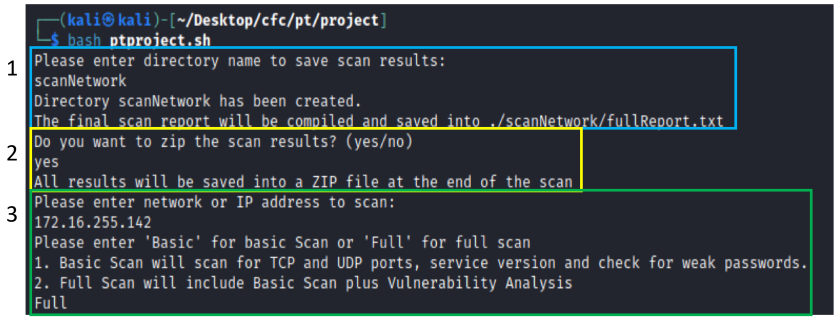
For this project, a bash script was written to get the following from the user:

1. Getting a network from user to conduct a scan
2. Check for weak credentials
3. Mapping for vulnerabilities

For the purpose of this report, the scan results are based on scanning an IP address instead of a full network as a full network scan is very time consuming. The password list provided by the user in the scan result is also a short list with just 2 passwords to hasten the Medusa brute force process.

Methodologies

1. Getting user input and criteria to initiate a scan

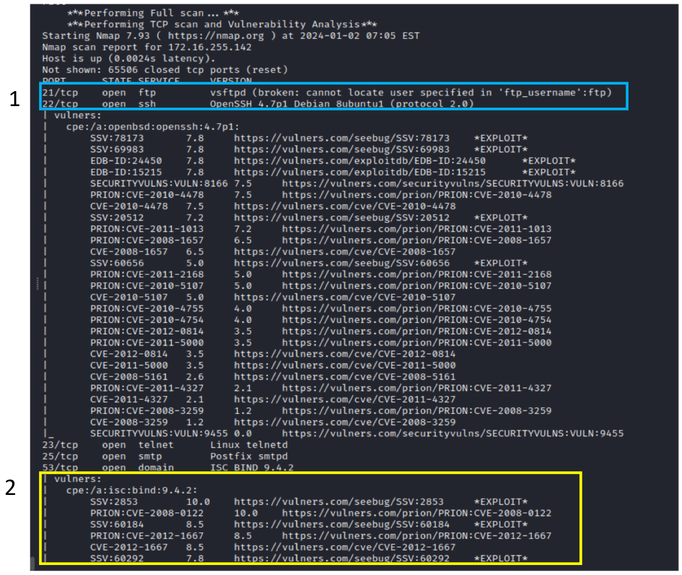


```
┌──(kali㉿kali)-[~/Desktop/cfc/pt/project]
└─$ bash ptproject.sh
Please enter directory name to save scan results:
scanNetwork
Directory scanNetwork has been created.
The final scan report will be compiled and saved into ./scanNetwork/fullReport.txt
Do you want to zip the scan results? (yes/no)
yes
All results will be saved into a ZIP file at the end of the scan
Please enter network or IP address to scan:
172.16.255.142
Please enter 'Basic' for basic Scan or 'Full' for full scan
1. Basic Scan will scan for TCP and UDP ports, service version and check for weak passwords.
2. Full Scan will include Basic Scan plus Vulnerability Analysis
Full
```

1. User being asked for directory name to save the scan results. Thereafter, a directory with the given name will be created and all scan output will be saved into that specific directory
2. User given the option to save the scan result into ZIP folder
3. Getting user input of IP address or network to scan and also whether to conduct a basic or full scan. User can choose to scan a entire network or just 1 IP address as a full network scan might take up some time. Thus this gives user the flexibility to select which the scale of the scan and also the scan type.

