

ENPM695 Final Project Report

Vinaykumar Yennam

Umi Hani Bacha

Pragya Gupta

Dheepshika Raghunathan

Contents

1. Task Group 1 – Evaluating the Security of a System	4
1.1. Task 1 – Determining the running and open services on the system.....	4
1.1.1. Nmap scan	4
1.1.1.1. Open Ports.....	4
1.1.1.2. Running services in the system	5
1.1.2. Nessus scan.....	5
1.2. Task 2 – Access the system by exploiting a vulnerable running or open service	5
1.2.1. Root Password	5
1.3. Task 3 – Detail the Flaws in the webserver running on the system	6
1.3.1. Version is not up to date	6
1.3.2. Possible compromise through Apache misconfiguration.....	6
1.3.3. Compromise through a vulnerability of the application	6
1.3.4. Compromising the system to vulnerabilities in DVWA and Mutillidae	6
1.3.4.1. DVWA Vulnerabilities	6
1.3.4.2. Mutillidae Vulnerabilities	7
1.4. Task 4 – Crack Passwords.....	7
1.5. Task 5 – Find information stored in specific file	7
1.6. Task 6 – Define Attack Surface of the System	7
1.7. Task 7 – Develop Threat Model for the system and detail various Threat Vectors	8
1.7.1. General Threats	8
1.7.2. Threats to Open Ports.....	8
1.7.2.1. Port 22 – SSH	8
1.7.2.2. PORT 25 – SMTP	9
1.7.2.3. PORT 80 – HTTP.....	9
1.7.2.4. Port 110 – POP3	9
1.7.2.5. Port 143 – IMAP	9
1.7.2.6. Dovecot IMAP.....	9
1.7.2.7. Postfix SMTP	10
1.7.3. STRIDE Model	10
1.7.4. DREAD Model.....	13

1.7.4.1.	PORT 22 – SSH	13
1.7.4.2.	PORT 25 – SMTP	13
1.7.4.3.	PORT 80 – HTTP	13
1.7.4.4.	PORT 110	13
1.7.4.5.	PORT 143	14
1.7.4.6.	WEB SERVER	14
1.7.4.7.	MAIL SERVER	14
1.7.4.8.	BROWSER CLIENT	14
1.7.4.9.	MySQL SERVER	14
1.7.4.10.	AUTHENTICATION PROCESS	14
1.7.4.11.	FILE SYSTEM	15
2.	Task Group 2 – Improve the Security of a System.....	16
2.1.	Task 1 – Develop Threat Model and Attack Surface Analysis.....	16
2.1.1.	Nmap Scan of hardened System.....	16
2.1.2.	Attack Surface Analysis.....	16
2.1.3.	Nessus Scan.....	16
2.1.4.	Threat Modelling	16
2.1.5.	STRIDE Model	17
2.2.	Task 2 – Lock down the Server.....	19
2.2.1.	Setting up HTTPS.....	19
2.2.2.	Removing banners to eliminate identifying information that the web server gives out.....	20
2.2.3.	Jailing the web server	20
2.2.4.	Hardening Process	21
2.3.	Task 3 – Find Cogs, Inc’s secret file	21

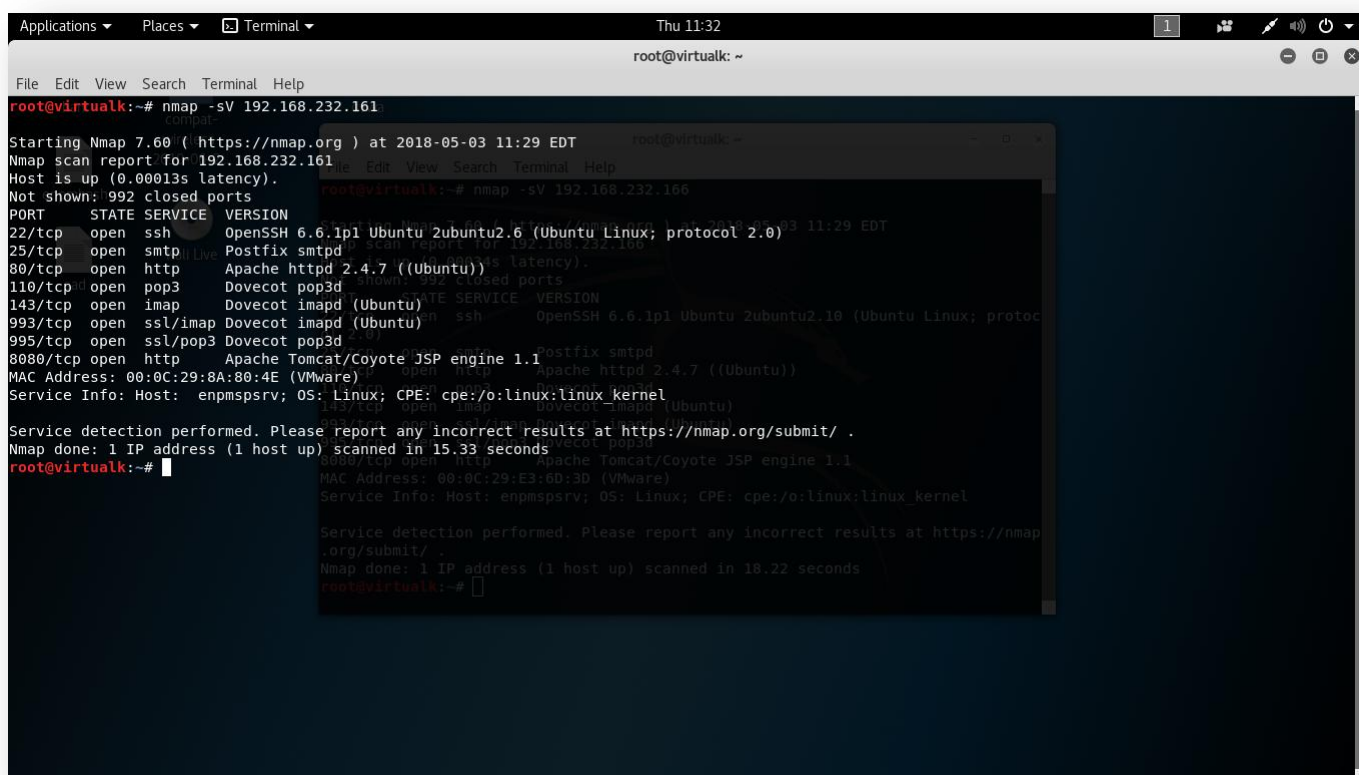
1. Task Group 1 – Evaluating the Security of a System

1.1. Task 1 – Determining the running and open services on the system

Nmap and Nessus scans were run to identify the open and running services.

1.1.1. Nmap scan

The Nmap scan helped us in identifying the open ports and running services on the system.



```
root@virtuallk:~# nmap -sV 192.168.232.161
Starting Nmap 7.60 ( https://nmap.org ) at 2018-05-03 11:29 EDT
Nmap scan report for 192.168.232.161
Host is up (0.00013s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.6 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp      Postfix smtpd
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
110/tcp   open  pop3      Dovecot pop3d
143/tcp   open  imap      Dovecot imapd (Ubuntu)
993/tcp   open  ssl/imap  Dovecot imapd (Ubuntu)
995/tcp   open  ssl/pop3  Dovecot pop3d
8080/tcp  open  http      Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:8A:80:4E (VMware)
Service Info: Host: enpmssprv; 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 15.33 seconds
root@virtuallk:~#
```

1.1.1.1. Open Ports

1. PORT 22/TCP – SSH (OpenSSH 6.6.1p1)
2. PORT 25/TCP – SMTP (Postfix smtpd)
3. PORT 80/TCP – HTTP (Apache httpd 2.4.7)
4. PORT 110/TCP – POP3 (Dovecot POP3d)
5. PORT 143/TCP – IMAP (Dovecot IMAP3d)
6. PORT 993/TCP – SSL/IMAP (Dovecot IMAPd)
7. PORT 995/TCP – SSL/POP3(Dovecot POP3d)

8. PORT 8080/TCP – HTTP (Tomcat/Coyote JSP engine 1.1)

1.1.1.2. Running services in the system

1. Dovecot/Postfix
2. MySQL
3. SSH
4. Apache2
5. Tomcat7

1.1.2. Nessus scan



1.2. Task 2 – Access the system by exploiting a vulnerable running or open service

1. The password for enpmuser account was guessed. This provided initial access to the system
 - a. Password was **enpmuser**
2. Found the rest of the users by navigating to **/home** directory. Users found were
 - a. smithy
 - b. ppan
 - c. chook
 - d. admin
 - e. enpmuser
3. DirtyCow local exploit used to gain root access
4. Brute Forced the password for **smithy** using Hydra
 - a. The password was **password**
5. Exploited SQL Injection vulnerability in the running DVWA to find the credentials for users **ppan** and **chook**.
 - a. **SQLMap** was used to perform SQL Injection and retrieve database
 - b. The passwords were
 - i. ppan – NotTelling
 - ii. chook – JollyRoger
6. Found the credentials for Tomcat7
 - a. tomcat – t0mcat
 - b. admin - @dm!n

1.2.1. Root Password

We made many attempts to find the root password. Initially we tried to brute force for the password using various detailed word lists.

After we received the hint that the password is a 14-character long string from a song in the movie 'My Fair Lady', we took up all the song lyrics and wrote a script to strip each line of any spaces or special characters. Then we found 14-character long strings. We then tried all these strings in all the cases.

After we received the final hint, we narrowed down the song to be 'The rain in Spain'. We found 14-character strings - "the rain in spain", up and down until", in spain in spain, one a.m. two a.m."

Then we used the below script to replace characters with numbers and metacharacters to create an exhaustive wordlist.



wordlist_creator.py.back

This script generated the below wordlist, which we used to crack the root password.



wordlist.zip

Unfortunately, we were not able to find the password for root

1.3. Task 3 – Detail the Flaws in the webserver running on the system

The web server running on the system is Apache 2.4.7. The various flaws associated with the same are

1.3.1. Version is not up to date

The system uses Apache 2.4.7 which is an outdated version and has a number of vulnerabilities and attacks like DOS, XSS, Overflows and Remote code executions are easy. Apache 2.4.33 is the latest version available and is better than the older versions.

1.3.2. Possible compromise through Apache misconfiguration

The default files and services provide a means for an attacker to reveal sensitive information and may also be used to elevate privileges. One way to prevent this is regular independent configuration assessments.

1.3.3. Compromise through a vulnerability of the application

The functional level of the application may cause problems by selecting the valid and invalid inputs. Better is to not allow read/write access or compiler.

1.3.4. Compromising the system to vulnerabilities in DVWA and Mutillidae

DVWA and Mutillidae are present which can be easy targets for the attackers. They are full of vulnerabilities and can allow easy attacks like SQL injections, Cross-Site Scripting (XSS), etc.