Methodologies

2. Results of full scan, which consist of port scans, services, vulnerability analysis

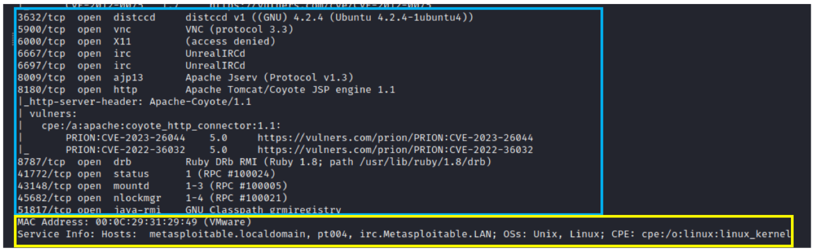


```
    ***Performing Full scan ... ***
    ***Performing TCP scan and Vulnerability Analysis***
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-02 07:05 EST
Nmap scan report for 172.16.255.142
Host is up (0.0024s latency).
Not shown: 65506 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd (broken: cannot locate user specified in 'ftp_username':ftp)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:4.7p1:
|       SSV:78173       7.8     https://vulners.com/seebug/SSV:78173    *EXPLOIT*
|       SSV:69983       7.8     https://vulners.com/seebug/SSV:69983    *EXPLOIT*
|       EDB-ID:24450    7.8     https://vulners.com/exploitdb/EDB-ID:24450      *EXPLOIT*
|       EDB-ID:15215    7.8     https://vulners.com/exploitdb/EDB-ID:15215      *EXPLOIT*
|       SECURITYVULNS:VULN:8166 7.5     https://vulners.com/securityvulns/SECURITYVULNS:VULN:8166
|       PRION:CVE-2010-4478     7.5     https://vulners.com/prion/PRION:CVE-2010-4478
|       CVE-2010-4478   7.5     https://vulners.com/cve/CVE-2010-4478
|       SSV:20512       7.2     https://vulners.com/seebug/SSV:20512    *EXPLOIT*
|       PRION:CVE-2011-1013     7.2     https://vulners.com/prion/PRION:CVE-2011-1013
|       PRION:CVE-2008-1657     6.5     https://vulners.com/prion/PRION:CVE-2008-1657
|       CVE-2008-1657   6.5     https://vulners.com/cve/CVE-2008-1657
|       SSV:60656       5.0     https://vulners.com/seebug/SSV:60656    *EXPLOIT*
|       PRION:CVE-2011-2168     5.0     https://vulners.com/prion/PRION:CVE-2011-2168
|       PRION:CVE-2010-5107     5.0     https://vulners.com/prion/PRION:CVE-2010-5107
|       CVE-2010-5107   5.0     https://vulners.com/cve/CVE-2010-5107
|       PRION:CVE-2010-4755     4.0     https://vulners.com/prion/PRION:CVE-2010-4755
|       PRION:CVE-2010-4754     4.0     https://vulners.com/prion/PRION:CVE-2010-4754
|       PRION:CVE-2012-0814     3.5     https://vulners.com/prion/PRION:CVE-2012-0814
|       PRION:CVE-2011-5000     3.5     https://vulners.com/prion/PRION:CVE-2011-5000
|       CVE-2012-0814   3.5     https://vulners.com/cve/CVE-2012-0814
|       CVE-2011-5000   3.5     https://vulners.com/cve/CVE-2011-5000
|       CVE-2008-5161   2.6     https://vulners.com/cve/CVE-2008-5161
|       PRION:CVE-2011-4327     2.1     https://vulners.com/prion/PRION:CVE-2011-4327
|       CVE-2011-4327   2.1     https://vulners.com/cve/CVE-2011-4327
|       PRION:CVE-2008-3259     1.2     https://vulners.com/prion/PRION:CVE-2008-3259
|       CVE-2008-3259   1.2     https://vulners.com/cve/CVE-2008-3259
|_      SECURITYVULNS:VULN:9455 0.0     https://vulners.com/securityvulns/SECURITYVULNS:VULN:9455
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
| vulners:
|   cpe:/a:isc:bind:9.4.2:
|       SSV:2853        10.0    https://vulners.com/seebug/SSV:2853     *EXPLOIT*
|       PRION:CVE-2008-0122     10.0    https://vulners.com/prion/PRION:CVE-2008-0122
|       SSV:60184       8.5     https://vulners.com/seebug/SSV:60184    *EXPLOIT*
|       PRION:CVE-2012-1667     8.5     https://vulners.com/prion/PRION:CVE-2012-1667
|       CVE-2012-1667   8.5     https://vulners.com/cve/CVE-2012-1667
|       SSV:60292       7.8     https://vulners.com/seebug/SSV:60292    *EXPLOIT*
```

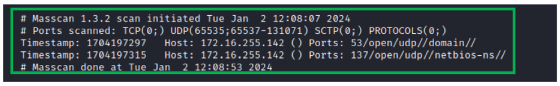
1. TCP and service scan being performed.
2. Vulnerability analysis being conducted using the Nmap-vulner script[1]. This script is more realistic as it is shorter and will display the related CVEs accordingly.