1.3.4.1. DVWA Vulnerabilities

1. Brute-force login

2. Command Execution
3. CSRF (Cross Site Request Forgery)
4. File Inclusion
5. File Upload
6. SQL Injection
7. Cross-Site scripting(XSS)
8. Insecure CAPTCHA
9. Weak Session IDs

1.3.4.2. Multitool Vulnerabilities

1. Cross-site scripting
2. SQL injection
3. Broken authentication
4. Broken Access control
5. Sensitive Data Exposure

1.4. Task 4 – Crack Passwords

User	Password
enpmuser	enpmuser
ppan	NotTelling
smithy	password
chook	JollyRoger
root	

1.5. Task 5 – Find information stored in specific file

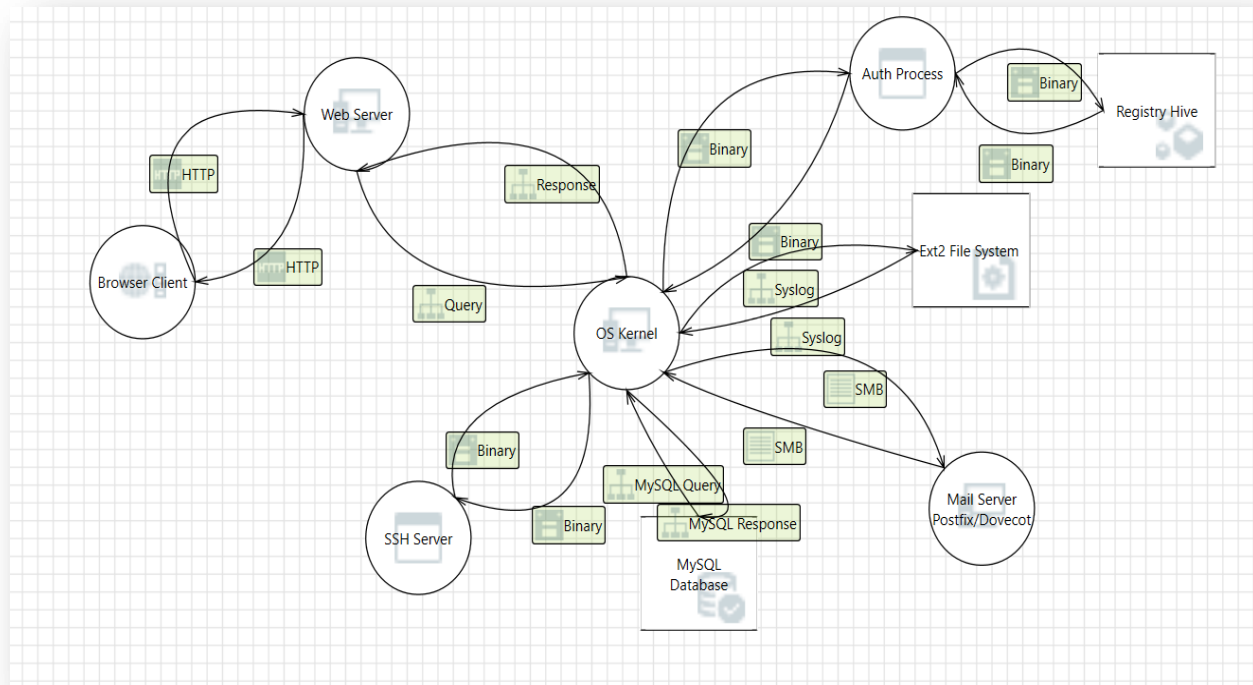
This task was waived off because there was no such file in the machine

1.6. Task 6 – Define Attack Surface of the System

The attack surface of the system is the open ports which an attacker can use as a source to get into the system.

- 22/TCP – SSH
- 25/TCP – SMTP
- 80/TCP – HTTP
- 110/TCP – POP3
- 143/TCP – IMAP
- 993/TCP – IMAPS
- 995/TCP – POP3S
- 8080/TCP – HTTP-PROXY

1.7. Task 7 – Develop Threat Model for the system and detail various Threat Vectors



1.7.1. General Threats

1. Weak passwords
2. Man-in-the-Middle attack
3. Sniffing of traffic
4. SQL injection
5. Cross-site scripting
6. Denial of service
7. Brute-Force
8. Dictionary attacks
9. Overflows
10. Remote code injection

1.7.2. Threats to Open Ports

1.7.2.1. Port 22 – SSH

1. Weak passwords can make SSH and port 22 easy targets
2. Default or easily guessed user names and passwords

1.7.2.2. PORT 25 – SMTP

1. **Attacks using account enumeration**

Allow attackers to verify what mailing lists exist on a server.

2. **E-mail header disclosures**

The attackers might find critical pieces of information like Internal IP address of the e-mail client machine, Software versions of the client and e-mail server along with their vulnerabilities as well as hostnames that can divulge the network naming convention

3. **Malware**

Mail systems are regularly attacked by such malware as viruses and worms. This generally happens if there is no antivirus software or if it's not working.

4. **RELAY**

SMTP relay lets users send e-mails through external servers. Open e-mail relays aren't the problem they used to be, but we still need to check for them. Spammers and hackers can use an e-mail server to send spam or malware through e-mail under the guise of the unsuspecting open-relay owner.