Methodologies

2. Results of full scan, which consist of port scans, services, vulnerability analysis



```
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
|_http-server-header: Apache-Coyote/1.1
| vulners:
|   cpe:/a:apache:coyote_http_connector:1.1:
|       PRION:CVE-2023-26044    5.0     https://vulners.com/prion/PRION:CVE-2023-26044
|_      PRION:CVE-2022-36032    5.0     https://vulners.com/prion/PRION:CVE-2022-36032
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
41772/tcp open  status      1 (RPC #100024)
43148/tcp open  mountd      1-3 (RPC #100005)
45682/tcp open  nlockmgr    1-4 (RPC #100021)
51817/tcp open  java-rmi    GNU Classpath grmiregistry
MAC Address: 00:0C:29:31:29:49 (VMware)
Service Info: Hosts:  metasploitable.localdomain, pt004, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```



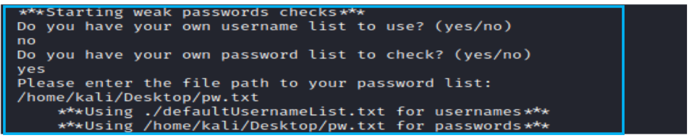
```
# Masscan 1.3.2 scan initiated Tue Jan  2 12:08:07 2024
# Ports scanned: TCP(0;) UDP(65535;65537-131071) SCTP(0;) PROTOCOLS(0;)
Timestamp: 1704197297   Host: 172.16.255.142 () Ports: 53/open/udp//domain//
Timestamp: 1704197315   Host: 172.16.255.142 () Ports: 137/open/udp//netbios-ns//
# Masscan done at Tue Jan  2 12:08:53 2024
```

1. Both the basic and full scan will do a full port scan, meaning all 65535 ports will be scanned instead of the top 1000 ports. This is to ensure all ports are covered and vulnerabilities hidden in uncommon port would not go undetected.
2. Additional information of the machine on the scanned network is also essential in identifying vulnerabilities and protecting the network.
3. UDP scan is done using Masscan to reduce the duration of the scan.

Methodologies

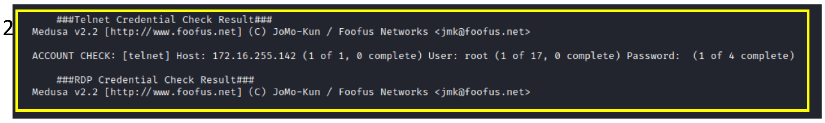
3. Login Credentials Checks

1



```
***Starting weak passwords checks***
Do you have your own username list to use? (yes/no)
no
Do you have your own password list to check? (yes/no)
yes
Please enter the file path to your password list:
/home/kali/Desktop/pw.txt
    ***Using ./defaultUsernameList.txt for usernames***
    ***Using /home/kali/Desktop/pw.txt for passwords***
```

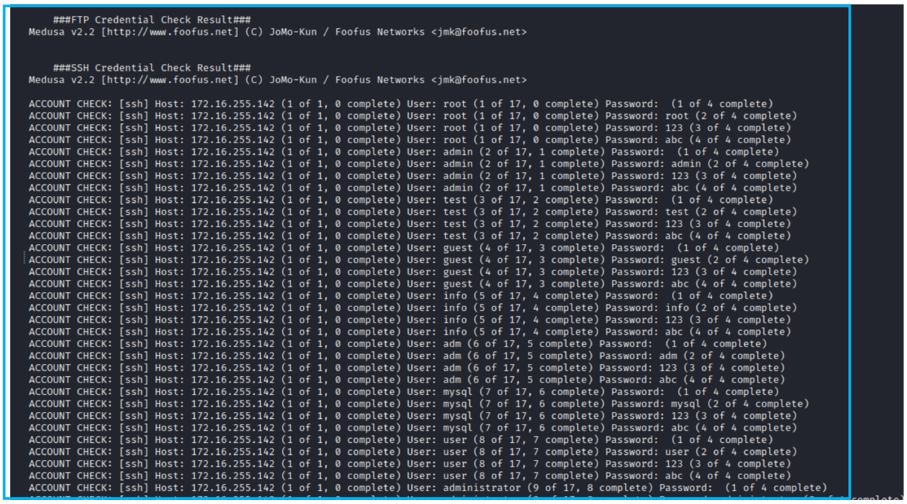
2



```
    ###Telnet Credential Check Result###
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [telnet] Host: 172.16.255.142 (1 of 1, 0 complete) User: root (1 of 17, 0 complete) Password:  (1 of 4 complete)