1.7.2.3. PORT 80 – HTTP

1. A number of Trojans/backdoors can be used on this port
2. Denial of Service
3. Sniffing of traffic
4. SQL injections, Cross site scripting attacks, Cross-site request forgery
5. Information leakage

1.7.2.4. Port 110 – POP3

1. No auditing of connections and attempts
2. Buffer overflows that allow compromise during login
3. Format string vulnerability allows attackers to execute arbitrary code
4. POP3 on port 110 is the older of the two popular protocols used to retrieve email from remote mail servers
5. Brute Force
6. Denial of service
7. Banner grabbing

1.7.2.5. Port 143 – IMAP

1. Format string vulnerability allows attackers to execute arbitrary code.
2. Remote attackers can cause Denial of service.

1.7.2.6. Dovecot IMAP

1. Brute force attacks
2. Denial of service attacks
3. Parse attacks

1.7.2.7. Postfix SMTP

1. TLS based attacks (renegotiation attacks)
2. DoS
3. Unauthorized access
4. Server port kept busy by Storm zombies
5. SQL Injection
6. Man-in-the-Middle attacks to insert commands into encrypted SMTP sessions by sending a cleartext command that is processed after TLS is in place, related to a "plaintext command injection" attack.

1.7.3. STRIDE Model

	SPOOFING	TAMPERING	REPUDIATION	INFORMATION DISCLOSURE	DENIAL OF SERVICE	ELEVATION OF PRIVILEGE
Port 22(SSH)	Yes If the attacker can replace the SSH keys, he can spoof the user identity.	Yes Attacker can get access to the SSH code and tamper with it.	Yes Attacker can use someone's else credentials and later deny.	Yes Disclose info about the protocol version, ubuntu version.	Yes Connection can be refused. Users can be blocked after certain attempts	Yes Attacker can compromise the SSH code running with high privilege and cause EOP.
PORT 110(POP3)	Yes Attackers can spoof the mails by simply setting the display name or "from" field of outgoing messages to show a name or address other than the actual one which the message is sent.	Yes Attackers can use the format string vulnerabilities and access the system	Yes It is possible that the sender will deny that he sent the message. can be dealt by using SSL/TLS.	Yes Attackers can use banners to leak information regarding the system.	Yes Remote attackers can cause denial of service.	Yes The attackers can send phishing emails and ask the users for their credentials.
PORT 143(IMAP)	Yes Attackers can spoof the mails by	Yes Attackers can use the format string	Yes A sender can deny that a particular	Yes If attackers use format string vulnerability	Yes Remote attackers can cause	Yes The attackers can send phishing

	simply setting the display name or "from" field of outgoing messages to show a name or address other than the actual one which the message is sent.	vulnerabilities and access the system.	message came from him.	present, they can leak any information they want.	dos.	emails and ask the users for their credentials.
PORT 25-SMTP	Yes Attackers can exploit SMTP by setting up their own MTA (message transfer agent	Yes Attackers can use the vulnerabilities in the mail server and access the system	Yes A sender can deny that a particular message came from him.	Yes Attackers can use account enumeration to get information about the mailing lists.	Yes An attacker can use a flooding attack to cause denial of service.	Yes The attackers can send phishing emails and ask the users for their credentials.
Port 80 HTTP	Yes Attackers can use HTTP Flood attacks and cause IP spoofing.	Yes If web server gets access to memory or given the ability to control what browser client executes		Yes The attacker can grab banners and disclose file paths.	Yes The attacker can use HTTP GET or POST requests to cause a DDOS attack	Yes
Port 8080 Apache tomcat	Yes Attackers can use the vulnerabilities in tomcat to spoof the name of the target user and client IP address	Yes Attackers can use SQL injection, CSRF, memory corruption and tamper with the information.		Yes Attackers can use vulnerabilities in tomcat to get access to sensitive information	Yes Attackers can use vulnerabilities in common file uploads and cause denial of service.	Yes Attackers can get unrestricted access to global resources.
Web server	Yes Web Server may be	Yes The web server could	Yes Web Server claims that it did	Yes Sniffing attacks are very	Yes The attacker	Yes An attacker may pass

	spoofed by an attacker and this may lead to unauthorized access to Authentication Process	be a subject to a cross-site scripting attack because it does not sanitize untrusted input	not receive data from a source outside the trust boundary	common	can cause the web server to halt, stop or run slowly.	data into web server in order to change the program flow execution.
Mail Server	Yes The attacker can use the user credentials to send email to other users	Yes Attacker can edit the mail boxes.	Yes Attacker can send a mail on behalf of a legitimate user and later deny it	Yes The attacker can get access to the mail server and get the information in the mails	Yes The attacker can change user password and stop them from accessing the system	Yes Attacker can get access to the code behind the mail server and cause EOP.
Browser Client	Yes Browser Client may be spoofed by an attacker and this may lead to unauthorized access to Web Server.	-	Yes The client might deny that it tried accessing the web server.	-	-	-
MySQL Server	Yes Attackers can access to the server if weak passwords are used.	Yes SQL injection attacks are possible if vulnerabilities are found		Yes Credentials / data flow can be sniffed by the attacker.	Yes Requests can be timed out if resource consumption attacks are not dealt with	Yes the attacker can use SQL injection, unauthorized access or eavesdropping to elevate privilege.
Authentication Process	Yes Authentication Process may be spoofed by an attacker and this may lead to information disclosure by Web Server	Yes The authentication process could be a subject to a cross-site scripting attack because it does not sanitize untrusted input	Yes Authentication Process claims that it did not receive data from a source outside the trust boundary	Yes Credentials / data flow can be sniffed by the attacker	Yes Requests can be timed out if resource consumption attacks are not dealt with.	Yes Authentication Process may be able to remotely execute code for Web Server
File System		Yes Attacker can		Yes Credentials / data	Yes Requests can	

	edit the text files on the file system	flow can be sniffed by the attacker	be timed out if resource consumption attacks are not dealt with.
--	--	-------------------------------------	--

1.7.4. DREAD Model

	High - 3	Medium - 2	Low - 1
(D) – Damage potential	High damage to the system	Medium damage to the system	Low damage to the system
(R) Reproducibility	Very easy to reproduce	Requires one or two steps to be reproduced	Very hard or impossible
(E) – Exploitability	Very easy even using a browser	Tools required to perform exploits	Advanced programming knowledge required
(A) – Affected users	All the users	Many but not all users	Very few users
(D) – Discoverability	Very easy to discover vulnerability	Can be found using monitoring or guessing techniques	Very hard to find- requires source code.