    ###RDP Credential Check Result###
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
```

1. The script has a default list for username(./defaultUsernameList.txt) and password (./defaultPwList.txt) for the login credential checks. However, user can also choose to use their own username list or password list. This option allows us to conduct a more comprehensive check on the credentials as the list from the user might be more accurate.
2. Medusa[2] was used to perform the login credential checks on 4 commonly used services and ports: FTP port 21, SSH port 22, Telnet port 23 and RDP port 3389

3

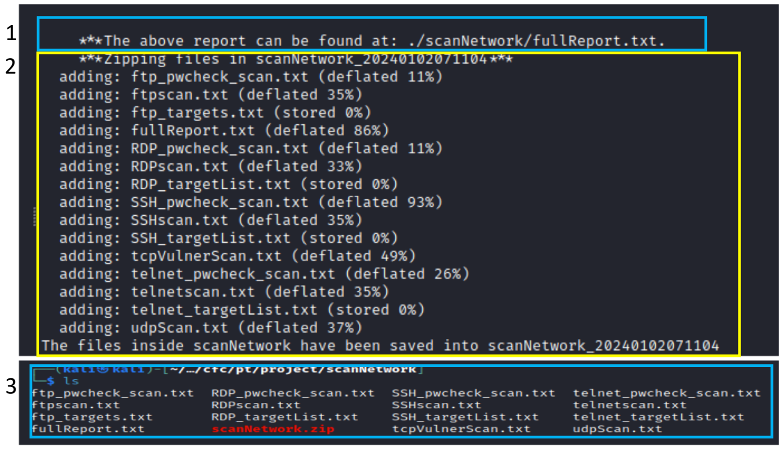


```
    ###FTP Credential Check Result###
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>


    ###SSH Credential Check Result###
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: root (1 of 17, 0 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: root (1 of 17, 0 complete) Password: root (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: root (1 of 17, 0 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: root (1 of 17, 0 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: admin (2 of 17, 1 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: admin (2 of 17, 1 complete) Password: admin (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: admin (2 of 17, 1 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: admin (2 of 17, 1 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: test (3 of 17, 2 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: test (3 of 17, 2 complete) Password: test (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: test (3 of 17, 2 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: test (3 of 17, 2 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: guest (4 of 17, 3 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: guest (4 of 17, 3 complete) Password: guest (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: guest (4 of 17, 3 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: guest (4 of 17, 3 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: info (5 of 17, 4 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: info (5 of 17, 4 complete) Password: info (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: info (5 of 17, 4 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: info (5 of 17, 4 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: adm (6 of 17, 5 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: adm (6 of 17, 5 complete) Password: adm (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: adm (6 of 17, 5 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: adm (6 of 17, 5 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: mysql (7 of 17, 6 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: mysql (7 of 17, 6 complete) Password: mysql (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: mysql (7 of 17, 6 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: mysql (7 of 17, 6 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: user (8 of 17, 7 complete) Password:  (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: user (8 of 17, 7 complete) Password: user (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: user (8 of 17, 7 complete) Password: 123 (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: user (8 of 17, 7 complete) Password: abc (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.255.142 (1 of 1, 0 complete) User: administrator (9 of 17, 8 complete) Password:  (1 of 4 complete)
```

3. Process of the brute force attempts being shown. The uploaded password list used in this example is a very short list with 2 password to showcase the use of user's list

Methodologies

4. **Scan result being saved into ZIP file**



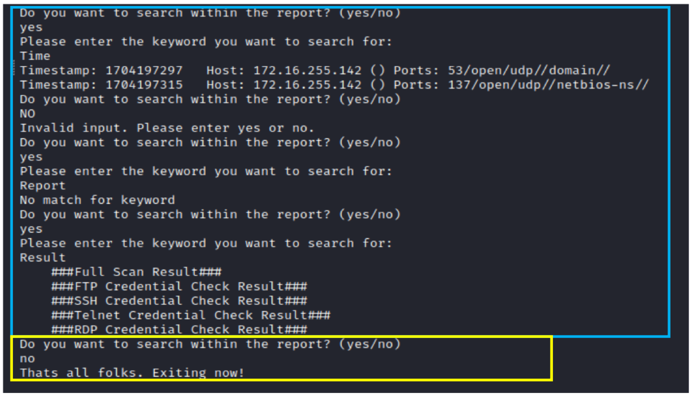
```
   **The above report can be found at: ./scanNetwork/fullReport.txt.
   **Zipping files in scanNetwork_20240102071104**
  adding: ftp_pwcheck_scan.txt (deflated 11%)
  adding: ftpscan.txt (deflated 35%)
  adding: ftp_targets.txt (stored 0%)
  adding: fullReport.txt (deflated 86%)
  adding: RDP_pwcheck_scan.txt (deflated 11%)
  adding: RDPscan.txt (deflated 33%)
  adding: RDP_targetList.txt (stored 0%)
  adding: SSH_pwcheck_scan.txt (deflated 93%)
  adding: SSHscan.txt (deflated 35%)
  adding: SSH_targetList.txt (stored 0%)
  adding: tcpVulnerScan.txt (deflated 49%)
  adding: telnet_pwcheck_scan.txt (deflated 26%)
  adding: telnetscan.txt (deflated 35%)
  adding: telnet_targetList.txt (stored 0%)
  adding: udpScan.txt (deflated 37%)
The files inside scanNetwork have been saved into scanNetwork_20240102071104
```