1.7.4.1. PORT 22 – SSH

Threats on Port 22 (SSH)	D	R	E	A	D	Risk value
SSH key replacement	3	1	1	3	1	9
Tampering SSH code	3	1	1	3	1	9
Stolen credentials	3	3	2	3	2	13
OS information disclosure	2	3	3	2	3	13
Connection refusal(DOS)	3	2	2	3	2	12
Privilege escalation using SSH code	3	1	1	3	2	10

1.7.4.2. PORT 25 – SMTP

Threats on Port 25 (SMTP)	D	R	E	A	D	Risk value
Account Enumeration	2	2	3	3	2	12
Flooding attacks.	3	2	2	2	2	11

1.7.4.3. PORT 80 – HTTP

Threats on Port 80 (HTTP)	D	R	E	A	D	Risk value
SQL injection	2	2	1	3	1	9
Banner grabbing	2	3	3	1	3	12
DDOS	3	2	2	3	1	11

1.7.4.4. PORT 110

Threats on Port 110	D	R	E	A	D	Risk value
Format string vulnerability	3	2	1	2	3	11

Banner grabbing	2	3	3	1	3	12
Denial of service	3	2	2	3	1	11

1.7.4.5. PORT 143

Threats on Port 143(IMAP)	D	R	E	A	D	Risk value
Format string vulnerability	3	2	1	2	3	11
Banner grabbing	2	3	3	1	3	12
Denial of service	3	2	2	3	1	11

1.7.4.6. WEB SERVER

Threats on web server	D	R	E	A	D	Risk value
Unauthorized access using spoofing	3	2	2	3	2	12
Cross-site scripting	2	2	2	3	2	11
Repudiation	2	2	1	2	3	10
Sniffing attacks	1	3	2	1	2	9
Stop running due to DDOS attacks	3	2	2	3	1	11
Change the program flow execution	3	1	1	3	1	9

1.7.4.7. MAIL SERVER

Threats on Mail Server	D	R	E	A	D	Risk value
Stolen credentials	2	3	2	2	2	11
Editing mail boxes	2	2	1	2	1	8
Mailing in behalf of the legitimate user	2	3	1	1	2	9
Disclosing information in mails by getting access	2	3	1	2	2	10
Change the user password cause DOS	3	2	1	3	1	10
Access to code behind server	3	1	1	3	1	9

1.7.4.8. BROWSER CLIENT

Threats on Browser Client	D	R	E	A	D	Risk value
Spoofing to get unauthorized access	3	3	2	3	2	13
Deny accessing the system(repudiate)	2	2	1	3	1	9

1.7.4.9. MySQL SERVER

Threats on MySQL server	D	R	E	A	D	Risk value
Weak passwords	2	3	3	1	3	12
SQL injection	2	3	1	3	1	10
Data flow sniffing	1	3	2	2	2	10
Requests timed out due to resource consumption	3	3	3	3	1	13

1.7.4.10. AUTHENTICATION PROCESS

Threats on Authentication Process	D	R	E	A	D	Risk value
Spoofing	1	3	2	2	2	10
Cross-site scripting	2	2	2	2	2	10
Credential/data flow sniffing	1	3	2	1	2	9
Requests timed out(DOS)	3	3	3	3	1	11
Remote execution of code	3	1	1	3	1	9

1.7.4.11. FILE SYSTEM

Threats on File System	D	R	E	A	D	Risk value
Editing files on the system	2	1	1	3	2	10
Credentials sniffed	1	3	2	1	2	9
Requests timed out (DOS)	3	2	3	3	2	12

2. Task Group 2 – Improve the Security of a System

2.1. Task 1 – Develop Threat Model and Attack Surface Analysis

2.1.1. Nmap Scan of hardened System

We first did an Nmap scan to determine the ports that are currently open in the hardened system.

```
root@shika:~# nmap -sV 192.168.159.138 -p-
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-13 22:15 EDT
Nmap scan report for 192.168.159.138
Host is up (0.00057s latency).
Not shown: 65530 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 (protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
80/tcp    open  http         Apache httpd
443/tcp   open  ssl/http     Apache httpd
8080/tcp   open  http-proxy   Filtered
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?new-service :
SF-Port8080-TCP:V=7.70%I=7%D=5/13%Time=5AF8F1AB%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,222,"HTTP/1.1\x20200\x200K\r\nAccept-Ranges:\x20bytes\r\nETag
SF::\x20W/\x20310-1525553491000"\r\nLast-Modified:\x20Sat,\x2005\x20May\x20
SF:2018\x2020:51:31\x20GMT\r\nContent-Type:\x20text/html\r\nContent-Length
SF::\x20310\r\nDate:\x20Mon,\x2014\x20May\x202018\x2002:17:15\x20GMT\r\nCo
SF:nnection:\x20close\r\nServer:\x20Filtered\r\n\r\n<\x20?xml\x20version=\x20"1\
SF:.0"\x20encoding=\x20"ISO-8859-1"\x20?>\n<!DOCTYPE\x20html\x20PUBLIC\x20"-/
SF:/W3C//DTD\x20XHTML\x201.0\x20Strict//EN"\x20"\x20\x20"\x20"http://www.w
SF:3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">\n<html\x20xmlns=\x20"http://www
SF:.w3.org/1999/xhtml"\x20xml:lang=\x20"en"\x20lang=\x20"en">\n<head>\n\x20
SF:\x20\x20\x20<title>Apache</title>\n</head>\n<\n<body>\n<h1>It\x20works\x
SF:20!</h1>\n</body>\n</html>\n")%r(HTTPOptions,9E,"HTTP/1.1\x20200\x200K
```