```
┌──(kali㉿kali)-[~/…/cfc/pt/project/scanNetwork]
└─$ ls
ftp_pwcheck_scan.txt  RDP_pwcheck_scan.txt  SSH_pwcheck_scan.txt  telnet_pwcheck_scan.txt
ftpscan.txt           RDPscan.txt           SSHscan.txt           telnetscan.txt
ftp_targets.txt       RDP_targetList.txt    SSH_targetList.txt    telnet_targetList.txt
fullReport.txt        scanNetwork.zip       tcpVulnerScan.txt     udpScan.txt
```

1. All the different components of the scan will be compiled and saved at the location as shown.
2. All scan results being saved into a ZIP file.
3. All individuals scan results are also saved separately for easier reference. The compiled result will be saved into fullReport.txt.

Methodologies

5. Search function



```
Do you want to search within the report? (yes/no)
yes
Please enter the keyword you want to search for:
Time
Timestamp: 1704197297   Host: 172.16.255.142 () Ports: 53/open/udp//domain//
Timestamp: 1704197315   Host: 172.16.255.142 () Ports: 137/open/udp//netbios-ns//
Do you want to search within the report? (yes/no)
NO
Invalid input. Please enter yes or no.
Do you want to search within the report? (yes/no)
yes
Please enter the keyword you want to search for:
Report
No match for keyword
Do you want to search within the report? (yes/no)
yes
Please enter the keyword you want to search for:
Result
    ###Full Scan Result###
    ###FTP Credential Check Result###
    ###SSH Credential Check Result###
    ###Telnet Credential Check Result###
    ###RDP Credential Check Result###
Do you want to search within the report? (yes/no)
no
Thats all folks. Exiting now!
```

1. Search function to search according to keyword input by user. The search function also allows user to repeat the search function until user is done.

Discussion

As mentioned earlier, the main objectives of the automation process are:

1. Use Nmap port scan to scan the network for ports and services
2. Map vulnerabilities using Nmap-vulners scan
3. Check for weak login credentials using Medusa to brute force the various services

Upon testing the script, we are able to achieve the above objectives and use automation to identify the vulnerabilities present in the network. By doing so, we can come up with counter measures to improve on the security aspect of the network.

However, the main challenge faced is that the scanning process actually is very time consuming, especially when scanning a large network. Nmap port scans and brute forcing process is a long process and often, we might not have that much time to carry out the scan so this is an area that has to be improved.

Conclusion

In conclusion, security checks and penetration testing are very important parts of the cyber security scope as it allows us to identify vulnerabilities and rectify the issues.

Using automation on the above mentioned security aspects allow us to stay up-to-date and protect our assets. Automation can greatly reduce the work hours required to carry out the checks as compared to performing them manually. It also reduces human fatigue and thus reduces the probability of human errors when carrying out repetitive system checks.

References

[1] Esteban Borges, “**How to Detect CVEs Using Nmap Vulnerability Scan Scripts**,” Security Trails, May 26 2020. [Online]. Available: https://securitytrails.com/blog/nmap-vulnerability-scan [Accessed: Dec 27, 2023].

[2] Robert Gilbert, “**Scanning for Weak MS-SQL Passwords Using NMap and Medusa**,” Halock, 2011. [Online]. Available: https://www.halock.com/scanning-weak-ms-sql-passwords-nmap-medusa/ [Accessed: Dec 27, 2023].