2.1.2. Attack Surface Analysis

- Port 22/TCP – SSH (OpenSSH 6.6.1p1)
- Port 25/TCP – SMTP (Postfix SMTPD)
- Port 80/TCP – Apache HTTPD
- PORT 110/TCP – POP3 (Dovecot POP3d)
- PORT 143/TCP – IMAP (Dovecot IMAP3d)
- Port 443/TCP – Apache HTTPD
- Port 8080/TCP – HTTP-Proxy (Filtered)

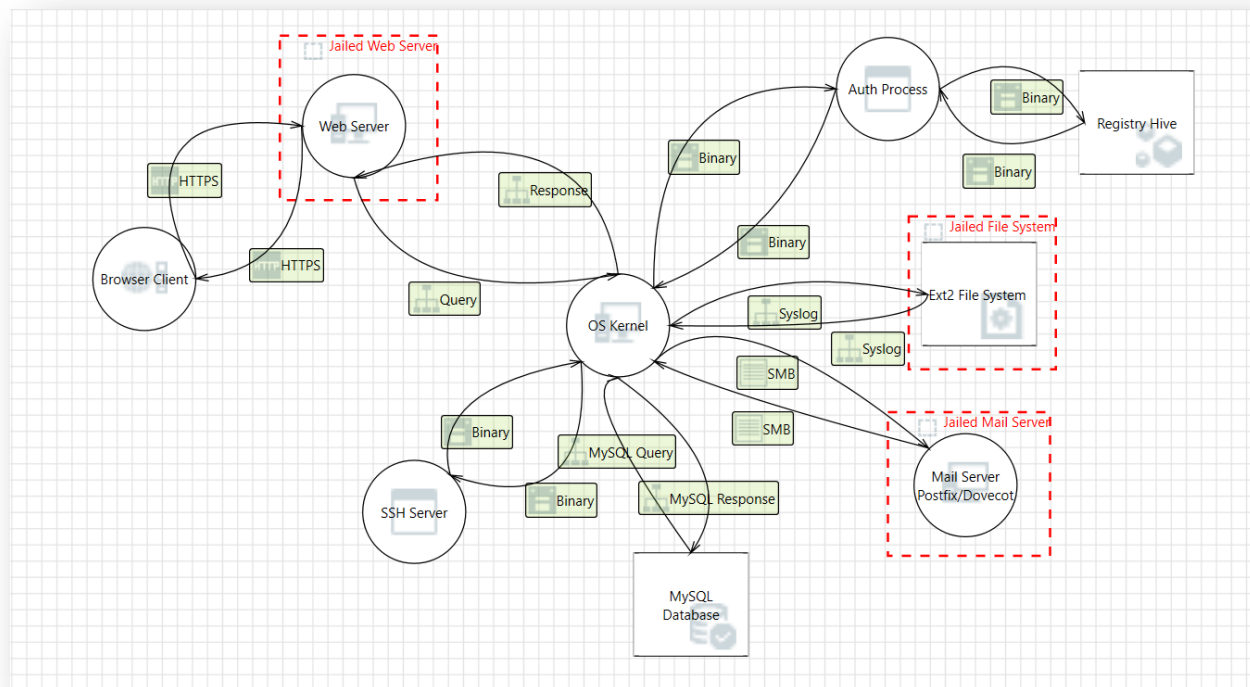
2.1.3. Nessus Scan



Nessus_Scan_Hardened_Machine.pdf

2.1.4. Threat Modelling

Threat modelling diagram after hardening the system,



2.1.5. STRIDE Model

	SPOOFING	TAMPERING	REPUDIATION	INFORMATION DISCLOSURE	DENIAL OF SERVICE	ELEVATION OF PRIVILEGE
PORT 80 – HTTPS	No We authenticated the server by changing the password.	Yes Man-in-the-middle attacks are still possible. Fiddler can be used.	Yes Still be possible at the origin	No Using SSH makes it encrypted.	Yes The attacker can use HTTP GET or POST requests to cause a DDOS attack	Yes We have use chroot thus the attackers can't get any privileges.
Port 25 – POSTFIX SMTP Updated	Yes Attackers can exploit SMTP by setting up their own MTA(message transfer agent	Yes Although updates the attackers can use the vulnerabilities in the mail server and access the system	Yes A sender can deny that a particular message came from him.	No We removed the banner thus making it difficult to see the version.	Yes An attacker can use a flooding attack to cause denial of service.	Yes The attackers can send phishing emails and ask the users for their credentials

PORT 22 – SSH Updated	Yes If the attacker can replace the SSH keys, he can spoof the user identity.	Yes Attacker can get access to the SSH code and tamper with it.	Yes Attacker can use someone's else credentials and later deny.	Yes Disclose info about the protocol version, ubuntu version.	No Using maxauthtries limited the number of tries	Yes Attacker can compromise the SSH code running with high privilege and cause EOP
PORT 8080- Apache Tomcat Updated	No authenticated tomcat changing password	Yes Attackers can use SQL injection, CSRF, memory corruption and tamper with the information.		No We removed the banner thus making it difficult to see the version	Yes Attackers can use vulnerabilities in common file uploads and cause denial of service.	Yes Attackers can get unrestricted access to global resources.
Port 143-IMAP	Yes Attackers can spoof the mails by simply setting the display name or "from" field of outgoing messages to show a name or address other than the actual one which the message is sent	Yes Attackers can use the format string vulnerabilities and access the system.	No Since we set up an SSL connection difficult to do this	No We removed the banner thus making it difficult to see the version	Yes Remote attackers can cause denial of service.	Yes The attackers can send phishing emails and ask the users for their credentials.
Port 110 -POP3 Updated.	Yes Attackers can spoof the mails by simply setting the display	Yes Attackers can use the format string vulnerabilities and access	No Since we set up an SSL connection difficult to	No We removed the banner thus making it difficult to see the	Yes Remote attackers can cause denial of service.	Yes The attackers can send phishing emails and ask the users

name or "from" field of outgoing messages to show a name or address other than the actual one which the message is sent	the system	do this.	version	for their credentials.
--	------------	----------	---------	---------------------------

2.2. Task 2 – Lock down the Server

2.2.1. Setting up HTTPS

1. Enable a module using `-sudo a2enmod ssl`
2. Restart the web server after enabling the SSL
sudo service apache2 restart
3. Create a Self-Signed SSL Certificate
 - a. Creating a subdirectory within Apache's configuration hierarchy to place the certificate files
sudo mkdir /etc/apache2/ssl
 - b. *sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt*
 - c. After answering some questions, the key and certificate will be created and placed in your `/etc/apache2/ssl` directory.
4. Configure Apache to Use SSL:
 - a. Open the file with root privileges
sudo nano /etc/apache2/sites-available/default-ssl.conf
 - b. Make changes as required
5. Activate the SSL Virtual Host
sudo a2ensite default-ssl.conf
sudo service apache2 restart

This enables a new virtual host, which will serve encrypted content using the SSL certificate we created.

6. Testing the Setup

2.2.2. Removing banners to eliminate identifying information that the web server gives out

- Apache default page- removing default data.
 - Editing /var/www/index.html
 - Removing the identifying information.
- Securing SSHd
 - Changing **maxAuthTries = 3** will limit the brute force limit to 3 per connection and refuse connection after.
- Removing identifying information from:
 - Mail Server – Removing Postfix banner number and version information in the scan
 - Sudo vi /etc/postfix/main.cf
 - Change the smtpd_banner line to “smtpd_banner = \$myhostname”
 - sudo service postfix restart
 - Tomcat – We made changes in the server.xml file using
 - find / -name server.xml
 - Added a server parameter to this line to specify how we want Tomcat to respond when a user asks for the system version.
 - Connector port=8080”
 - Server= “Filtered”
 - We tried to find a file named ServerInfo.properties to remove the version information of apache tomcat but couldn't find the file.

2.2.3. Jailing the web server

- Jailed the webserver and users
 - Used <https://olivier.sessink.nl/jailkit/index.html#intro> to download the JailKit
 - Jailing user's **smithy chook enpmuser** using command
jk_init -v -j /home/jail basicshell editors extendedshell netutils ssh sftp scp
jk_jailuser -m -j /home/jail smithy chook enpmuser
 - There were few dependency issues which were solved manually
 - For **Jailing web server**, the commands were
jk_init -v -j /home/webjail apache2
 - Few dependency issues were solved manually
 - Created a jail.sh bash script with contents as
jk_chrootlaunch -j /home/webjail -x /home/webjail/usr/sbin/apachectl -- start
 - This Bash script was made to run on every startup by editing the /etc/rc.d/rc.local startup file
 - And added this line in /etc/rc.d/rc.local
sh /home/webjail/jail.sh

2.2.4. Hardening Process

1. Update all services as per ubuntu repository
 - a. apt update
 - b. apt upgrade
2. Purged and reinstalled MySQL from v5.5 to v5.6
3. Jailed the users enpmuser, chook admin and smithy
 - a. Installed JailKit tool and used the following commands
 - b. The jailed users were given limited bash commands and no sudo privileges
4. Changed the passwords for all the users
5. Changed passwords for tomcat users in tomcat-user.xml
6. Changed the name of DVWA-master and mutillidae to ENPM695_needs_a_lot_of_security_upgrades and ENPM695_needs_a_lot_of_security_upgrades_as_well
7. Kept the file ENPM_18.pdf in /root with permission as 700.
8. Removed sudo privileges for chook:

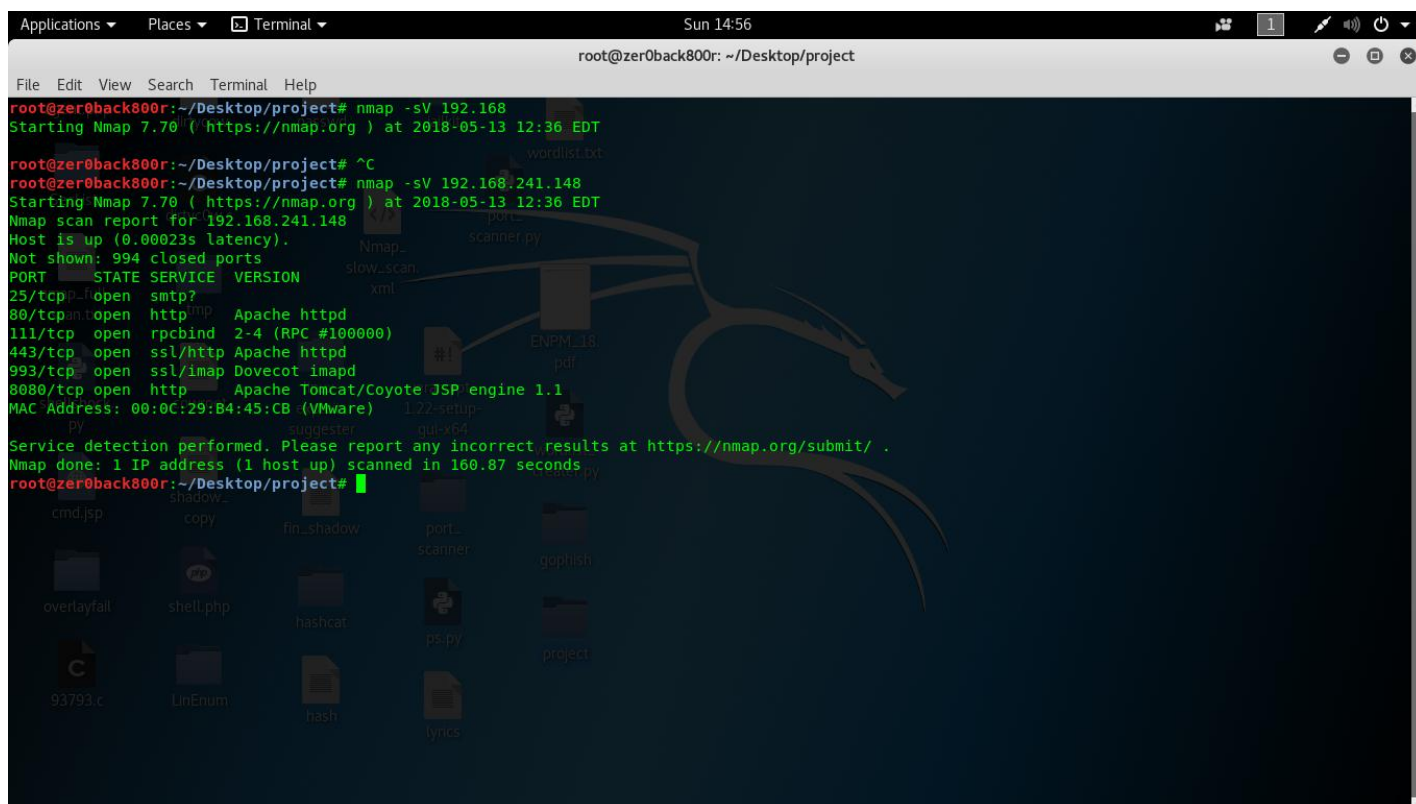
userdel chook sudo

9. File was placed in /root with only read, write and execute permissions for the root user:

chmod 700 ENPM_18.pdf

2.3. Task 3 – Find Cogs, Inc's secret file

The nmap scan screenshot of the team we got the Virtual Machine is attached below



```
root@zer0back800r: ~/Desktop/project
File Edit View Search Terminal Help
root@zer0back800r:~/Desktop/project# nmap -sV 192.168
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-13 12:36 EDT
root@zer0back800r:~/Desktop/project# ^C
root@zer0back800r:~/Desktop/project# nmap -sV 192.168.241.148
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-13 12:36 EDT
Nmap scan report for 192.168.241.148
Host is up (0.00023s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE VERSION
25/tcp    open  smtp?
80/tcp    open  http    Apache httpd
111/tcp   open  rpcbind 2-4 (RPC #100000)
443/tcp   open  ssl/http Apache httpd
993/tcp   open  ssl/imap Dovecot imapd
8080/tcp  open  http    Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:B4:45:CB (VMware)
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 160.87 seconds
root@zer0back800r:~/Desktop/project#
```

- SSH was hardened so that only known hosts were allowed. So hydra on SSH didn't work.
- So, we tried attacking tomcat7. First, tomcat7 credentials were cracked using metasploit auxilliary scanner module. Later, used tomcat upload vulnerability to upload a reverse shell and got access to the system
- Didn't have enough privileges to parse through the filesystem. So, used a local linux privilege escalation exploit dirty Copy On Write.
- Instantly, we got the root access. For continued access, we changed the root password to null.
- We logged in as root to find the secret file. Initially, the vm the team gave didn't include a file.
- We asked the team to verify. Later, they uploaded the new vm. I followed the same exploit method. However, this time they changed the login credentials. We changed the default wordlist and used a wordlist that was 13 Gigs big in size. We divided the file into 4 parts, and all four individually tried to crack the login credentials. Finally, it was cracked. We used this credentials for further exploit.
- After getting access to the system, we found very difficult to find the file. Asked help from the team 11 about the location. It was in the /dev/sda3. We mounted the file and listed the directory. The File ENPM_11.pdf had no read, write or execute permissions for any user. So, we changed the permission using chmod 777 ENPM_18.pdf and later downloaded the file into thumb drive. We have also attached the file as a separate attachment.
- The MD5 checksum of the file is **107A462A48EAF735A78A98FA5464742F**
- And the content of the file is



ENPM_11.pdf